An Information Theoretic Framework for Web Inference Detection

Hoi Le Thi and Reihaneh Safavi-Naini
Department of Computer Science
University of Calgary, Canada
{leh, rei}@ucalgary.ca

ABSTRACT

Document redaction is widely used to protect sensitive information in published documents. In a basic redaction system, sensitive and identifying terms are removed from the document. Web-based inference is an attack on redaction systems whereby the redacted document is linked with other publicly available documents to infer the removed parts. Web-based inference also provides an approach for detecting unwanted inferences and so constructing secure redaction systems. Previous works on web-based inference used general keyword extraction methods for document representation. We propose a systematic approach, based on information theoretic concepts and measures, to rank the words in a document for purpose of inference detection. We extend our results to the case of multiple sensitive words and propose a metric that takes into account possible relationship of the sensitive words and results in an effective and efficient inference detection system.

Using a number of experiments we show that our approach, when used for document redaction, substantially reduce the number of inferences that are left in a document. We describe our approach, present the experiment results, and outline future work.

Categories and Subject Descriptors

H.1.1 [**Systems and Information Theory**]: Information theory; H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

inference detection, information theory, document redaction, web-based inference detection

1. INTRODUCTION

The knowledge encapsulated in the Web combined with the power of search engines, makes it easy to look for unknowns and make unexpected findings. Search queries on multiple words and phrases that are relevant to a search target, can narrow down the search results and in many cases identify a single target. On the other hand this power of Web search makes it very hard to "hide" information or remain anonymous. Inferring individuals' details by combining anonymized Netflix databases and other public data made headline news a few years ago [22].

An important security mechanism that is severely challenged by the public knowledge on the Web and the power of search engines, is document redaction. Document redaction systems protect sensitive contents of documents by removing a subset of words. Document redaction is widely used in practice and is the main protection mechanism that is used for privacy protection when complying with Freedom of Information Legislations. For example in the U.S., redaction is used to protect patients' privacy when their information needs to be shared with other parties.

The information that is left in a redacted document however, can be combined with the public knowledge encapsulated in the Web to make inferences about the sensitive parts. Staddon et al. [24] showed a Web-based inference attack that recovers the removed words of a document using the public information on the Web. They noted that their approach can also be used for detecting unwanted inferences; that is inferences that could later leak information about the removed parts of the document. They also developed a set of tools to semi-automatically (with the aid of human) detect such inferences. In the rest of this paper we consider this latter application of their approach.

Web-inference system.
In a Web-inference system, a document that includes sensitive information, and a collection of documents (corpus) related to the sensitive topic, are at hand. The aim is to publish the document "safely": that is publish it such that the publication does not leak information about the sensitive content. The redacted document has to stay readable and so the number of removed words must be kept at a minimum. A Web-inference system uses the following steps: (i) use a Natural Language Processing (NPL) tool to extract a set of keywords from the document, (ii) query on subsets of these keywords to a Web search engine (e.g. Bing in our case), and (iii) analyze the returned documents to detect unwanted inferences. (Figure 1 shows the main steps of Web-inference detection approach for detecting unwanted inferences). An inference is modeled as co-occurrence of a set of words: if w_1 and w_2 both occur in many documents that include the sensitive word s, they are considered as precedent of an inference $w_1 \wedge w_2 \Rightarrow s$, where s is the sensitive word.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
AISec'12, October 19, 2012, Raleigh, North Carolina, USA.
Copyright 2012 ACM 978-1-4503-1664-4/12/10 ...$15.00.

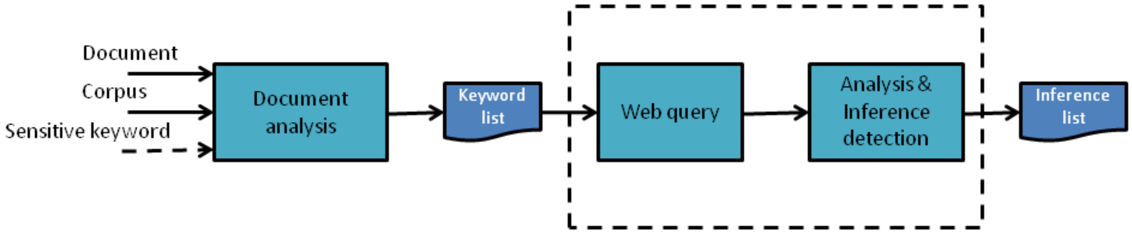


Figure 1: Main steps in inference detection.

The above framework can be instantiated using different algorithms for keyword extraction in Step (i) and inference detection in Step (iii), above. The effectiveness of each step in an instantiation can be measured by the number of inferences that can be detected, assuming other parts of the systems are kept fixed. For example, by keeping Steps (ii) and (iii) fixed, one can evaluate effectiveness of a particular keyword extraction algorithm (See 2.4).

To extract document keywords, Staddon et al. used a well-known metric, called TF.IDF (Term Frequency-Inverse Document Frequency) [19] that requires a set of documents that was related to the sensitive word. The TF.IDF value of a word w with respect to a document $\mathcal{D}$ and a corpus $\mathcal{C}$, is defined as $TF_w \times log(IDF_w)$, where TF_w is the number of times w occurs in $\mathcal{D}$ and $IDF_w = \frac{|\mathcal{C}|}{DF_w}$ with $|\mathcal{C}|$ is the number of documents in $\mathcal{C}$, and DF_w is the number of documents that contain the word w. TF.IDF tends to extract keywords that are frequent and specific to the document (IDF is inversely proportional to frequency of the occurrence of the word in the corpus). In this paper we focus on Step (i) and provide an information theoretic approach to keyword extraction, that naturally matches the inference model (word co-occurrence). For our future work we will use information theoretic measure for more accurate detection of inferences in Step (iii).

Our work

We propose an information theoretic approach to document keyword extraction for the purpose of inference detection, when inferences are modeled as word co-occurrence.

Keyword extraction. The goal is to select a set of keywords from the document that "best" capture the document with respect to the sensitive word(s). The set of keywords, (i) must be efficiently computable, and (ii) when used in the Web-inference system (Figure 1) should result in the least amount of undetected inferences in the final document, assuming other parts of the system are fixed.

Our key observation is that if keywords are extracted based on the strength of their relationship with the sensitive word that is calculated using the corpus documents, then in the Web search phase they would exhibit the similar relationship in the search results. Unlike the TF.IDF measure that focuses on the uniqueness of a document in a collection, our approach focuses on the relationship of words with the sensitive word(s) and considers those that are "more related" as more important. We use information theoretic measures to quantify the strength of the relationship.

Single sensitive word. Let the sensitive keyword be denoted by s_0. We rank the words in the document with respect to the strength of their relationship with s_0, and use the top ℓ ranked words to form the keyword list. The strength of the relationship of a word w_i with s_0, denoted by $I(w_i; s_0)$, is measured by the mutual information between two *binary random variables* X_{w_i} and X_{s_0} whose distributions correspond to the relative frequencies of the occurrences of the two words in the documents in the corpus $\mathcal{C}$, respectively. Note that X_{w_i} does not represent the frequency (number) of occurrences of w_i in $\mathcal{C}$. Rather, it captures presence of the word w_i in the corpus. Similarly, X_{w_i,s_0} represents co-occurrence of the word w_i and the sensitive word s_0 in $\mathcal{C}$. The list of values $I(w_i; s_0)$ are sorted, and the top l ranked words will be used as keywords.

Multiple sensitive words. In many cases more than one sensitive words must be protected. We consider *multiple sensitive keywords* $\{s_1 \cdots s_u\}$, and the aim is to detect unwanted inferences with respect to any of the words. Although this problem was not considered in [24], as we will explain in section 2.3, there are direct solutions that are based on single sensitive word algorithms. However these solutions are inefficient and require the single sensitive word algorithm to be repeated once for each sensitive word. We propose an information theoretic measure $I_{w,s_1,\cdots,s_u}$ that takes into account the strength of the relationship of w with respect to the whole set of sensitive words. When the sensitive words can be considered as "unrelated" (or independent), we will have:

$$I_{w_i,s_1,\cdots,s_u} = I(w_i; s_1) + I(w_i; s_2) + ... + I(w_i; s_u)$$

For the general case, the expression will be more complex and is given in Section 2.3.

Tool set and experiments.
We compare our proposed approach to [24] using the same framework and examples of Staddon et al. For this we developed a set of tools that allowed us to implement the framework and the two instantiations of keyword extraction algorithm.

We compare performance of the systems, and show the effectiveness of our keyword extraction approach through a number experiments. The results of our experiments (Section 3) show that our inference detection algorithm performs consistently better than [24] (almost double the percentage of inferences found) and [8].

We note that the result of the experiment in general, will depend on the choice of corpus. We show however that our algorithm is not very sensitive to this change, by repeating the experiments for 3 corpora with different qualities, where quality of a corpus is measured in terms of the relevance of the corpus to the sensitive word(s).

Significance and extensions. Modeling inference as co-occurrence has been considered in [24, 27] Our proposed information theoretic measure corresponds to the strength of co-occurrence of words in documents. The inference detection algorithm in Step (iii) also uses co-occurrence of words and so the two steps of the Web-inference system are aligned. (A simple inference detection procedure for Step (iii) in Figure 1, is to define a threshold γ and consider the word set used for the query as precedent of an unwanted inference, if the sensitive word(s) occurs in one of the top γ returned documents.)

Our approach can be used to model more complex inferences. As noted in [24] one may consider inferences in which the precedent and consequent are combinations of disjunctive and conjunctive forms, such as $A \Rightarrow B$ where A is of the form $A_1 \wedge A_2 ... \wedge A_n$, and B is of the form $B_1 \vee B_2 ... \vee B_m$. Defining appropriate information theoretic quantities to rank words for capturing such inferences is an interesting open question.

1.1 Related work

Inference detection has been considered by many researchers. Unwanted inferences in databases has been studied in [23, 28, 6, 7]. In these works, it is shown that the knowledge about the database such as data schema, data dependencies, domain semantic and even actual data, can be used to derive inferences about other sensitive information stored in the database. To protect against such inferences, each user's query is analyzed in the context of the past queries to find possible inferences. Based on this analysis, the query is denied or accepted. These works do not take into account other public information. In [6] and [7] authors considered the web access history of the user as an input to detect unwanted inferences. However, this information was not taken into account in their algorithm.

Inferences have also been used to de-identify individuals ([26], [25] and [27]). In [26] and [25], authors developed methods to make inferences about the identity of an individual or other sensitive data, when data is structured. Examples of structured data are data from banks and hospitals. This is different from our work (and Staddon et al.) where no structure is assumed for the data. In [27], a tool is developed that can be used to infer an identity from a document using other public documents. The tool takes specific information, that is, SSN numbers, to make inferences about the birth year and status of an employee.

Other works related to redaction are [14] which presents a signature scheme that tolerates document changes due to redaction, and [17] and [18], where attacks on redacted documents using natural language processing or image and font analysis are presented.

The works in ([24] and [8]) have the same goal as ours: inference detection using the Web as public knowledge with the aim of providing better document redaction. Our work builds on these works and proposes a keyword selection methods that best matches the underlying model of inference (word co-occurrence).

1.1.1 Web-based inference detection

Web-based inference is first proposed in [24] as a semi-automated way for detecting unwanted inferences in a document when public knowledge encapsulated in the Web is taken into account.

Let $\mathcal{D}$ be the document that must be redacted. Authors use TF.IDF (see Section 1) to construct a set $\mathcal{K}(D)$ including the words that have the highest TF.IDF values in $\mathcal{D}$. The keywords in $\mathcal{K}(D)$ are then used to detect inferences by issuing queries on subsets of keywords to a search engine. If the top γ returned documents contain the sensitive keyword(s), the set of keywords are considered to be inference enabling. Extracting keywords using TF.IDF results in words with high frequency in $\mathcal{D}$ that almost do not appear in other documents in $\mathcal{C}$ to be included in $\mathcal{K}(D)$. This includes some high frequency words that are not related to the sensitive keywords. For example in Section 4, in the case of the document with the name Bin Laden redacted, general words such as "resource" and "support" are more frequent than specific words such as "US" and "September", although, the latter words are more important.

1.1.2 Detecting Privacy Leaks Using Corpus-based Association Rules

[8] only focuses on the inference detection technique but the results can be applied to the redaction problem we are considering here. (Authors also develop a redaction tool based on this algorithm [9]). Authors consider a collection (corpus) of documents $\mathcal{C}$ that is related to a set of sensitive keywords, and search for sets of words (consisting of a number of words), that co-occur frequently in the collection $\mathcal{C}$. An editor then selects item sets of interest which are frequently occurring word sets, and closely related to a specific privacy application. The selected sets are then represented as a list of candidate inferences, each inference is of the form $A \Rightarrow B$, where A and B are Boolean formulas of items, which means A is of the form $A_1 \wedge A_2 ... \wedge A_n$, and B is of the form $B_1 \vee B_2 ... \vee B_m$ with A_i is from an item set and B_j is from a set of sensitive keywords. They calculate a confidence level for each inference and when the confidence is above a threshold, the inference will become part of the final output.

Our experimental results show that the set of inferences produced by this approach is incomplete. Also it would be more effective if words in the private documents are used directly in the search for these predicted inferences (currently only the words in the corpus are considered). The algorithm needs to produce a set of all unwanted inferences based on the corpus, so the "knowledge" about the sensitive topic represented in the corpus should be sufficient. Generating a "good corpus" that results in all unwanted inferences presents major challenges.

2. PROBLEM STATEMENT

We first state the inference detection problem in an abstract way, and proceed to give concrete values.
A document is a set of words over an alphabet. We consider unstructured documents such as those that can be found on the Web, which can include other symbols including numbers and punctuation marks. In defining inference, we use an intuitive definition of knowledge and knowledge combination that has been used in [24]. The aim is to capture the

notion of *inference* in its most general form, without limiting oneself to a particular model or notion inference. Let $\mathcal{D}$ denote a private document that contains some sensitive information. Informally, let $K(\mathcal{D})$ denote the "knowledge" (or facts, or axioms) that can be extracted from $\mathcal{D}$ and assume that there are knowledge composition rules, which specify how to derive new knowledge from the combination of existing pieces of knowledge. (Here we use the term knowledge that is represented as $K(\mathcal{D})$ without a formal definition and only to represent intuition about this concept.)

Denote by $\bar{K}(\mathcal{D})$ the closure of $K(\mathcal{D})$ under the knowledge composition rules. The rules are abstract rules that are assumed to be applied repeatedly to the knowledge to generate/derive new knowledge. The closure of $K(\mathcal{D})$ is the closed set of all knowledge that can be obtained from $K(\mathcal{D})$ by repeated application of the composition rules. Before defining the redaction problem, we define the inference detection problem.

Inferences detection. Let $\mathcal{R}$ denote a collection of reference documents (such as public Web) that are related to a topic. The "knowledge" that can be computed from the union of the private and the reference collection, $\bar{K}(\mathcal{D}\bigcup\mathcal{R})$, is in general larger than $\bar{K}(\mathcal{D})\bigcup\bar{K}(\mathcal{R})$, which is the union of what can be extracted separately from $\mathcal{D}$ and $\mathcal{R}$. In the most general formulation, the inference detection problem is to find and understand the difference:

$$\delta(\mathcal{D},\mathcal{R}) = \bar{K}(\mathcal{D}\bigcup\mathcal{R}) - (\bar{K}(\mathcal{D})\bigcup\bar{K}(\mathcal{R}))$$

Document redaction. We are given a set $\mathcal{S}$ of sensitive keywords that the publication of $\mathcal{D}$ should not expose. Ideally, redaction of a document $\mathcal{D}$ with respect to the sensitive keyword set $\mathcal{S}$ and a reference document set $\mathcal{R}$, is finding a subset $\mathcal{D}^{sub}$ of $\mathcal{D}$ such that the intersection $\mathcal{S}\bigcap\delta(\mathcal{D}^{sub},\mathcal{R}) = \emptyset$. While $\mathcal{D}^{sub} = \emptyset$ trivially satisfies this condition, the goal is to have a subset $\mathcal{D}^{sub}$ of a "good" size in the sense that it preserves the usefulness of the original document while protecting $\mathcal{S}$.

2.1 Redaction system

We consider a redaction system that detects and removes unwanted inferences. The system consists of two major steps: (i) Inference detection and (ii) Breaking inferences by removing words. We focus on (i) and in particular keyword extraction. For completeness we also include an overview of the other part.

2.1.1 Inference detection algorithm

Input: Document $\mathcal{D}$, a sensitive keyword s_0, public Web documents.

Output: A list $\mathcal{L}$ of inferences, of the form:

$$(w_1, ..., w_k) \Rightarrow s_0$$

where $w_1, ..., w_k$ are keywords extracted from document $\mathcal{D}$. Here we only consider a single sensitive word.

Step 1. Preprocessing. Use a NLP (Natural Language Processing) tool to "clean" $\mathcal{D}$ and remove "stop" words. These are common words in the language that are unlikely to be related to the sensitive keyword. For example, in English "the", "an", "a", "do", are considered stop words. Our tool takes a list of previously defined stop words and removes them together with s_0 from $\mathcal{D}$ to form $\mathcal{D}_p$.

Step 2. Extract keywords from the document $\mathcal{D}_p$ with respect to s_0. $\mathcal{K}(\mathcal{D})$ consists of all the words in $\mathcal{D}_p$ whose mutual information with s_0 is greater than a threshold α.

The algorithm requires a corpus $\mathcal{C}$ of related documents to s_0. We construct a corpus $\mathcal{C}$ of documents related to s_0 by using a search engine to search for documents that contain s_0 on the public Web and then choose a subset of documents that appear more related.

For each word $w \in \mathcal{D}_p$, we calculate the mutual information between w and s_0 ($I(w, s_0)$) for each document in $\mathcal{C}$, and average the result over all documents in the corpus. Details are below.

We consider each paragraph in a document as a subdocument. Let X_w and X_{s_0} denote two binary variables that takes the value 1, if w and s_0 appear in a subdocument, respectively. We estimate the probability $\Pr(X_w = 1)$ by finding the number of paragraphs that contain w, divided by the total number of paragraphs in the document. A similar approach will be used to find distribution of $\Pr(X_{s_0})$ and also the joint distribution $\Pr(X_w, X_{s_0})$. The mutual information between X_w and X_{s_0} based on the corpus $\mathcal{C}$ is calculated, and the the top ℓ words with the highest mutual information are outputted.

```
SelectKeywords ($\mathcal{D}_p$, $s_0$)
– For each document $\mathcal{D}_j$ in $\mathcal{C}$
– Calculate $I_j(X_{w_i}; X_{s_0}) = H(X_{w_i})+H(X_{s_0})-H(X_{w_i}, X_{s_0})$ for each $w_i \in \mathcal{D}_p$
– Take $I_{w_i} = \frac{1}{n}\sum_{j=1}^{n} I_j(w_i; s_0)$
– Select the top $l$ words that have highest $I_w$
```

Here n is the number of documents in the corpus $\mathcal{C}$, $H(X)$ is the entropy of the random variable X, and $H(X_1, X_2)$ is the joint entropy of two random variables X_1 and X_2 :

$$H(X) = -\sum_{x\in X} p(x)log_2p(x)$$

$$H(X_1, X_2) = -\sum_{x_1\in X_1, x_2\in X_2} p(x_1, x_2)log_2p(x_1, x_2)$$

The probabilities of the random variables X_w and X_{s_0} can be estimated as relative frequencies in $\mathcal{C}$ as follows:

$$\Pr(X_{w_i} = 1, X_{s_0} = 1) = \frac{n_{w_i\wedge s_0}}{n_j}$$

$$\Pr(X_{w_i} = 1) = \frac{n_{w_i}}{n_j}$$

$$\Pr(X_{s_0} = 1) = \frac{n_{s_0}}{n_j}$$

where n_j is the number of subdocuments in the document $\mathcal{D}_j$, $n_{w_i\wedge s_0}$ is the number of subdocuments in $\mathcal{D}_j$ that contain both w_i and s_0, and n_{w_i}, n_{s_0} are the numbers of subdocuments in $\mathcal{D}_j$ contain w_i, s_0 respectively.

Step 3. Inference analysis. The list $\mathcal{L}$ of inferences is initially empty. We consider every subset $\mathcal{K}' \subseteq \mathcal{K}(\mathcal{D})$ of size $|\mathcal{K}'| = \beta$, $\mathcal{K}' = (w_1, ..., w_\beta)$, and do the following:

1. Issue queries on $(w_1, ..., w_\beta)$ to a search engine and use the top γ documents based on the search engine's rankings (γ is an integer > 0).

2. Find s_0 in the top γ documents, if s_0 appears, we add to $\mathcal{L}$ the inference $\mathcal{K}' \Rightarrow s_0$.

2.1.2 Breaking inferences by removing words.

A simple approach to secure redaction is to remove all words in the precedents of unwanted inferences in $\mathcal{L}$ from $\mathcal{D}$ to obtain $\mathcal{D}^{sub}$. However to increase the readability, one can use more refined methods to minimize the number of words that need to be removed to eliminate unwanted inferences. This is our future research.

2.2 Parameters of inference detection

Performance of the inference detection step depends on a number of parameters, (i) ℓ: the size of $\mathcal{K}(\mathcal{D})$, (ii) β which is the size of subsets that are used in queries, and (iii) γ that controls the search depth for documents in the public Web. These parameters can be tuned to achieve the required trade-offs between the computation cost of the inference detection and the strength of inferences (i.e., weak inferences are not considered in lieu of more efficient algorithms). Allowing larger ℓ, β and γ results in more costly algorithms but allows finer inferences (and probably weaker ones) to be detected.

Figure 5 shows that subsets of words (subsets of 2 words- in this experiment) are more likely to result in inferences than single words; however, the number of queries increases from $O(l)$ to $O(l^2)$ (l is the size of the keyword set)(Figure 6).

The depth of the search also affects the efficiency and effectiveness of the scheme. Our experiments in Section 3 shows with increasing depth, extra weaker inferences can be found with reasonable amount of extra computation (see Figure 7, Figure 8) .

2.3 Multiple Sensitive Keywords

We first give three algorithms that are direct applications of single word algorithm to multi sensitive word case. Our proposed algorithm as well as TF.IDF based algorithm of Staddon et al. both can be used for single sensitive word case. We next extend our information theoretic approach to define a combined metric for ranking of words in the document that takes into account the relationship among the sensitive words.

Suppose we are to redact a document with respect to a given list of sensitive keywords $s_1, s_2, ...s_u$.

Let Π denote the algorithm that is used for keyword extraction with respect to a single sensitive keyword. Π takes three inputs, a document $\mathcal{D}$, a sensitive keyword s_0, and the length ℓ of the keyword list that it needs to construct.

A direct solution when there are multiple sensitive keywords, is to repeat the above procedure for each sensitive keyword and take the union of the unwanted inferences obtained in each case.

The trivial algorithm below follows this approach.

```
TrivMultiWrdInfr ($\mathcal{D}, s_1 \cdots s_u; \ell$)
– For $i = 1, \cdots u$,
– Call $\Pi(\mathcal{D}, s_i, \ell) \rightarrow \mathcal{K}_{\mathtt{i}}(\mathcal{D})$
– Find $\mathtt{Infr}_i$, the set of inferences on $\mathcal{D}$ using $\mathcal{K}_{\mathtt{i}}(\mathcal{D})$
– Find $\mathtt{Infr} = \cup_{i=1}^{u} \mathtt{Infr}_i$ where $\mathtt{Infr}$ is the set of all inferences.
```

The algorithm will have u complete passes over the document and the Web-query phase for finding inferences. The redacted document will have many words (and some unnecessarily) removed which will reduce the quality of the output.

Merging lists. To reduce the number of Web searches, the keyword lists corresponding to sensitive words are first merged and then used for Web search. The two algorithms are the same, except **MergListMultiWrdInfr_2** considers the top ℓ elements of the merged list.

```
MergListMultiWrdInfr_1 ($\mathcal{D}, s_1 \cdots s_u; \ell$)
– For $i = 1, \cdots u$,
– Call $\Pi(\mathcal{D}, s_i, \ell/u) \rightarrow \mathcal{K}_{\mathtt{i}}(\mathcal{D})$
– Form $\mathcal{K}_{\mathtt{merg}}(\mathcal{D}) = \cup_{i=1}^{u} \mathcal{K}_{\mathtt{i}}(\mathcal{D})$
– Find $\mathtt{Infr}$, the set of inferences on $\mathcal{K}_{\mathtt{merg}}(\mathcal{D})$
```

```
MergListMultiWrdInfr_2 ($\mathcal{D}, s_1 \cdots s_u; \ell$)
– For $i = 1, \cdots u$,
– Call $\Pi(\mathcal{D}, s_i, \ell) = \mathcal{K}_{\mathtt{i}}(\mathcal{D})$
– Select the top $\ell$ words in $\cup_{i=1}^{u} \mathcal{K}_{\mathtt{i}}(\mathcal{D})$ to form $\mathcal{K}_{\mathtt{merg}}(\mathcal{D})$.
– Find $\mathtt{Infr}$, the set of inferences on $\mathcal{K}_{\mathtt{merg}}(\mathcal{D})$
```

In all above algorithms, relationship of a word w in the document is considered separately with each sensitive word, and so if a word w has strong relationship with combination of two or more sensitive words, it may not be included in the final keyword list. This shortcoming is addressed by extending our information theoretic measure. The advantage of these two latter algorithms compared to **TrivMultiWrdInfr** is efficiency, that is, they use Web query phase only once. We will show the efficiency and effectiveness of each algorithm in the experiments in Section 3.

Information theoretic approach. We generalize the keyword extraction to cater for multiple sensitive words. We consider two cases:

$s_1, \cdots, s_u$ **are independent.**

For each word $w_i \in \mathcal{D}$ and each sensitive keyword s_j, we find $I(w_i; s_j)$, and use the top ℓ words that have the highest sum of mutual information with the sensitive words as the keywords. In other words, we rank words in $\mathcal{D}$ based on the following quantity:

$$I_{w,s_1,\cdots,s_u} = I(w_i; s_1) + I(w_i; s_2) + ... + I(w_i; s_u)$$

This quantity effectively captures the strength of the relationship (in terms of co-occurrence) between w and the set of sensitive keywords: if there is one sensitive keyword in the set that has high value of $I(w; s_i)$, the sum will be high. Similarly, if mutual information is moderate with respect to a number of sensitive words, the total may be sufficient to move the word into the keyword list. Only if the mutual information with respect to all sensitive words is very small, the sum will be small and the word will not be included.

$s_1, \cdots, s_u$ **are not independent.**

We assume $s_1, \cdots, s_u$ are not independent if $I(s_i; s_j) \geq t$ for at least one pair: $i, j \in \{1, \cdots, u\}$ and $i \neq j$, where t is a threshold that needs to be determined for each application. Users can also explicitly specify s_i and s_j as related words.

Let $u = 2$. That is, there are two sensitive words that are related to each other and so $I(s_i; s_j) \geq t$. We use the following metric to quantify the strength of the relationship between w and (s_1, s_2):

$$I_{w,s_1,s_2} = I(w; s_1) + I(w; s_2) - I(w; s_1, s_2)$$

For the general case, suppose that every two words in u sensitive words are related to each other and $I(s_i; s_j) \geq t$ for all $i, j \in \{1, \cdots, u\}$ and $i \neq j$. We then have:

$$I_{w,s_1,\cdots,s_u} = \sum_{i=1}^{u} I(w;s_i) - \sum_{i,j=1,i\neq j}^{u} I(w;s_i,s_j) + \sum_{i,j,m=1,i\neq j\neq m}^{u} I(w;s_i,s_j,s_m) - \sum_{i,j,m,n=1,i\neq j\neq m\neq n}^{u} I(w;s_i,s_j,s_m,s_n) + \cdots$$

To calculate mutual information of more than two random variables, one can use the chain rule for mutual information as described in [11]:

$$I(X_1,\cdots,X_n;Y) = \sum_{i=1}^{n} I(X_i;Y|X_{i-1},X_{i-2},\cdots,X_1)$$

The algorithm to extract ℓ keywords from the document following this approach will be as follows.

CumulMultiWrdInfr $(\mathcal{D}, s_1, \cdots s_u; \ell)$
– For each word w_i in $\mathcal{D}$,
– Calculate $I_{w,s_1,\cdots,s_u}$
– To form the keyword list $\mathcal{K}(\mathcal{D})$, find the top ℓ words in $\mathcal{D}$ with the highest $I_{w,s_1,\cdots,s_u}$.
– Find `Infr`, the set of inferences on $\mathcal{D}$ using $\mathcal{K}(\mathcal{D})$.

The algorithm is one pass and so uses Web query phase only once. For detailed comparisons, see Section 3.2.3.

2.4 Evaluating Web-based inference detection systems

Web-based inference detection systems extract a *keyword list* $\mathcal{K}(\mathcal{D})$ from a document and result in a list $\mathcal{L}^{\mathtt{pre}}$ that contain precedents of the found unwanted inferences.

We first define *the set of all inferences of a document* $\mathcal{D}$ as follows:
Definition 1: The set of all inferences $\mathtt{A}$ of a document $\mathcal{D}$ with a parameter β and γ (as described in Section 2.1) consists of the inferences that are found by querying all subsets of β words in $\mathcal{D}_p$. For each search result, the returned documents are analyzed (as in the *Inference analysis* phase) and the queried subset is detected as precedent of an inference, if the test is passed.

Important aspects of a Web-inference detection algorithm for the purpose of redaction are:

1. Effectiveness of inference detection: this is measured by the parameter ρ defined as the percentage of all inferences detected by using a specific algorithm. For a fixed document, ρ is a function of, (i) $\mathcal{K}(\mathcal{D})$, (ii) query algorithm parameters: β, γ, the ranking algorithm of the search engine, and the inference analysis stage.

2. Efficiency: This consists of:
(i) Computational complexity of finding $\mathcal{K}(\mathcal{D})$;
(ii) Query efficiency of the algorithm: is the number of queries and is measured as $\binom{|\mathcal{K}(\mathcal{D})|}{\beta}$;
(iii) Computational complexity of inference analysis stage: that depends on the number of returned pages (in all queries) and the specific algorithm that is used to analyze these pages to detect inferences.

Both our algorithm and [24] require the same computation cost for extracting keywords, that is $O(m)$ where m is the number of words in the private document, and for the inference analysis phase that is $O(\gamma\binom{|\mathcal{K}(\mathcal{D})|}{\beta})$.

3. EXPERIMENTS

We developed a tool set written in Java, and performed the experiments reported in [24] and also new experiments required when employing our proposed keyword extraction algorithm.

We did two sets of experiments: a single sensitive keyword and multiple sensitive keywords. The first experiment is for redacting a record about "Bin Laden" and the aim is to prevent inference of "Bin Laden" when the redacted document is combined with the documents on the public Web. The second experiment is about redacting all sensitive information related to a patient's diseases before the document is released. We limited our search to inferences with precedents consisting of two words (pairs of words used for issuing queries).

3.1 Experimental challenges and tools

Ideally, the approach needs to be tested on real documents and related corpus. For example, for protecting a document against inferences made for "Bin Laden", we need a corpus of FBI documents that are related to "Bin Laden", so that a correct estimate of probabilities can be calculated. However, such a corpus is hard to obtain. We used instead publicly available information about the sensitive keywords under consideration ("Bin Laden" in our first experiment, "anxiety", "depression" in our second experiment). We built 3 different corpora for algorithm's stability test. Each corpus consists of 30 documents and are mostly from Wikipedia pages. These documents are in html or text format. For html documents, the text is extracted from html. All the stop words and sensitive words are then removed from those documents and a list of keywords are identified.

All of our experiments use the tool set to remove stop words and extract the keyword list. Our code for extracting text from html uses standard techniques for removing html tags. We used Bing Search API [5] to make queries using keywords. Bing Search API is an API that allows us to issue queries automatically to the search engine of Bing.

3.2 Experiment description

3.2.1 Redaction of a Military document: one sensitive keyword

The experiment takes as input the document that must be redacted, in this case, a page about Bin Laden [4]. The page is then anonymized and the name and aliases of the person are removed. To identify words that might allow "Bin Laden" to be inferred, we issued queries on subsets of words and examined top 5 returned pages, and detected an inference if at least one document included the word "Bin Laden".

The following describes our experiment in more details.
Input: A plaintext file of the article about Bin Laden [4].

1. Remove the subject "Osama Bin Laden" and aliases from the text.

2. Extract the top ℓ words (ℓ varies from 20 to 100) from the text following the algorithm **SelectKeywords**. This forms the set $\mathcal{S}_{\mathcal{B}}$.

3. Issue queries on pairs of keywords from the set $\mathcal{S}_{\mathcal{B}}$. Select the top 5 pages returned by Bing Search. We considered only the hits consist of html or text.

4. We add that pair of keywords to a set of precedents of inferences $\texttt{Infr}_1^{\texttt{pre}}$ if it passes the test in the *Inference analysis stage.*

Output: Set of precedents of inferences $\texttt{Infr}_1^{\texttt{pre}}$.

We compared this algorithm with the results obtained from the algorithms in [24] and [8]. For the algorithm [24], we used exactly the above steps, except in step 2 used TF.IDF technique as described in [24]. This results in a different set of precedents of inferences $\texttt{Infr}_2^{\texttt{pre}}$. The table 1 presents the keyword lists extracted using two keyword extraction algorithms.

The experiments showed that keyword pairs from our algorithm generated more inferences for "Bin Laden", with the same efficiency (both algorithms used the same number of queries, and required the same computation cost for extracting keywords and inference analysis phase). This is expected as our algorithm extracts in the document a list of keywords related to "Bin Laden" and not general high frequency words (see Figure 2- our algorithm constantly gives better results with different sizes of the keyword set). As the result, the redacted document produced by our algorithm contains less inferences about "Bin Laden" than the redacted one produced by the previous work.

As described in Section 1.1.2, the algorithm [8] outputs a set of candidate inferences based on a corpus. In the experiment, we followed the algorithm and outputted a set of candidate inferences $\texttt{CandInfr}^{\texttt{pre}}$.

The experiment showed that approximately 95% of sensitive inferences in the private document were not detected by the set $\texttt{CandInfr}^{\texttt{pre}}$. The corpus should contain sufficient "knowledge" related to the sensitive topic so that the algorithm can perform effectively (outputs a set $\texttt{CandInfr}^{\texttt{pre}}$ that captures more sensitive inferences) (see Section 1.1.2).

We changed the range of the parameters to examine the affect on the performance of the system. Our results show that, as expected, the number of inferences increases as the number of extracted words increases (see Figure 3). This increase however will plateau when the number of extracted words gets close to the number of words in the document.

3.2.2 Different corpora experiment.

The quality of the corpus (how much the corpus is related to the sensitive words) affects the effectiveness of the scheme. We built up three corpora with different relevance to the sensitive words. The first corpus was selected to make sure that the contents of the documents were not duplicated and they were related to the sensitive words. This corpus could be referred as a quality corpus with the aid of human. To create two other corpora, we used the sensitive words to query a search engine. The second and the third corpus were the documents 1-30 and 31-60, respectively, returned by that search. No further processing was used on these two corpora. We expected the second corpus to have lower quality. The effect of corpus is shown in the Figure 4. Our proposed keyword selection scheme behaved fairly stably when the quality of corpus varied. The second corpus which consisted of the top 30 pages returned by the search engine, still can be considered as a good one: the documents in this corpus were all about the sensitive topic. The result of using this corpus was slightly different from the first one. The third corpus contained some irrelevant documents thus resulted in lower effectiveness on average.

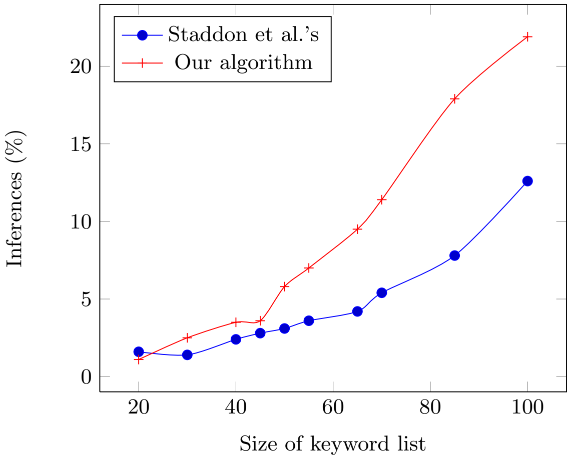


Figure 2: Comparisons between our algorithm and [24] when ℓ changes.

3.2.3 Medical record redaction experiment: multiple sensitive keywords

In this experiment, the goal is to redact a medical record that is about a patient who suffers a mental illness [16]. Suppose he has to publish his medical record to a third party but does not want to reveal his health status. In this experiment, the sensitive keywords that need to be protected are "anxiety", "depression" (which are marked as related to each other). The detailed procedure is below.

For each algorithm *TrivMultiWrdInfr*, *MergMultiWrdInfr_1*, *MergMultiWrdInfr_2* and *CumulMultiWrdInfr*:
Input: A plaintext file of the medical record [16].

1. Remove "anxiety", "depression" from the article.

2. Extract words from the document as described in the algorithm with $\ell = 50$. A keyword set $\mathcal{S_B}'$ is produced.

3. We issued queries on every two words of the set $\mathcal{S_B}'$ to the Bing Search engine. Select the top 5 pages returned by that search. We considered only the hits consist of html or text.

4. We add that pair of keywords to a set of precedents of inferences $\texttt{Infr}'^{\texttt{pre}}$ if it passes the test in the *Inference analysis stage.*

Output: set of precedents of inferences $\texttt{Infr}'^{\texttt{pre}}$.

Table 2 compares the results of the 4 algorithms. The algorithm *TrivMultiWrdInfr* detects more inferences, and has higher query cost (also requires more computational cost for inference analysis stage). This is due to *TrivMultiWrdInfr* operates on two keyword lists separately, one for each sensitive word. With the same number of queries *CumulMultiWrdInfr* could result in more than 60% inferences found. The remaining 3 algorithms have the same query cost. Among these, *CumulMultiWrdInfr* results in the most sensitive inferences found. We note that finding $\mathcal{K}(\mathcal{D})$ in *CumulMultiWrdInfr* is more complex than the other algorithms, but the extra cost is negligible.

Input: The article Osama Bin Laden and the Al Qaeda group [4]

Keywords extracted using our algorithm: saudi, world, US, arabia, afghanistan, soviet, terrorist, government, group, September, children, states, born, first, family, construction, 1957, network, time, islamic, muslim, joined, bombing, afghan, country, international, 2001, troops, pakistan, middle, 10, father, resistance, union, university, king, alqaeda, holy, iraq, mohammed, organization,...

Keywords extracted using algorithm [24]: groups, mohammad, mak, alqaeda, azzam, vast, alias, campaign, ideology, kashmir, channeled, islamic, cells, reach, islami, abu, resources, support, rulers, algeria, leader, membership, principal, hamas, organizations, le, jane, domestic, broad, turki, closely, banks, dr, phil, drawn, gunaratna, expertise, indonesia, kosovo, morocco, tan, mentor, hirschkorn, combat, fighting, inline, lebanon, fight,...

Keywords	Links
joined alqaeda	http://online.wsj.com/article/SB10001424052748704107204575038674123215854.html
terrorist iraq	http://www.weeklystandard.com/Content/Public/Articles/000/000/006/550kmbzd.asp
bombing pakistan	http://news.yahoo.com/s/ap/as_pakistan
soviet mohammed	http://www.terroristplanet.com/mullahomar.htm

Table 1: Example of keywords extracted using our algorithm and [24]. In the smaller box, the left column is examples of pairs of words that we used to issue queries, and the right column is the resulted pages from the queries that contain sensitive content.

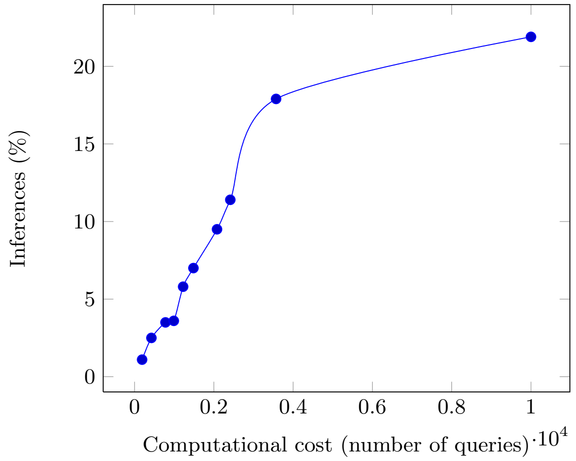


Figure 3: Effectiveness vs. efficiency of the scheme while increasing ℓ

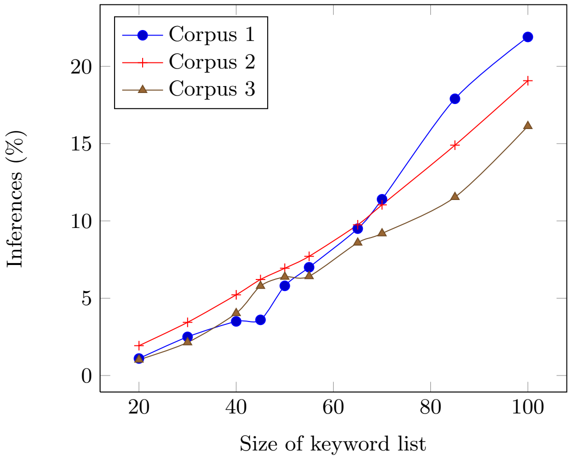


Figure 4: Effectiveness of the scheme with different corpora

3.3 Performance analysis

The running time of the system mainly depends on the number of queries that need to be issued and the stability of the network. Concretely, if we extract from the input document 100 keywords, the query step using Bing Search for each experiment requires around 8 hours to complete. This is due the total number of keyword pairs in each experiment is 4950, and also in most of the running time, Bing Search has network response delays which are sometimes some minutes for one query. Interestingly, the other modules of the system take only 30 seconds to finish all the tasks.

4. CONCLUSION AND FUTURE WORK

We proposed an information theoretic approach to document keyword extraction with respect to a single sensitive word and extended it to the case of multiple sensitive words. Our approach is a natural way of extracting keywords when inferences are modeled as co-occurrences. We showed superior performance of the proposed approach in detecting unwanted inferences compared to [24]. The novelty and significance of our work is in defining appropriate information theoretic measures for ranking document words with respect to a corpus, that captures their potentials to result in unwanted inferences. The keyword list can be seen as a list of words in the target document that give strong direct (single word) inference of the form $w \Rightarrow s_0$ about the sensitive keyword(s), and the follow on stage of Web search is to find subsets of more than one keywords that give new unwanted inferences.

Using inferences in the context of redaction has other challenges related to the overall performance of the system. It is always easy to remove many words from the document to reduce unwanted inferences. However this will reduce readability and usefulness of the redacted document. One may also use complex NPL analysis and human experts to remove inferences. The result will be a costly process that cannot scale to the large volume of documents that must be released, for example by states and in response to the Freedom of Information Legislations. Our theoretical approach opens possible directions for developing automated redaction systems that do not leak unwanted inferences and maintain high readability of the redacted document.

5. REFERENCES

[1] E. Bier, L. Good, K. Popat, and A. Newberger. A document corpus browser for in-depth reading. In *Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '04, pages 87–96, New York, NY, USA, 2004. ACM.

[2] E. A. Bier and E. W. Ishak. Entity quick click: rapid text copying based on automatic entity extraction. In *Abstracts of the Conference on Human Factors in Computing Systems (CHI*, pages 562–567. ACM Press, 2006.

[3] E. A. Bier and E. W. Ishak. Entity workspace: an evidence file that aids memory, inference, and reading. In *Proceedings of Intelligence and Security Informatics (ISI 2006*, pages 466–472. Springer-Verlag, 2006.

[4] R. o. Bin Laden. http://www.webspawner.com/users/islamicjihad15, Aug. 2001.

[5] Bing-API. www.bing.com/toolbox/bingdeveloper, 2012.

[6] Y. Chen and W. W. Chu. Database security protection via inference detection. In *IEEE International Conference on Intelligence and Security Informatics*, 2006.

[7] Y. Chen and W. W. Chu. Protection of database security via collaborative inference detection. *IEEE Trans. on Knowl. and Data Eng.*, 20:1013–1027, August 2008.

[8] R. Chow, P. Golle, and J. Staddon. Detecting privacy leaks using corpus-based association rules. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 893–901, New York, NY, USA, 2008. ACM.

[9] R. Chow, I. Oberst, and J. Staddon. Sanitization's slippery slope: the design and study of a text revision assistant. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, SOUPS '09, pages 13:1–13:11, New York, NY, USA, 2009. ACM.

[10] P. Cimiano and S. Staab. Learning by googling. *SIGKDD Explor. Newsl.*, 6:24–33, December 2004.

[11] T. M. Cover and J. A. Thomas. *Elements of information theory.* Wiley-Interscience, New York, NY, USA, 1991.

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

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

[14] S. Haber, Y. Hatano, Y. Honda, W. Horne, K. Miyazaki, T. Sander, S. Tezoku, and D. Yao. Efficient signature schemes supporting redaction, pseudonymization, and data deidentification. In *Proceedings of the 2008 ACM symposium on Information, computer and communications security*, ASIACCS '08, pages 353–362, New York, NY, USA, 2008. ACM.

[15] M. Koppel, J. Schler, S. Argamon, and E. Messeri. Authorship attribution with thousands of candidate authors. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 659–660, New York, NY, USA, 2006. ACM.

[16] linden method.com. http://www.linden-method.com/medical-records/, 1993.

[17] D. Lopresti and A. L. Spitz. Quantifying information leakage in document redaction. In *Proceedings of the 1st ACM workshop on Hardcopy document processing*, HDP '04, pages 63–69, New York, NY, USA, 2004. ACM.

[18] D. Lopresti, A. L. Spitz, D. Lopresti, and A. L. Spitz. Information leakage through document redaction: Attacks and countermeasures. In *In DRR*, pages 183–190, 2004.

[19] C. D. Manning and H. Schütze. *Foundations of statistical natural language processing.* MIT Press, Cambridge, MA, USA, 1999.

[20] T. M. Mitchell. *Machine learning.* McGraw Hill, New York, 1997.

[21] C. E. Shannon and W. Weaver. *A Mathematical Theory of Communication.* University of Illinois Press, Champaign, IL, USA, 1963.

[22] Slashdot.org. Anonymity of netflix prize dataset broken, 2007.

[23] D. L. Spooner, S. A. Demurjian, and J. E. Dobson, editors. *Proceedings of the ninth annual IFIP TC11 WG11.3 working conference on Database security IX : status and prospects: status and prospects*, London, UK, UK, 1996. Chapman & Hall, Ltd.

[24] J. Staddon, P. Golle, and B. Zimny. Web-based inference detection. In *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, pages 6:1–6:16, Berkeley, CA, USA, 2007. USENIX Association.

[25] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10:571–588, October 2002.

[26] L. Sweeney. k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10:557–570, October 2002.

[27] L. Sweeney. Ai technologies to defeat identity theft vulnerabilities. *AAAI Spring Symposium, AI Technologies for Homeland Security*, 2005.

[28] R. Yi and K. Levitt. Data level inference detection in database systems. In *Proceedings of the 11th IEEE workshop on Computer Security Foundations*, pages 179–, Washington, DC, USA, 1998. IEEE Computer Society.

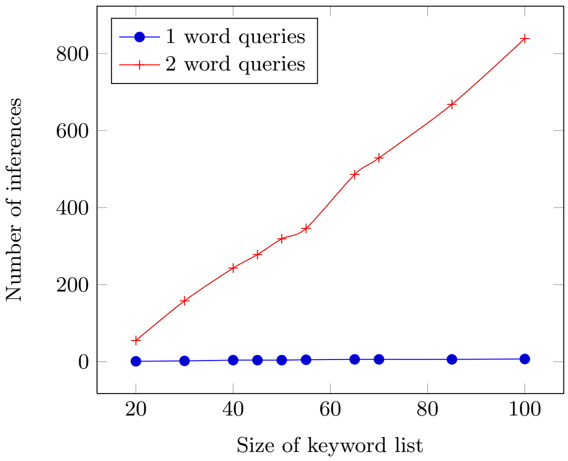


Figure 5: Effectiveness of the scheme with different values of β

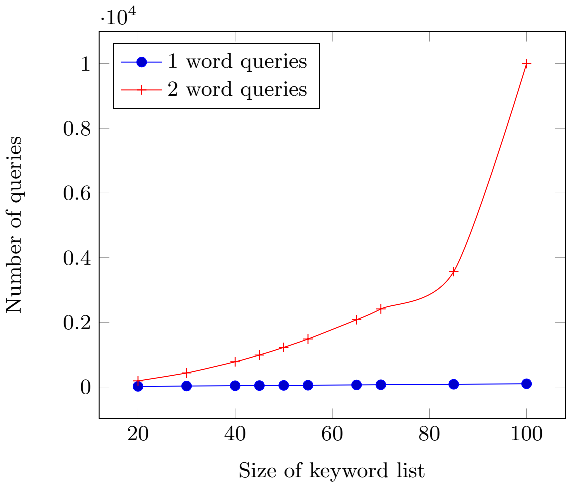


Figure 6: Efficiency of the scheme with different values of β (in number of queries)

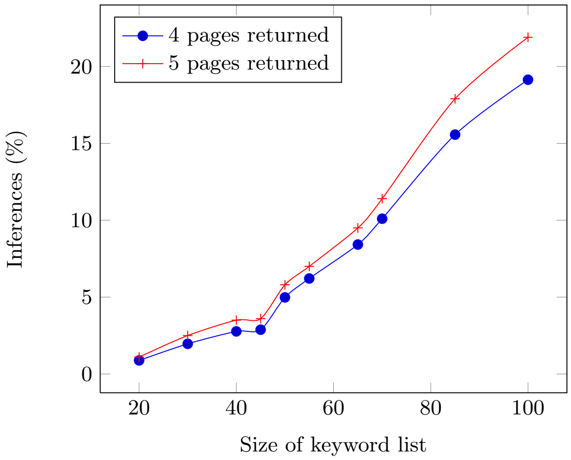


Figure 7: Effectiveness of the scheme with different values of γ

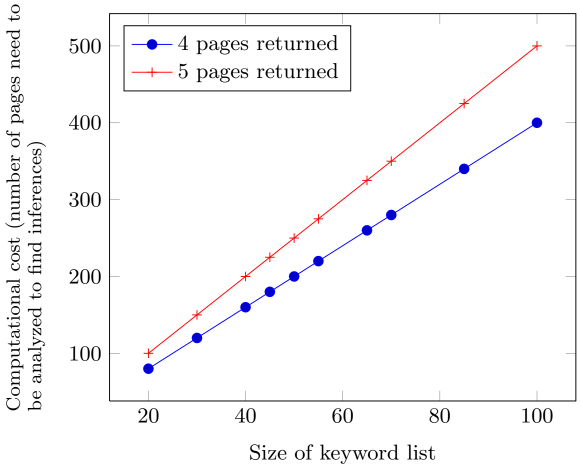


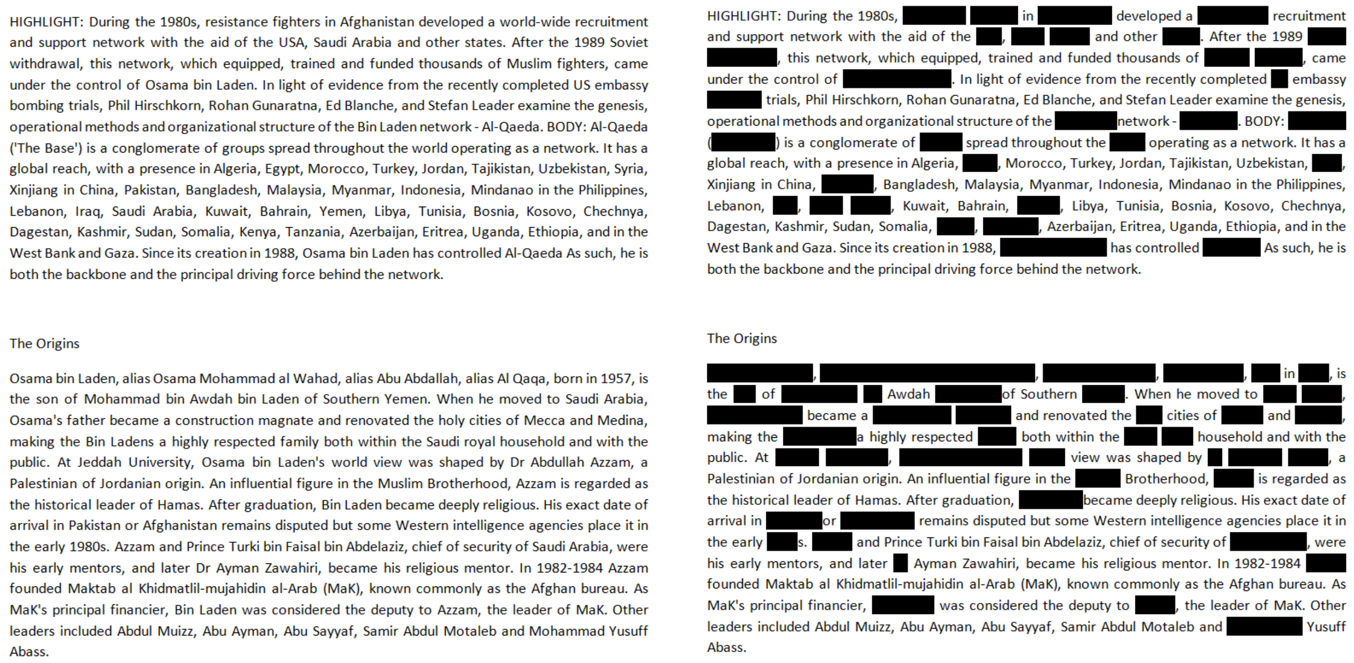
Figure 8: Efficiency of the scheme with different values of γ

	Effectiveness	Finding $\mathcal{K}(\mathcal{D})$	Query efficiency	Inference analysis
TrivMultiWrdInfr	45.46%	$O(u\vert\mathcal{D}\vert)$	11175	$O(\gamma\binom{\vert\mathcal{K}(\mathcal{D})\vert}{\beta})$
MergMultiWrdInfr_1	18.77%	$O(u\vert\mathcal{D}\vert)$	1225	$O(\gamma\binom{\vert\mathcal{K}(\mathcal{D})\vert}{\beta})$
MergMultiWrdInfr_2	18.89%	$O(u\vert\mathcal{D}\vert)$	1225	$O(\gamma\binom{\vert\mathcal{K}(\mathcal{D})\vert}{\beta})$
CumulMultiWrdInfr	19.89%	$O(u^2\vert\mathcal{D}\vert)$	1225	$O(\gamma\binom{\vert\mathcal{K}(\mathcal{D})\vert}{\beta})$

Table 2: Comparison results of 4 multiword algorithms

APPENDIX

The redacted document generated by our algorithm is shown in the figure below.



HIGHLIGHT: During the 1980s, resistance fighters in Afghanistan developed a world-wide recruitment and support network with the aid of the USA, Saudi Arabia and other states. After the 1989 Soviet withdrawal, this network, which equipped, trained and funded thousands of Muslim fighters, came under the control of Osama bin Laden. In light of evidence from the recently completed US embassy bombing trials, Phil Hirschkorn, Rohan Gunaratna, Ed Blanche, and Stefan Leader examine the genesis, operational methods and organizational structure of the Bin Laden network - Al-Qaeda. BODY: Al-Qaeda ('The Base') is a conglomerate of groups spread throughout the world operating as a network. It has a global reach, with a presence in Algeria, Egypt, Morocco, Turkey, Jordan, Tajikistan, Uzbekistan, Syria, Xinjiang in China, Pakistan, Bangladesh, Malaysia, Myanmar, Indonesia, Mindanao in the Philippines, Lebanon, Iraq, Saudi Arabia, Kuwait, Bahrain, Yemen, Libya, Tunisia, Bosnia, Kosovo, Chechnya, Dagestan, Kashmir, Sudan, Somalia, Kenya, Tanzania, Azerbaijan, Eritrea, Uganda, Ethiopia, and in the West Bank and Gaza. Since its creation in 1988, Osama bin Laden has controlled Al-Qaeda As such, he is both the backbone and the principal driving force behind the network.

The Origins

Osama bin Laden, alias Osama Mohammad al Wahad, alias Abu Abdallah, alias Al Qaqa, born in 1957, is the son of Mohammad bin Awdah bin Laden of Southern Yemen. When he moved to Saudi Arabia, Osama's father became a construction magnate and renovated the holy cities of Mecca and Medina, making the Bin Ladens a highly respected family both within the Saudi royal household and with the public. At Jeddah University, Osama bin Laden's world view was shaped by Dr Abdullah Azzam, a Palestinian of Jordanian origin. An influential figure in the Muslim Brotherhood, Azzam is regarded as the historical leader of Hamas. After graduation, Bin Laden became deeply religious. His exact date of arrival in Pakistan or Afghanistan remains disputed but some Western intelligence agencies place it in the early 1980s. Azzam and Prince Turki bin Faisal bin Abdelaziz, chief of security of Saudi Arabia, were his early mentors, and later Dr Ayman Zawahiri, became his religious mentor. In 1982-1984 Azzam founded Maktab al Khidmatlil-mujahidin al-Arab (MaK), known commonly as the Afghan bureau. As MaK's principal financier, Bin Laden was considered the deputy to Azzam, the leader of MaK. Other leaders included Abdul Muizz, Abu Ayman, Abu Sayyaf, Samir Abdul Motaleb and Mohammad Yusuff Abass.

HIGHLIGHT: During the 1980s, [redacted] [redacted] in [redacted] developed a [redacted] recruitment and support network with the aid of the [redacted], [redacted] [redacted] and other [redacted]. After the 1989 [redacted] [redacted], this network, which equipped, trained and funded thousands of [redacted] [redacted], came under the control of [redacted]. In light of evidence from the recently completed [redacted] embassy [redacted] trials, Phil Hirschkorn, Rohan Gunaratna, Ed Blanche, and Stefan Leader examine the genesis, operational methods and organizational structure of the [redacted]network - [redacted]. BODY: [redacted] ([redacted]) is a conglomerate of [redacted] spread throughout the [redacted] operating as a network. It has a global reach, with a presence in Algeria, [redacted], Morocco, Turkey, Jordan, Tajikistan, Uzbekistan, [redacted], Xinjiang in China, [redacted], Bangladesh, Malaysia, Myanmar, Indonesia, Mindanao in the Philippines, Lebanon, [redacted], [redacted] [redacted], Kuwait, Bahrain, [redacted], Libya, Tunisia, Bosnia, Kosovo, Chechnya, Dagestan, Kashmir, Sudan, Somalia, [redacted], [redacted], Azerbaijan, Eritrea, Uganda, Ethiopia, and in the West Bank and Gaza. Since its creation in 1988, [redacted] has controlled [redacted] As such, he is both the backbone and the principal driving force behind the network.

The Origins

[redacted], [redacted], [redacted], [redacted], [redacted] in [redacted], is the [redacted] of [redacted] [redacted] Awdah [redacted]of Southern [redacted]. When he moved to [redacted] [redacted], [redacted] became a [redacted] [redacted] and renovated the [redacted] cities of [redacted] and [redacted], making the [redacted]a highly respected [redacted] both within the [redacted] [redacted] household and with the public. At [redacted] [redacted], [redacted] [redacted] view was shaped by [redacted] [redacted] [redacted], a Palestinian of Jordanian origin. An influential figure in the [redacted] Brotherhood, [redacted] is regarded as the historical leader of Hamas. After graduation, [redacted]became deeply religious. His exact date of arrival in [redacted]or [redacted] remains disputed but some Western intelligence agencies place it in the early [redacted]s. [redacted] and Prince Turki bin Faisal bin Abdelaziz, chief of security of [redacted], were his early mentors, and later [redacted] Ayman Zawahiri, became his religious mentor. In 1982-1984 [redacted] founded Maktab al Khidmatlil-mujahidin al-Arab (MaK), known commonly as the Afghan bureau. As MaK's principal financier, [redacted] was considered the deputy to [redacted], the leader of MaK. Other leaders included Abdul Muizz, Abu Ayman, Abu Sayyaf, Samir Abdul Motaleb and [redacted] Yusuff Abass.

Figure 9: The left hand side shows the document [4] before being redacted. The right hand side shows an example of a redacted document of [4] using our algorithm where the back rectangles represent redacted words recommended by our algorithm.