On the effectiveness of Anonymizing Networks for Web Search Privacy

Sai Teja Peddinti
Computer Science and Engineering
Polytechnic Institute of New York University
psaiteja@cis.poly.edu

Nitesh Saxena
Computer Science and Engineering
Polytechnic Institute of New York University
nsaxena@poly.edu

ABSTRACT

Web search has emerged as one of the most important applications on the internet, with several search engines available to the users. There is a common practice among these search engines to log and analyse the user queries, which leads to serious privacy implications. One well known solution to search privacy involves issuing the queries via an anonymizing network, such as Tor, thereby hiding one's identity from the search engine. A fundamental problem with this solution, however, is that user queries are still obviously revealed to the search engine, although they are "mixed" among the queries issued by other users of the same anonymization service.

In this paper, we consider the problem of identifying the queries of a user of interest (UOI) within a pool of queries received by a search engine over an anonymizing network. We demonstrate that an adversarial search engine can extract the UOI's queries, when it is equipped with only a short-term user search query history, by utilizing only the query content information and off-the-shelf machine learning classifiers. More specifically, by treating a selected set of 60 users – from the publicly-available AOL search logs – as the users of interest performing web search over an anonymizing network, we show that each user's queries can be identified with 25.95% average accuracy, when mixed with queries of 99 other users of the anonymization service. This average accuracy drops to 18.95% when queries of 999 other users of the anonymization service are mixed together. Though the average accuracies are not so high, our results indicate that few users of interest could be identified with accuracies as high as 80–98%, even when their queries are mixed among queries of 999 other users. Our results cast serious doubts on the effectiveness of anonymizing web search queries by means of anonymizing networks.

Categories and Subject Descriptors

C.2.0 [**Computer-Communications Networks**]: General—*Security and protection*; K.4.1 [**Computers and Society**]: Public Policy Issues—*Privacy*

Keywords

Web Search Privacy, Anonymizing Networks, Query Obfuscation

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.
ASIACCS '11, March 22–24, 2011, Hong Kong, China.
Copyright 2011 ACM 978-1-4503-0564-8/11/03 ...$10.00.

1. INTRODUCTION

Today's world wide web hosts an enormous amount and a wide variety of data. Efficiently searching and retrieving this vast amount of information is very important, and currently more than one search engine is available to the users. To improve their search results, these search engines adopted the practice to log and analyse the queries issued by the user. This received considerable attention from media and public as well as researchers all over the world because of the possible privacy breaches. The issue was first brought into limelight in August 2005, when the US Department of Justice issued a subpoena to Google for a week's worth of search query records [14]. Later AOL has published three months (pseudonymized) search query logs, from which identities of certain users had been extracted based on personal information embedded in their queries [8, 2]. Since then, the media started shedding more light on how several major search engines (Yahoo!, AOL, MSN and Google) log, store and analyse individual search query logs.

Archiving and analysing search queries is important from a search engine's perspective for improving the quality of search results, and for generating revenue through sponsored search advertising. However, this logging of queries has serious privacy implications which can be categorized into *explicit* and *implicit* versions. *Explicit* privacy breach happens because of the information embedded in the query itself, and some common examples include searching for a particular disease the user or a family member might be suffering from, searching for one's social security number to check if it exists on the web, and performing "ego-surfing[1]". *Implicit* privacy violations happen when the sensitive information can not be learned directly from the query logs, but has to be extracted using aggregation and profiling methods or data mining techniques. An apt example could be to infer the income level of a user by keeping track of the brand of products he/she often searches for[23].

Many techniques have been proposed to address this problem of privacy breach through web search queries. First class of solutions involves use of private information retrieval (PIR) protocols [11], which are a generic body of work. However, current PIR protocols, due to their high communication and computation overload, are not feasible to be deployed in practice with the existing infrastructure. A second class of solutions is based on the principle of *query obfuscation* [24], where a client-side software injects *noisy* queries into the stream of real queries transmitted to the search engine. These methods protect the user against profiling, thereby preventing implicit privacy violations. Unfortunately, a practical query obfuscation tool - TrackMeNot [10, 22], has recently been shown to be vulnerable [15]; an adversarial search engine can distinguish between user's queries and obfuscation queries with high accuracy,

[1]A common practice seen among users to search for their own names, just to check what results appear.

and can extract a large fraction of user's queries.

Another class of solutions involves the use of third-party infrastructure such as a single proxy, e.g., Scroogle [18] or an anonymizing network, e.g., Tor [21]. The use of single proxy is problematic because it requires the users to impose (unwanted) trust on to a single server hosted by a third-party company. Web search over an anonymizing network, which is the focus of our paper, certainly provides better protection and fault-tolerance than the use of a single proxy. An anonymizing network is typically implemented using onion routing and it involves routing one's queries over a path consisting of a series of nodes (called relay servers) distributed all over the internet. This way the actual source of a query would potentially remain hidden from the search engine. Private Web Search (PWS) [17] is a client side tool that can be used to route search queries privately over the Tor network. It has been mentioned in [17] that search queries, even though stripped of the accompanying information like IP address and cookies, reveal some information about the user. They associate it to the linkability among queries, which has been mentioned as an open problem in Table 1 in [17]. We try to address this open problem of linking queries by using machine learning techniques and show that queries of a user can be identified with reasonable accuracies, just by looking at the query content.

Our Contributions.

A higher level goal of this work is to analyse how effective an anonymizing network can be – in preserving users' privacy in practice – against an adversarial search engine. (From here on, we will call the anonymizing network as Tor, for simplicity of presentation and without loss of generality.) We observe that the web search over Tor network has one fundamental drawback: the search query has to reach the search engine in clear text format for the search engine to be able to process the query and return the response back to the user. However, these queries are indeed "mixed" among the queries issued by other users of the same anonymization service.

We ask the following question: *Is it possible for an adversarial search engine to associate queries coming out of Tor exit nodes to Tor users who issued these queries?*

In our attack model, the search engine is a passive adversary who tries to make identification decisions only by analysing the received and logged queries. In an attempt to keep our attacks generic, we assume that the search engine does not make use of any information associated with the queries besides the queries themselves, such as the Tor exit node IP address or even exact query timestamps. We base our work on an important observation, made by other researchers [4], that although a potentially large number of users might be accessing web search over the Tor network, only a small fraction of these users really remain anonymous to the search engine in practice. The reason is that a significant number of users, even while using Tor, remain logged in using their accounts with search engines (e.g., Gmail accounts with Google) and may not disable cookies and other identifying information [4]. This implies that a user's queries might be getting mixed among queries of only a small number of other (anonymous) Tor users, potentially making these queries more identifiable.

Indeed, we answer the aforementioned question affirmatively. More specifically, by treating a selected set of 60 users – from the publicly-available AOL search logs – as 'users of interest' performing web search over an anonymizing network, we show that their queries can be identified with 25.95% average accuracy, when queries of upto 99 other Tor users are mixed together and this average accuracy drops to 18.95% when queries of 999 other Tor users are mixed together. For few users, these accuracies reached upto 80–98%, even when queries of upto 999 'other users' (other Tor users) are mixed together. Our results cast serious doubt on the effectiveness of anonymizing web search queries by means of anonymizing networks.

In the rest of this paper, we denote the AOL users, whose queries are to be separated from the mixed query set received at Tor exit nodes as 'users of interest' and the total number of users using web search over Tor as $\mathbb{U}$. The users present in $\mathbb{U}$, apart from the users of interest, will be referred as 'other users'.

2. PROBLEM FORMULATION AND STUDY METHODOLOGY

In this paper, we investigate if it is possible for an adversarial search engine to associate queries coming out of Tor exit nodes to Tor users. While using any anonymizing network, the search query must reach the search engine in clear format so we base our work on the query information alone, which is always going to be revealed to the search engine. We assume that the search engine does not make use of any accompanying information like exit node IP address, unblocked cookies accompanying the queries or the Click-through patterns followed by the user. Specifically, in our attack model, the search engine is a passive adversary which tries to make identification decisions only by analysing the received and logged queries.

For conducting our study, we should simulate real user queries being channelled through anonymizing networks, like Tor, but since our model does not require any other information apart from the query content, it is needless to simulate the Tor functionality. Therefore, we work directly with real user search query logs. For obtaining real user queries, one option could be to seek volunteers who may allow us to record their search queries over a period of time, but it is not a viable solution considering the privacy concerns (the same concerns that motivate our work). Instead, we chose to utilize the publicly available real user search logs, such as the AOL search logs [1] released in 2006. These AOL logs are spread over a reasonably long duration (3 months).

For our study, we assume that the adversarial search engine has access to the list of possible Tor users performing web search over Tor($\mathbb{U}$). This is a realistic assumption because the search engine has access to a user's search history and it can determine whether a particular user has possibly started to use Tor by identifying changes in his query patterns (such as query frequency). If for a reasonable duration, e.g., a week, the search engine does not receive any queries from a user or an IP address, violating the user's typical querying pattern, it can mark that user as a potential Tor user. Even when the users delete their cookies, there are chances that the search engine might mistake these users to be possible Tor users. However, since the user querying patterns do not change, the search engine might be able to map these mistaken Tor users to new cookies if the querying patterns match, and continue profiling the users. Such mapping techniques or anti-aliasing techniques have been studied before in [13]. Though the content they deal with in [13] are large texts like bulletin boards and web pages, we believe similar techniques could be developed for mapping web search users associated with different cookies. Thus, we assume throughout our study that the search engine can generate a possible Tor user list and keep updating it.

Over a period of time, the search engine logs a set of queries (denoted $\mathbb{Q}$) that it receives from the Tor exit nodes (list of exit nodes is publicly available). These queries are issued by the users appearing in the list $\mathbb{U}$ and are all mixed with one another. Our goal (i.e., the adversarial search engine) is to identify the queries in $\mathbb{Q}$ that correspond to some or all users in $\mathbb{U}$. We model this identification problem as a classification problem in machine learning, whereby we train a classifier with the prior search history of the Tor

users (collected prior to the time they started using Tor) and ask the classifier to classify the queries in $\mathbb{Q}$ to their respective users in $\mathbb{U}$. Since we might want to associate the queries to all users, we will need one class per user. Therefore, the problem reduces to a multiclass classification problem.

The size of $\mathbb{U}$ (denoted N) is an important parameter for our study and for the level of privacy that can be provided to the users over Tor. We argue that in practice N may not be very large. As discussed in Section 1, recent research [4] shows that although there are, on an average 1893 Google users at one Tor exit node over one week, about 872 of these users access the services by signing into Google, making themselves identifiable even while using Tor. Similarly, a significantly large fraction of users may not disable their cookies, due to unawareness about tools like PWS [17]. In summary, even if there are 1500 Tor exit nodes in total and a large number of users might be using Tor for private web search, only a small fraction of these users actually remain anonymous to the search engine. In addition, the search engine might not want to track each and every one in this anonymous user set, but instead it might want to concentrate on few users - selected based on the kind of sensitive queries they send or based on their real world identities (like suspected terrorists). In light of these important observations, we consider a maximum of $N = 1000$ anonymous web search users, and try to associate the queries in $\mathbb{Q}$ to these users. We believe that this number 1000 is reasonable for experimental purposes.

Let us assume that the search engine is interested in identifying the queries corresponding to an user of interest, A. Let us say that $\mathbb{Q}$ contains n_u number of A's queries and n_o number of other users' queries (note that generally $n_u << n_o$). The search engine can select a query from $\mathbb{Q}$ at random and can trivially identify it to be A's query with a probability $p_{naive} = \frac{n_u}{(n_u+n_o)}$. Instead, if we apply our classification approach which has an accuracy of x (i.e., probability of correctly identifying A's queries) and a misclassification rate of y (i.e., probability of incorrectly identifying others' queries as A's queries) (These "accuracy" and "misclassification rates" are not in accordance with the standard Machine learning definitions), then we obtain a (very small) subset of $\mathbb{Q}$ which consists of $x * n_u$ A's queries, and $y * n_o$ other users' queries. Now, if we pick a random query from this subset, then the probability that this query is A's query is $p_{class} = \frac{x*n_u}{(x*n_u+y*n_o)}$. If our classification is doing a good job, i.e., if x is high and y is low, then p_{class} would be significantly higher than p_{naive}, which in turn would mean that we are doing a much better job of identifying user's queries than we do with a random guess. As a concrete example, our classification attacks for AOL user #67910 yield $p_{class} = 0.73$ ($x = 45/192$ and $y = 16/46062$), which is about 183 times more than the probability of a random guess $p_{naive} = 0.004$, when $N = 100$.

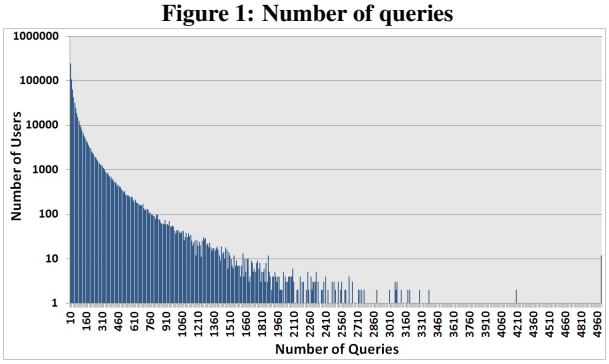
Through our classification experiments in the rest of this paper, we aim to find out the values of x and y for a diverse set of users firing different type of queries, and for different values of N (100, 200, 300, 500 and 1000).

3. EXPERIMENTAL PRELIMINARIES

3.1 AOL Data Statistics

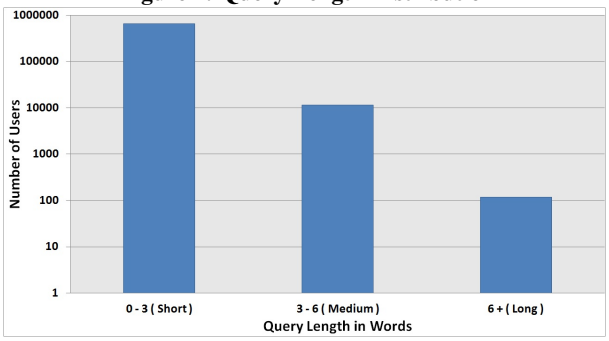
As mentioned earlier, the AOL logs are spread across three months duration. For our purpose, we consider the first two months data as the search history available to the search engine and the last month's data is the new queries information to be classified. These sets shall be referred to as the training set and test set henceforth. Instead of identifying the queries of all users in $\mathbb{U}$(since it can be time consuming), we want to concentrate on few specific users who we consider to be important - users of interest. To this end, we selected a set of 60 users of interest from the AOL logs and tried to find the accuracy in identifying their queries from the query pool of upto 1000 Tor users ($N = 1000$) using web search over Tor. The selection is made according to the AOL query statistics described below.

Figure 1: Number of queries



Different users send different number of queries over a time period. The query frequency of a user plays an important role in deciding whether a query coming out of a Tor exit node should be associated to the user. We calculated the total number of queries sent by all the AOL users over the 3 months period and plotted the number of users in different query bands. From the graph in Figure 1, we can see that most users (about 98.72%) issue less than 100 queries in a 3 month period. The rest are spread throughout the graph in smaller numbers, with the user density decreasing as the number of queries increase.

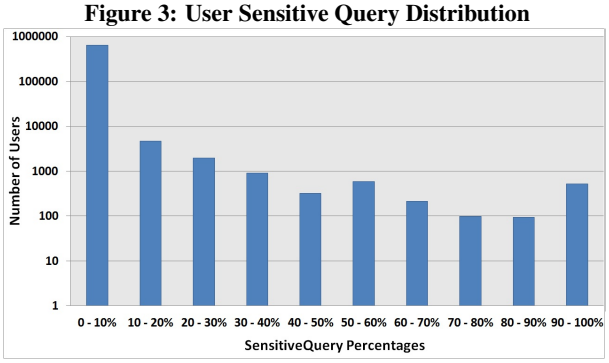
Figure 2: Query Length Distribution



Though query lengths are implicitly attached to the queries, they may contribute towards identifying the user. We plotted the number of users across three different query length bands – Short, Medium and Long. The users in short band have average query length less than 3 words, those in medium band have average query length lying between 4 to 6 words, and users in long band have average query length greater than 6 words. From Figure 2, we can observe that a large number of users send short queries.

The query content varies from user to user, and so it can provide additional information in identifying the user sending the query. We consider two broad categories of queries, namely – Sensitive and Insensitive. There is no hard and fast way to define these type of queries, and it simply depends on the application as to what is considered sensitive or insensitive. For example, with the current rise in terrorism, sensitive content might include the queries related to bombs, military details, etc. For our purpose, after observing queries of 2000 AOL users, we identified certain sensitive keywords relating to medical data, terrorism, weaponry, child abuse and pornography. If any one of these keywords occurred in a query, the query was labelled as sensitive. We labelled the queries of all

Figure 3: User Sensitive Query Distribution



the AOL users and found the percentage of sensitive-insensitive query distribution for each user. We plotted the user distribution across different percentages of sensitive queries, which is shown in Figure 3.

3.2 Selecting Users

As mentioned earlier, we only concentrate on identifying queries of a specific few (60) users of interest - chosen according to the categories discussed above. We had selected 20 users of interest from each category as follows:

Number of Queries: In order to comply with the statistics in Figure 1, we selected 14 users of interest at random from set of users who fire less than 100 queries, 4 users of interest at random from set of users who fire 101-500 queries and 2 users of interest at random from the set of users who fire more than 500 queries over a period of three months.

Query Length: Following the statistics in Figure 2, we have chosen 15 users of interest randomly from the set of users sending short queries, 3 users of interest were selected at random from the set sending medium length (3-6 words) queries and 2 users of interest were selected at random from the set sending long (more than 6 words) queries.

Sensitive Queries: Based on statistics in Figure 3, 10 users of interest are selected at random from the set of users sending 0–10% sensitive queries, 2 users of interest are selected at random from 10–20% sensitive query band, 2 users of interest are selected at random from 20–30% sensitive query band, 3 are selected from 50–60% sensitive query band and another 3 from 90–100% sensitive query band in proportion to the actual user distribution.

3.3 Selecting Classifiers

For our machine learning needs, we use WEKA[5], an open source software which includes a large number of classifiers and preprocessing options. We wanted to use this publicly available, off-the-shelf classification tool and estimate the accuracy levels that can be achieved. These accuracies can most likely be increased if we use classification algorithms specifically designed and tailored for this particular requirement.

WEKA has many in-built classifiers which can be used for our attacks. We have chosen Support Vector Machine (SVM) as the best classifier for our purpose based on the strong recommendations, such as [9], to use SVMs for textual classification or categorization, and its wide spread application in similar projects.There is more than one implementation of SVM algorithm in WEKA, and we have selected C-SVC binary classifier because of the simplicity in choosing the classifier parameters.

C-SVC is a binary classifier which identifies and associates data instances belonging to two classes. Multiclass classification can be solved by converting multi-class problem into multiple binary classification problems. These are popularly called as One-vs-All (OVA) and All-vs-All (AVA), as described in [16]. In OVA model, we build one separate classifier for each class in the dataset. For i^{th} classifier, the positive examples will be the training data with label i and the negative examples include all the data with a label different from i. In AVA model, we build $N(N-1)$ classifiers, one for each pair of classes i and j, where N is the number of classes. Each of these classifiers(i, j) gets trained on only the data belonging to classes i and j. At the end, the label is predicted by following a voting mechanism (see [16] for details). Both OVA and AVA are applicable to our problem and can yield good accuracies.

3.4 Selecting Attributes

Each entry in the AOL log is a tuple of the form: <AnonymousID, Query, Time, ItemRank, ClickURL>. In our model, since we work with the query content alone and do not concentrate on additional clickthrough patterns of the user, we neglect the ItemRank and ClickURL features.

The AnonymousID feature is used for distinguishing the AOL users and is treated as the Class Label for classification. Since the Query feature is a string and WEKA can not handle strings directly, we converted the strings to word vectors using the in-built WEKA preprocessing filter *StringToWordVector*. We added another feature Query Length, as described in Section 3.1, though it is implicit in the Query information. Since time feature cannot be used directly because of the inherent delay when queries are sent over Tor, we considered timing windows of considerable duration. Since it is hard to predict what size of the timing window might provide better results, we divided 24 hours in a day into different non-overlapping windows of sizes of 3, 4, 6 and 12 hours and compared the accuracies with each timing window size.

AnonymousID and Query are necessary attributes and in order to determine the impact of each additional attribute on the classification results, we tried to identify the average accuracy of all the users of interest when N=100, by including one additional attribute at a time. The average accuracies are indicated in Table 1. We can see that by including the Query Length feature reasonable performance is achieved both in the case of OVA and AVA. Addition of timing windows did not provide much improvement over the existing accuracies, both in the case of OVA and AVA. There could be other possible and better uses of these query times, but we neglect them for now. Hence for all the following experiments we included Query Length as an additional attribute along with Query and the AnonymousID.

4. EXPERIMENTS AND RESULTS

In our experiments, we tried to estimate the accuracy of the classifiers in correctly identifying queries of 60 users of interest. For each user of interest, we measure the accuracy across five datasets, where in each dataset, we vary the number of 'other users' whose queries are mixed with that of the current user of interest. The five datasets containing randomly selected 99, 199, 299, 499 and 999 other users were generated. In order to be consistent across all 60 users of interest, we used the same 'other user' datasets. Thus, when the user of interest's query set is mixed with the queries of these 'other users', we form datasets with N as 100, 200, 300, 500 and 1000 users.

For every user of interest A, we intuitively call the fraction of correctly identified user A's queries (denoted as x in Section 2) as *Correctly Classified* and the fraction of other users' queries incorrectly classified as user A's queries (denoted as y in Section 2) as *Misclassified* (These terms are not in accordance with the standard Machine Learning definitions). As discussed in Section 2, we want the *Correctly Classified* value to be as high as possible and *Mis-*

Classifier	Accuracies – No Additional Attributes	Accuracies – Including Query Length	Accuracies – Including Timing window			
			3 hrs	4 hrs	6 hrs	12 hrs
AVA	16.26%	14.58%	13.16%	14.08%	13.62%	14.41%
OVA	13.65%	14.41%	13.98%	12.99%	12.63%	14.15%

Table 1: Comparison of Accuracies for Attribute Selection

Figure 4: OVA Results Summary for Number of Queries

Number of Queries Fired on Average (M)	Total number of users	100 users		200 users		300 users		500 users		1000 users	
		Correctly Classified	Mis - classified	Correctly Classified	Mis- classified	Correctly Classified	Mis- classified	Correctly Classified	Mis- classified	Correctly Classified	Mis- classified
M < 100	14	4.35% (1/23)	0.02% (1/4497)	4.35% (1/23)	0.06% (5/8775)	4.35% (1/23)	0.06% (7/11800)	4.35% (1/23)	0.01% (3/20144)	0.00% (0/23)	0.02% (7/39481)
100 < M < 500	4	8.86% (7/79)	0.38% (20/5247)	8.86% (7/79)	0.11% (11/10239)	8.86% (7/79)	0.13% (18/13766)	7.59% (6/79)	0.06% (15/23501)	6.33% (5/79)	0.07% (33/46061)
500 < M	2	36.71% (76/207)	0.36% (19/5248)	19.32% (40/207)	0.22% (23/10238)	13.53% (28/207)	0.11% (15/13766)	14.01% (29/207)	0.06% (15/23501)	14.98% (31/207)	0.05% (22/46061)

classified value to be as low as possible, and consider it to be an optimal measure of the performance.

We obtained the results for all the 60 users of interest belonging to the three categories discussed in Section 3.2. For each category, we summarized the OVA results indicating the average values of *Correctly Classified* and *Misclassified* for all the users of interest in specific sub-categories. The summary of OVA results for Number of Queries is given in Figure 4, summary of OVA results for Query Length is given in Figure 5 and the summary of OVA results for Sensitive Queries is depicted in Figure 6. The results for AVA classification, for each category, came out to be very similar to that of OVA classification, and are thus not reported in the paper.

5. INTERPRETATION AND DISCUSSION OF RESULTS

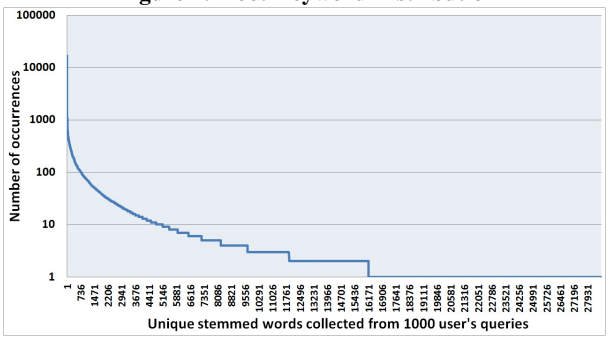
The first observation looking at Figure 4, 5 and 6 is that the average accuracies (i.e., the fraction of correctly classified queries) are reasonable, i.e., in most cases some fraction of users queries can always be correctly classified. The average accuracy across all the 60 users of interest is 25.95% when N=100, and this decreases to 18.95% when N=1000. These accuracies show that at least a quarter of the user's queries can be easily identified. The misclassification rates are also very low in almost all cases.

Looking at Figure 4 across the rows, we find that the accuracies are likely to increase with the number of queries posed by the user. We can explain this by looking at the distribution in (Figure 1); the number of users go down considerably with increase in the number of queries. If there are only few users who pose a large number of queries, their queries do not get mixed well and thus can be identified with a high probability. Figure 5 shows that longer queries are likely more identifiable. This is because only a very small fraction of users issue longer queries (more than 6 words), as seen from Figure 2.Following the same reasoning, we observe, from Figure 6, that accuracies are expected to increase as the sensitivity of the query content increases (recall the sensitive/insensitive query distribution in Figure 3). Though the results might seem intuitive and follow the trend that users who stand out are easily identifiable, they provide us a very good estimate of what percentage of these user queries can actually be identified because of the query properties.

Reasons behind the accuracies.

To understand these results, we try to find the reasons behind how and why a query gets identified as a user query. Since the data fed to the machine learning algorithm contained queries broken down into word vectors, we tried to identify the word usage distribution among the 1000 users. By trimming down all the query words, using stemming algorithms present in WEKA, we identified the 'root' keywords appearing in the queries and sorted them in the decreasing order of occurrence. The distribution can be seen in Figure 7. There were 28659 root keywords(i.e. stemmed words) in total and of these only 4797 root keywords had more than 10 occurrences. This distribution mimics the 'Long Tail' behaviour of web search queries, as discussed in [6], where each user is considered a bit eccentric and is expected to send both common queries (all root keywords with more than 10 occurrences in the distribution) and a few unique queries (at least one keyword with less than 10 occurrences in the distribution). These unique keywords are what we think might be contributing for query identification.

Figure 7: Root Keyword Distribution



Reasons behind accuracy decrease.

We understood how the accuracies are obtained, but the question that is remaining is - *"Why do these accuracies decrease when we increase the value of N, the number of users using web search over anonymizing network?"*. We determine the label of a test query based on the number of occurrences of similar query words in the training sets. More the occurrences of such words in the user of interest's train set, more are the chances for it to be identified as a user of interest's query and vice versa. With the increase in the value of N, the size of the *other users*'s training set greatly increases, improving the chances of occurrence of test query words within. This decreases the possibility of classifying the query as a user of interest's query. Here is one example that we came across for one AOL user."j c penney catalog" was a test query and there was no exact looking query, or a query formed by subset of its keywords in the other users' training set. The words "j c penney" and "catalog" had occurred before in the particular user's train set, and hence it was labelled as the particular user's query when N=100. However as

Figure 5: OVA Results Summary for Query Length

Query Length	Total number of users	100 users		200 users		300 users		500 users		1000 users	
		Correctly Classified	Mis-classified	Correctly Classified	Mis-classified	Correctly Classified	Mis-classified	Correctly Classified	Mis-classified	Correctly Classified	Mis-classified
Short	15	19.40% (13/67)	0.025 (1/4497)	17.91% (12/67)	0.20% (19/9555)	16.42% (11/67)	0.15% (19/12848)	16.42% (11/67)	0.10% (21/21934)	15.15% (10/66)	0.06% (26/42991)
Medium	3	20.0% (4/20)	0.15% (8/5248)	20.0% (4/20)	0.01% (1/10238)	20.00% (4/20)	0.01% (1/13766)	20.00% (4/20)	0.00% (1/23501)	20.00% (4/20)	0.00% (1/46061)
Long	2	93.33% (28/30)	0.02% (1/5247)	90.00% (27/30)	0.00% (0/10238)	90.00% (27/30)	0.01% (1/13767)	90.00% (27/30)	0.035% (6/23501)	90.00% (27/30)	0.03% (14/46061)

Figure 6: OVA Results Summary for Sensitive Queries

Sensitive Percentage Bands	Total number of users	100 users		200 users		300 users		500 users		1000 users	
		Correctly Classified	Mis-classified	Correctly Classified	Mis-classified	Correctly Classified	Mis-classified	Correctly Classified	Mis-classified	Correctly Classified	Mis-classified
0%-10%	10	39.2% (245/625)	0.83% (39/4723)	38.24% (239/625)	0.53% (49/9215)	38.24% (239/625)	0.38% (47/12390)	37.76% (236/625)	0.22% (46/21151)	35.68% (223/625)	0.16% (65/41455)
10%-20%	2	14.78% (34/230)	0.06% (3/5248)	13.04% (30/230)	0.05% (5/10239)	13.04% (30/230)	0.03% (4/13767)	12.61% (29/230)	0.02% (5/23502)	13.04% (30/230)	0.06% (29/46062)
20%-30%	2	36.36% (24/66)	0.13% (7/5248)	37.88% (25/66)	0.12% (12/10239)	37.88% (25/66)	0.12% (16/13767)	37.88% (25/66)	0.14% (34/23502)	37.88% (25/66)	0.09% (40/46062)
50%-60%	3	22.22% (4/18)	0.08% (4/5248)	22.22% (4/18)	0.03% (3/10239)	22.22% (4/18)	0.03% (4/13767)	22.22% (4/18)	0.02% (4/23502)	22.22% (4/18)	0% (1/46062)
90%-100%	3	96.3% (78/81)	0.1% (5/5248)	96.3% (78/81)	0.06% (6/10239)	96.3% (78/81)	0.06% (8/13767)	74.07% (60/81)	0.03% (7/23502)	29.63% (24/81)	0.03% (13/46062)

N increased to 300, this query was not labelled as the user of interest's query, since there were more occurrences of "j c penney" and "catalog" terms in the *other users*'s query training set. In this way, depending on the occurrence of query terms in the training sets, the accuracies decrease as the value of N increases.

Influence of a time gap.

The web content that the user is interested in varies with time. A time gap between the test and the training data sets might make it harder to de-anonymize the data, because the common content between the test and the train sets decreases with time. However, prior research [20] shows that users tend to pose exact same queries over and over, as it is easier compared to remembering the url of the search result, or more efficient than performing an internal search within the website. This behaviour is described as "Bookmarking" [20]. For a considerable long period, these queries do not change and this helps the machine learning approach in identifying at least a small fraction of the user queries. We tried identifying the percentage of same queries repeated in the test and train sets (assumed to be bookmark queries), for two AOL users with IDs 20894930 and 67910. The percentage of bookmark queries were less than 6%, but the machine learning accuracies were higher than 58%, infact higher than 90% for user ID 20894930. Thus at least a small fraction of user queries could still be identified, even when there is time gap between the test and training data sets.

6. RELATED WORK

Query classification problem studied in this paper is very similar to authorship attribution. Authorship attribution has a long history beginning in the 19th century. "Federalist Papers" is an early example of the problem [12]. In early authorship studies, the primary goal is to model unique author styles by looking at text characteristics, such as vocabulary richness (zipf's word frequency distribution), word length, choice of rhymes and habit of hyphenation.

During the evolution of authorship attribution problem, type of data being studied changed from published articles to electronic text – emails, tweets, blogs and online messages [25]. In the traditional authorship attribution problem, studied texts were long articles with few authorship possibilities as in the case of "Federalist Papers". However, electronic data may neither be long nor does it have a few possible authors. Also in the case of emails, style of the text changes according to receiver of the e-mail; same author can write in different styles for different recipients. These issues make authorship attribution problem more challenging in the context of electronic data.

Identification of search queries is even harder compared to e-mails or articles since the query length is much shorter. However, recent studies show that "Vanity searches" [19], sending search queries containing personally identifying information such as name, address and telephone number; significantly contribute towards query sender identification. These vanity searches leak the user privacy, even when privacy preserving tools like TrackMeNot or Tor are used.

To the best of authors' knowledge, this paper is the first to study the problem of identifying web search queries given a pool of queries from users of an anonymizing network. This is related but different from the problem of identifying queries from a search log. First, an adversary in our application is the search engine itself and not a third party attempting to de-anonymize a search log. Second, unlike a third party, the search engine is already in possession of users' search history using which it can effectively train a classifier. Moreover, the goals of our study were also different; we were interested in evaluating existing classifiers to address this problem so as to keep our attacks simple.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we studied the problem of identifying a user's queries from a pool of queries received by a search engine over an anonymizing network. We demonstrated that an adversarial search engine, equipped with only a short-term search history, can extract user of interest's queries by utilizing only the query content and off-the-shelf machine learning classifiers. More specifically, by treating a selected set of 60 users – from the publicly-available AOL

search logs – as users of interest performing web search over an anonymizing network, we showed that their queries can be identified with 25.95% average accuracy when N=100, and with 18.95% average accuracy when N=1000. Though the average accuracies are not so high, our results show that few users of interest can be identified with accuracies as high as 80–98%, even when the value of N=1000. We tried to identify the reasons behind how and why a query gets classified as a user query, and answer why the accuracies tend to decrease as the number of users using the anonymization service increase. Our results, therefore, cast serious doubt on the effectiveness of anonymizing web search queries by means of anonymizing networks.

One of the strengths of our attacks is that they only use minimal information (query content) for identification of users' queries and use off-the-shelf classification techniques. Under realistic conditions, it would certainly be possible to improve our attacks by taking into account other information that would be available to the search engine under normal circumstances. For instance, exact query timestamps may very well be a useful attribute. Similarly, exit node IP address is also likely to improve the accuracies. Finally, a search engine can build better classifiers by training them on long-term (longer than 2 months) search histories of the users. By utilizing the geographical locality information accompanying the queries and the users [7] (place names and details pertaining to certain localities) and using contextual information for query classification [3], we plan to further improve the results. We defer these items as an interesting avenue for future research.

Acknowledgments

We are grateful to ASIACCS'10 anonymous reviewers for their insightful feedback. We also thank Lisa Hellerstein for discussion on machine learning classifiers and her helpful comments on our work, and Yasemin Avcular for her suggestions and help with the experiments.

8. REFERENCES

[1] AOL Search Log Mirrors, `http://www.gregsadetsky.com/aol-data/`.

[2] M. Barbaro and T. J. Zeller. A Face Is Exposed for AOL Searcher No. 4417749. In *The New York Times*, August 09 2006.

[3] H. Cao, D. H. Hu, D. Shen, D. Jiang, J.-T. Sun, E. Chen, and Q. Yang. Context-aware query classification. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 3–10, New York, NY, USA, 2009. ACM.

[4] C. Castelluccia, E. De Cristofaro, and D. Perito. Private information disclosure from web searches (the case of google web history). In *Privacy Enhancing Technologies Symposium (PETS), to appear*, 2010. Available at: `http://planete.inrialpes.fr/%7Eccastel/PAPERS/historio.pdf`.

[5] I. W. . E. Frank. *Data Mining–Practical Machine Learning Tools and Techniques, Second Edition*. Elsevier, 2005.

[6] S. Goel, A. Broder, E. Gabrilovich, and B. Pang. Anatomy of the long tail: ordinary people with extraordinary tastes. In *WSDM '10: Proceedings of the third ACM international conference on Web search and data mining*, pages 201–210, New York, NY, USA, 2010. ACM.

[7] L. Gravano, V. Hatzivassiloglou, and R. Lichtenstein. Categorizing web queries according to geographical locality. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 325–333, New York, NY, USA, 2003. ACM.

[8] S. Hansell. Marketers Trace Paths Users Leave on Internet. In *The New York Times*, September 15 2006.

[9] M. Hearst, B. Schvlkopf, S. Dumais, E. Osuna, and J. Platt. Trends and controversies - support vector machines. *IEEE Intelligent Systems*, 13(4):18-28, 1998.

[10] D. Howe and H. Nissenbaum. TrackMeNot: Resisting Surveillance in Web Search. In *On the Identity Trail: Privacy, Anonymity and Identity in a Networked Society, Ian Kerr, Carole Lucock and Valerie Steeves (editors)*, 2008.

[11] E. Kushilevitz and R. Ostrovsky. Replication is not needed: single database, computationally-private information retrieval. In *FOCS '97: Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS '97)*, 1997.

[12] F. Mosteller and D. L. Wallace. Inference and Disputed Authorship: The Federalist. Addison-Wesley, Reading, Massachusetts, 1964.

[13] J. Novak, P. Raghavan, and A. Tomkins. Anti-aliasing on the web. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 30–39, New York, NY, USA, 2004. ACM.

[14] NYTimes: Google Resists U.S. Subpoena of Search Data, `http://www.nytimes.com/2006/01/20/technology/20google.html?_r=1`.

[15] S. T. Peddinti and N. Saxena. On the Privacy of Web Search Based On Query Obfuscation: A Case Study of TrackMeNot. In *Privacy Enhancing Technologies Symposium (PETS), to appear*, 2010.

[16] R. Rifkin. *Multiclass Classification*, 06 March 2006. 9.520 Class 08, Available at: http://ocw.mit.edu/NR/rdonlyres/2CA61B6A-9C0D-4E8A-AEE3-F4B1BFD96AF0/0/lec8.pdf.

[17] F. Saint-Jean, A. Johnson, D. Boneh, and J. Feigenbaum. Private web search. In *WPES '07: Proceedings of the 2007 ACM workshop on Privacy in Electronic Society*, 2007.

[18] Scroogle.org, `http://scroogle.org/`.

[19] C. Soghoian. The Problem of Anonymous Vanity Searches. *SSRN eLibrary*, 2007.

[20] J. Teevan and E. Adar. Information re-retrieval: repeat queries in yahoo's logs. In *In SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 151–158. ACM, 2007.

[21] Tor Anonymizing Network, `http://www.torproject.org/`.

[22] TrackMeNot: Browser Plugin, `http://www.mrl.nyu.edu/~dhowe/trackmenot/`.

[23] B. Trancer. Click: What millions of people are doing online and why it matters. Hyperion, 2008.

[24] S. Ye, S. F. Wu, R. Pandey, and H. Chen. Noise injection for search privacy protection. In *International Conference on Computational Science and Engineering (CSE)*, pages 1–8, 2009.

[25] R. Zheng, J. Li, H. Chen, and Z. Huang. A framework for authorship identification of online messages: Writing-style features and classification techniques. *J. Am. Soc. Inf. Sci. Technol.*, 57(3), 2006.