A Large-scale Study of Automated Web Search Traffic

Greg Buehrer
Microsoft Live Labs
One Microsoft Way
Redmond, WA 98052
+1 425 704 0049
buehrer@microsoft.com

Jack W. Stokes
Microsoft Research
One Microsoft Way
Redmond, WA 98052
+1 425 703 1993
jstokes@microsoft.com

Kumar Chellapilla
Microsoft Live Labs
One Microsoft Way
Redmond, WA 98052
+1 425 707 7575
kumarc@microsoft.com

ABSTRACT

As web search providers seek to improve both relevance and response times, they are challenged by the ever-increasing tax of automated search query traffic. Third party systems interact with search engines for a variety of reasons, such as monitoring a website's rank, augmenting online games, or possibly to maliciously alter click-through rates. In this paper, we investigate automated traffic in the query stream of a large search engine provider. We define automated traffic as any search query not generated by a human in real time. We first provide examples of different categories of query logs generated by bots. We then develop many different features that distinguish between queries generated by people searching for information, and those generated by automated processes. We categorize these features into two classes, either an interpretation of the physical model of human interactions, or as behavioral patterns of automated interactions. We believe these features formulate a basis for a production-level query stream classifier.

Categories and Subject Descriptors

D.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.3.m [**Information Storage and Retrieval**]: Miscellaneous.

General Terms

Algorithms, Measurement, Performance, Experimentation.

Keywords

Query, Search, Bot

1. INTRODUCTION

The Web has quickly become the de facto method for general information gathering. This transition has allowed web search to grow into a multi-billion dollar industry in only a few years. In addition, the proliferation of the web has allowed it to become an environment for cultivating advancements along many dimensions of research. One such dimension is adversarial informal retrieval. Popular challenges in this arena are email spam and web link spam. Email spam is designed to return the receiver to a location in which he would then be coaxed into purchasing a product, relinquishing his bank passwords, etc. This type of email is almost always automated. One study suggested that 85% of all email spam, which constitutes well more than half of all email, is generated by only 6 botnets [1].

With web spam, the generator is attempting to manipulate the search engine towards the end goal of improving its rank. For example, a high number of automatically generated web pages can be employed to redirect static rank to a small set of paid sites [5].

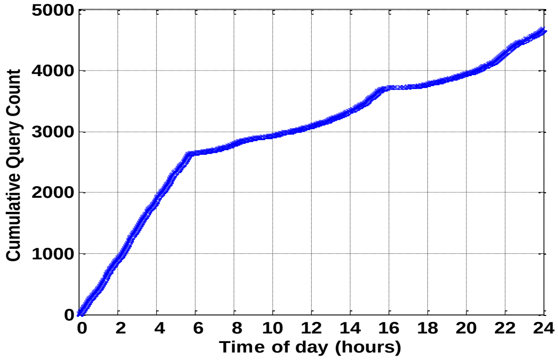

Figure 1. Time of day vs. aggregate queries for one user Id.

In this paper, we focus our attention on an understudied form of automation, namely web search engine queries. We define legitimate search queries as those queries entering the system which are typed by a human to gather information. Then, all other traffic is deemed automated traffic. Automated search traffic is of significant concern because it hampers the ability of large scale systems to run efficiently, and it lowers user satisfaction by hindering relevance feedback. Because search engines are open for public consumption, there are many automated systems which make use of the service. A *bot* – the entity generating the automated traffic – may submit queries for a variety of reasons, most of which are benign but not overly monetizable.

Airweb'08, April 22, 2008, Beijing, China.

[1] http://computerworld.co.nz/news.nsf/scrt/C70ED4E3A608806CCC25740100186FC6

As an example, *rank bots* periodically scrape web pages to determine the current ranking for <*query,URL*> pairs. A Search Engine Optimization company (SEO) may employ a rank bot to evaluate his web page ranking optimizations for his clients. If a client's current rank is low, a user may need to generate many *NEXT PAGE* requests to find it in the search engine's results. Since SEOs can have many clients, this practice can result in significant amount of traffic for a single user Id. As an example, consider the user represented in Figure 1. This bot queries every 7 seconds (approx) from midnight to about 6A.M., then at a slightly slower rate (approx every 30 seconds) for the rest of the day. The total number of queries is about 4,500, far more than a human would do in normal browsing (many different pairs were queried).

There are several motivations for detecting automated traffic. Most importantly, correctly separating legitimate queries from automated queries can improve the end user experience in a number of ways. First, search latency can be reduced for legitimate queries; the search engine company may wish to throttle users to improve the Quality of Service (QoS) for interactive users. By reducing the total traffic serviced, or by reordering requests, response times for human users could be lowered (or maintained with less hardware). In addition, some search engines may consider click-through data implicit feedback on the relevance of a URL for a given query [11][13]. This feedback may then be used to modify the rankings of the associated URLs. This may extend beyond explicit clicks, and include the absence of a click, such as to demote all sent URLs which were not clicked. Conversely, if the fourth result is clicked 3 times as often as the first result for a given query, it may imply that the fourth result is ranked too low. However, this form of ranking is as susceptible to malicious behavior as link ranking algorithms – an SEO could easily generate a bot to click on his clients' URLs. This form of automatically clicking links is commonly referred to as click fraud.

Click fraud for paid search results has been a challenge for some time [3][10][11]. This activity may involve rival companies automating click activity on paid search results and banner ads in an attempt to increase an opponent's marketing costs. Another source of click fraud occurs when illegitimate businesses attempt to pose as intermediate search engines who host ads and forward illegitimate click traffic. Recently, a study by *Click Forensics* reported that click fraud for paid results in the 4th quarter of 2007 represents over 16% of all ad click traffic, up from 14% in the same quarter last year[2]. In particular, the report notes that search engine ads experience a fraud rate of 28.3%. Other reports suggest a lower fraud rate, closer to 6% [3], which is still rather high. In this paper, we target automated traffic and clicks for *unpaid* results, which do not have the potential benefit of using conversion rates (*e.g.* Cost-Per-Action metrics) as secondary indicators [10] for legitimate activity.

Detecting automated traffic can be difficult for several reasons. To begin with, the identification of user sessions is not trivial. One method to achieve this is through the use of cookies. A cookie is placed on the user's machine whenever the browser visits the site. Some users do not allow cookies, so each visit to the site appears to be a new user. In this case, the IP address of the request can be used, if there is sufficient confidence that the IP address is not a shared resource. In this work, we assume the ability to identify user sessions.

A second challenge in detecting automated traffic is that it may not be clear, even to a panel of experts viewing the queries, whether the source is automated. Although current techniques used to automatically query search engines can be relatively simple, sophisticated botnets certainly improve the ability of the programmer to mimic human traffic patterns [4].

Finally, as with most adversarial challenges, the behavior of the adversary changes over time. This suggests that specific signature-based solutions are only effective in the near term. For example, a bot may use the same IP address for an extended period of time, permitting a short term solution of ignoring traffic from that address. These types of practical solutions lead to black lists for IP addresses, user agents, referrers, etc., which must be constantly updated by the search engine.

This paper makes the following contributions.

- A large-scale study of search engine traffic (100M requests) is performed.
- Several real-world bot patterns are described.
- Based on the study, a set of discriminating features is presented, designed to separate automated traffic from human traffic.

The rest of the paper is organized as follows. Related work is provided in Section 2. In Section 3, we describe the search query data used in this study. Section 4 describes behavioral features of current-day bots. In Section 5, we provide details of our proposed features. We partition the features into two groups, namely physical model features and behavioral features. We provide preliminary results using the features in classification, and then conclude in the final section.

2. RELATED WORK

Relatively little work has specifically targeted classification of automated traffic in query logs. Agichtein, Brill, Dumais and Ragno developed models depicting user behavior for web search [1]. In this work, the authors are primarily interested in modeling users to guide relevance rankings, but some of these features could be used to partition humans from automated traffic as well. They point out that users provide more than click-through data when interacting with search engines. The authors consider deviations from normal behaviors, such as large increases in click-through rates for <*query,URL*> pairs. In addition, they incorporate page dwell time, query reformulation, and query length, among other features.

Research which studies click fraud in sponsored search results has examined traffic patterns and user behavior. These works do not address bot traffic with respect to organic results, but they do offer insight into the nature of the query stream. Daswani, *et al.* [4] dissect a large click botnet called *ClickBot.A.* and describe its functionality and technique in detail, with accompanying source code. The botnet is of particular interest because it exhibits controlled execution so as to avoid detection, while still generating significant fraudulent impact. It replicates client bots on over 100,000 machines, each of which have a separate IP address and only click on at most twenty items. The authors do not provide a detection method.

[2] http://www.clickforensics.com/Pages/Releases.aspx?r=01312008

A report by Tuzhilin [11] describes the challenges and issues with click fraud detection. In the report, the author concludes that Google, Inc is taking sufficient steps towards mitigating click fraud. Techniques include both static analysis and dynamic analysis, although exact measures are not described. The report also discusses an alternate reward system, in which rather than employing a system based on click-through rates, it is more advantageous for both parties if conversion rates were employed instead. Schluessler, Goglin and Johnson [10] develop a client-side framework for detecting whether input data has been automatically generated. The technique targets online gaming, but also mentions that it can be used to address some forms of click fraud in online advertising.

Fetterly, Manasse and Najork [5] perform a similar study to our work to discover web link spam. They illustrate that statistical analysis of web page properties, in particular features such as out degree distributions, host-machine ratios, and near duplicate document clusters can provide significant lift in labeling portions of the web as spam or legitimate material.

Anick [2] removes both known and suspected bots coming from internal AltaVista addresses for a study on web searcher behavior using terminological feedback. To eliminate bots traffic from a study on mobile web search, Kamvar and Baluja only considered traffic from a single large wireless carrier [8]. Karasaridis, Rexroad and Hoeflin [9] analyze the transport layer to discover IRC-based botnets attempting Denial-of-service attacks, among other malicious behavior. The method does not user signatures, instead monitoring ports for controller traffic patterns. The work does not investigate botnets attacking search engines.

3. DATA SET DESCRIPTION

In this section, we describe the data used in our study. We obtained a random sample of approximately 100M requests to a popular search engine from a single day (August 7, 2007). We sampled user queries, such that if a user is in our sample, then all his queries from that day are also included the sample. For this study, we further prune the data to include only users who query at least five times in the day, resulting in 46M requests.

In this study, users are sessionized with a cookie and assigned a user Id. It is also common to sessionize by the requesting IP address. Although in some cases a single IP will service multiple users (*i.e.* proxies), and in some cases a single user may request from several IPs (see Figure 4), the technique of sessionizing by IP can be a useful feature [12]. A brief study of a single day's data showed that of 19,332,100 distinct IPs which only had sessions from users without cookies enabled, 19,192,560 of these IPs had between 2 and 100 queries. Upon inspection, we felt the majority of these low query count IPs were distinct users. We believe a classifier trained on cookied data could also help to validate these potential sessions as well. Finally, it is possible for a single machine to produce both bot and human search engine traffic. In these cases, we do not attempt to separate multiple signals from a single cookie.

Finally, we offer some nomenclature. A *query* is an ordered set of keywords sent to the search engine. The engine then provides as a response an *impression set* (or simply *impressions*), which is a set of displayed results (both sponsored and organic). A query may have multiple *requests*, such as for result page two, upon which the engine will respond with additional impressions, *e.g.* the results for page two. Thus, the total number of requests may be more than the total number of queries. A *click* is always with respect to the impression presented to the user.

4. QUALITATIVE ANALYSIS

We now describe several bots discovered through the course of studying the query stream. While these are not inclusive, they are meant to present the reader with common forms of automated traffic.

The first bot that we present scans the index for top spam words. Typically, the goal of improving a website is to offer goods or services for financial gain; thus, a metric relating the query term to the other terms often found in email and/or web spam may be an indication that the query is generated by a bot. This class of bot rarely clicks, often has many queries, and most words have high correlation with typical spam. An example of 12 queries from one particular spam bot are presented in Table 1.

Query	Query
Managing your internal communities	based group captive convert video from
mailing list archives	book your mountain resort
studnet loan bill	agreement forms online
your dream major	find your true love
computer degrees from home	products from thousands
free shipping coupon offers	mtge market share slips

Table 1. An example of a simple spam bot.

A second bot, which has a similar pattern of a large number of queries without clicking, but a different bag of words is a finance bot. Eighteen sample queries are presented in Table 2. Most of the keywords in the query are associated with mortgages, credit, money and the housing industry in general. The goal of this bot is to ascertain which web sites are most correlated with these finance terms in the search index.

Query	Query	Query
1sttimehomebuyer	badcreditmortgage	equity
1sttimehomebuyer	badcreditrefinance	equityloans
2ndmortgage	banks	financing
2ndmortgage	bestmortgagerate	financinghouse
badcredithomeloan	debtconsolidation	financinghouse
badcreditloan	debtconsolidationloan	firstmortgage

Table 2. An example of a finance bot.

Some bot activity implies less benign intent. The bot whose queries appear in Table 3 seems to be querying the system for various URLs which are either web sites owned by spammers and operated as spam sites (e.g. **http://adulthealth.longlovetabs.biz/cialis.htm**) or web sites on legitimate, hijacked domains created to host spam (e.g. **http://astro.stanford.edu/form_1/buy_cialis_oneline.html**). Presumably, the bot is attempting to boost its search engine rank.

Some bots not only repeatedly query the system for information with respect to a particular category, but query in such a way that provides an unnatural signature. For example, the stock bot queries in Table 4 almost all are single keywords, and those keywords are primarily of length three or four. This bot appeared to be searching for financial news related to particular companies.

Query
`http://astro.stanford.edu/forum/1/buy.cialis.online.html`
`http://adulthealth.longlovetabs.biz/cialis.htm`
`http://www.bigdrugstoreforyou.info?Viagra.cialis`
`http://www.cheap.diet.pills.online.info/drugs/pagemaker.html`
`http://dosap.info/d.php?search=ed,viagra,levitra,cialis`
`http://www.generic.viagra.cialis.levitra.info/index/cialis.php`
`http://contrib.cgi.club.cc.cmu.edu/jjimenez foro language lang english email user activate dir 14 tramadol er.html`
`http://www.pharmacydirectory.biz/submitlink5.html`
`http://www.get.prescriptions.online.biz/buy.viagra.online.htm`
`http://www.redloungebiz.section.gb?page=5`
`http://www.emprenderengalicia.biz/index.php?option=com joomlaboard itemid 49 func post do reply replyto 4673 catid 8`

Table 3. A URL bot.

pae	cln	eu3	eem	olv	oj	lqde	igf	ief
nzd	rib	xil	nex	intc	tei	wfr	ssg	sqi
nq	trf	cl	dax	ewl	bbdb	csco	pl	idti
nesn	edf	intl	spx	ewj	tasr	ibkr	lat	hb1
mesa	edl	dram	iev	sndk	rukn	ifg	igv	ms

Table 4. Stock bot queries.

Another common bot scenario is when a user Id sends queries from many different cities within a short amount of time. An example is shown in Table 5 (IP addresses have been coded to preserve anonymity). This user Id sent 428 requests over a 4 hour period, from 38 different cities. Also, the bot always uses the *NEXT_PAGE* buttons when available, and this bot never clicks. The bot's queries had an unusually high number of adult terms. We suspect the user Id is automating traffic through anonymous browsing tools, but oddly those tools did not account for machine cookies.

Time	IP	city
4:18:34 AM	IP1	Charlottesville, Virginia
4:18:47 AM	IP2	Tampa, Florida
4:18:52 AM	IP3	Los Angeles, California
4:19:13 AM	IP4	Johnson City, Tennessee
4:22:15 AM	IP5	Delhi, Delhi
4:22:58 AM	IP6	Pittsburgh, Pennsylvania
4:23:03 AM	IP7	Canton, Georgia
4:23:17 AM	IP8	Saint peter, Minnesota

Table 5. Bot from the same cookie but many cities.

It is not uncommon for a source of automated traffic and a legitimate user to originate from the same machine. In some cases, it may be botnet-related activity. However, a second common scenario is that the originator of the bot program is also using the search engine, possibly to set up the program. For example, a particular user Id issued 6,534 queries, with only four clicks. Upon inspection, the four clicks were from the first five queries in the day, namely "pottery barn", "pottery barn kids", "pottery barn kids outlet", and "pottery barn kids outlet store", and "pier 1". These queries spanned about 7 minutes, which is a typical usage pattern. The user Id then issued 6,529 queries over the course of three hours without a click – clearly bot activity.

In a final example, one user Id queried for the same terms 1,892 times over the course of the day. Of those requests, 1,874 had click responses. A possible motive for a very high click rate is to glean why the top results are so ranked. Then, the user can improve the rank of his page by incorporating the discovered attributes. For example, if a user queries the index for "best flowers in San Francisco" and then scrapes the html of the top 1,000 impressions, he can find the most common keywords in those pages, their titles, etc. and incorporate them into his own site.

Name	Description	Type
Number of requests, queries, clicks	Number of requests, queries, clicks	Physical
Query Rate	The max number of queries in any 10 second period	Physical
Number of IPs/locations	Number of originating IPs/cities.	Physical
Click-Through Rate	Ratio of queries to clicks	Behavioral
Alphabetical Score	Indicator that the queries are in alphabetical order	Behavioral
Spam Score	Indicator that the keywords are associated with spam	Behavioral
Adult Content Score	Indicator that the keywords are pornographic in nature	Behavioral
Keyword Entropy	Informational entropy of query terms	Behavioral
Keyword Length Entropy	Informational entropy of query term lengths	Behavioral
Request Time Periodicity	Periodicity of requests, queries, clicks	Behavioral
Advanced Query Syntax Score	Number of advanced syntax terms in requests, eg inURL:, intitle:, site:	Behavioral
Category Entropy	Informational entropy of categories associated with distinct queries	Behavioral
Reputation	Blacklisted IPs, user agents, country codes, etc.	Behavioral

Table 6. Summary of feature set.

5. QUANTITATIVE ANALYSIS

Table 6 provides an overview of our set of potential features for detecting automated traffic in the query stream. We generally classify these features into two groups. The first group is the result of considering a physical model of a human. The second group is a set of observed behaviors of current-day automated traffic. In the following two subsections, we investigate each potential feature in some detail. Histograms are built for the features, which are then normalized to 100,000 users. Areas of high bot class lift in the graphs are then circled. Thus, the vertical axes are counts of users for the feature, and the horizontal axes are discretized ranges of that feature. In a few cases, we normalized to one million user Ids to allow the areas of interest to be sufficiently displayed.

5.1 Physical Model Feature Set

In this section, we discuss several features which are designed to model the interaction of a user and the search engine. Humans have physical limitations for entering queries, reading the results, and clicking on URLs. For example, a typical person can only issue and absorb a few queries in any ten second period. A user with 100 distinct request in ten seconds would lie outside the boundary of normal use. Search query traffic entered by automated means are not subject to these physical limitations. Thus, the following features may be used to discriminate between web search traffic from humans and automated bots.

5.1.1 Number of Requests, Queries, Clicks

A strong first indicator of automated traffic is volume. Bots often submit many more queries (and possibly clicks) in a given day than the typical person. Volume represents a class of features for which simple aggregate statistics can provide insight into the class of a user Id.

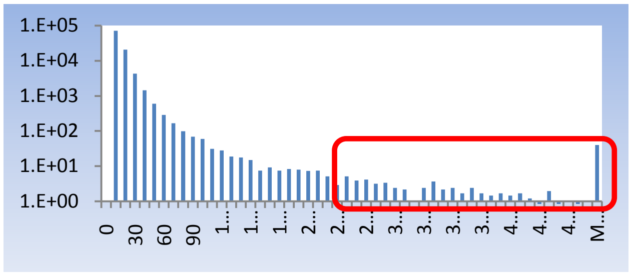

Figure 2. Number of requests.

For example, in Figure 2 we plot the distribution of the number of search requests from each unique user in our sample. While it is possible that a human user submits more than 200 queries in a given day, the histogram suggests it occurs with an unnatural probability. Upon inspection, we found that most of the traffic at this volume appeared automated. As an example, one user Id queried the search engine for “mynet” 12,061 times during this day.

5.1.2 Query Rate

Since bots are automated, they often enter queries at a much higher rate than queries which have been entered on a keyboard by a human. Various statistics of the query rate such as the average, median, and maximum can help distinguish queries generated by bots versus humans. We studied the query rates for human traffic and concluded that humans rarely submit more than 7 requests in any 10 second interval. In Figure 2, we plot the distribution of the maximum queries for a user in any 10 second interval over the course of the day. The users falling into the circled area were by and large bot traffic.

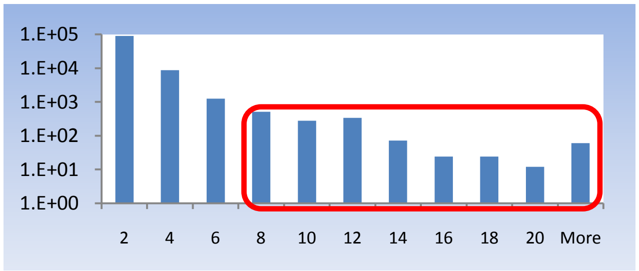

Figure 3. Max requests in any 10 seconds interval.

5.1.3 Number of IP Addresses / Locations

A human cannot be in two distant places at the same time. We maintain a list of requester IP addresses used by each user Id. The motivation is to discover potential bot nets. If a user’s cookie is compromised by miscreants and is used to make queries from two or more IP addresses, possibly located across large geographical distances, or is used in an interleaved fashion from two IP locations again separated by significant distances, then the unique Id likely belongs to two or more computers each of which are owned by a botnet[3]. A second usage scenario is when a user Id is querying the system through an anonymous browsing tool, but has not disabled cookies.

When correlating IP addresses, care must be taken to allow for mobile computers and devices which are used in the morning in one city, but later in the day at one or more additional cities. Also, users accessing the internet via a dial-up modem are often assigned a new IP address by the internet service provider (ISP) each time the user logs into the internet service. As a result the feature must ignore small variances in geographic location. In Figure 4, we show a histogram of the number of users employing Multiple IP addresses (normalized to one million users). Figure 5 depicts the same users wherein only the first two octets of an IP address are considered. This allows for multiple IP addresses in the same geographical region. We have highlighted the region where we find significant lift in bot classification. The bot in Table 5 would be flagged by this feature.

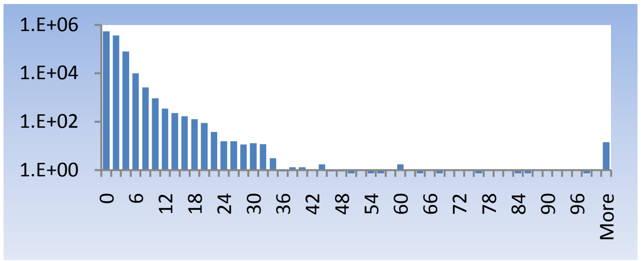

Figure 4. Distinct IP address (all four octets).

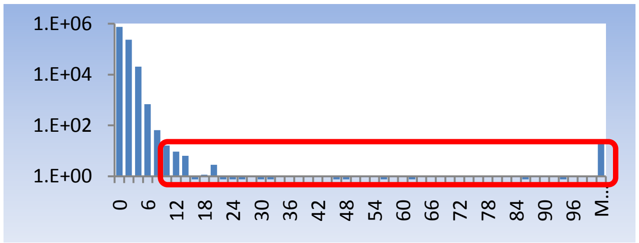

Figure 5. Distinct IP address (first two octets).

5.2 Behavioral Feature Set

The previous subsection introduces physical features that attempt to discriminate traffic generated by humans from that produced by automated means. However, automated search query traffic can be modeled to mimic human input. For these reasons we now provide additional features which seek to classify legitimate web

[3] http://www.hitslink.com/whitepapers/clickfraud.pdf

search traffic generated by typical users from illegitimate traffic generated by automated means. In many cases, we will illustrate the efficacy of the feature with an example of a discovered bot.

5.2.1 Click-through Rate

Much of automated traffic is likely used for information gathering purposes, either to examine the search engine's index, or to collect data for self-use, and thus exhibits lower than typical click-through rates. Previously published click-through rates for humans vary, but most show that most users click at least once in ten queries. Our own inspection of the data suggests that many of the zero-click users are automated. Further, when used in conjunction with the total number of queries issued over the day, the feature provides very good lift.

We illustrate this principle with two distributions, Figure 6a and Figure 6b. In Figure 6a, we plot click-through rates for all users in the sample with at least a modest number of queries. We then further prune the data to those users with ten times as many queries, shown in Figure 6b (neither are log plots). Clearly, as the number of queries increases, the percentage of zero-click users increases. This is counter-intuitive if we limit the study to human users, since each query has a non-zero probability for clicking. However, if we consider automated traffic, we can reason about this increase; most bots do not need to click on the results.

Even in the case where the bot requires extended information about the URL target, the bot can be programmed to load this URL directly. Thus there are three typical bot click through rates; a bot that clicks on no links, a bot that clicks on every link, and a bot that only clicks on targeted links. Of these, the first is the most common by a wide margin.

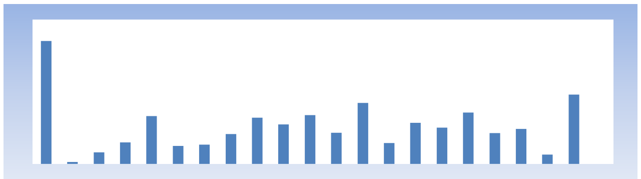

Figure 6a. Click-through rates, low minimum queries.

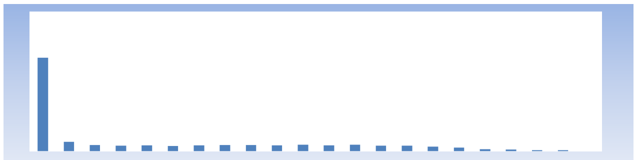

Figure 6b. Click-through rates, 10X minimum queries.

As an example, one user Id queried for 56,281 times without a single click. On the other extreme, a second user Id made 1,162 requests and clicked each time. Upon inspection of the queries, it appeared the user Id was downloading the html for each impression in the index for the keywords "168.216.com.tw." Also, the user Id in Section 4 who clicked on 1,874 out of 1,892 requests would also be discovered by this feature.

5.2.2 Alphabetical Ordering of Queries

We have identified a number of instances of bot-generated queries which have significant alphabetical ordering. It may be that the authors of the programs use the alphabetical ordering for improved searching or analyzing. When submitted to the search engine, it is quite detectable. Returning to the bots in Table 2 we witness this behavior. To calculate an alphabetical score for a user, we order the queries chronologically and for each query pair $< i, i + 1 >$, we add 1 if $i + 1$ sorts after i, and subtract 1 if $i + 1$ sorts before i. This number is then normalized by the total number of queries. In the majority of cases, the alphabetical score is near zero, as shown in Figure 7. The discretization $[-0.05, +0.05]$ contains more than 50% of the mass in the distribution. In almost all cases where the user Id has more than a couple queries and the alphabet score was outside $[-0.30, +0.30]$, we believed the traffic to be automated.

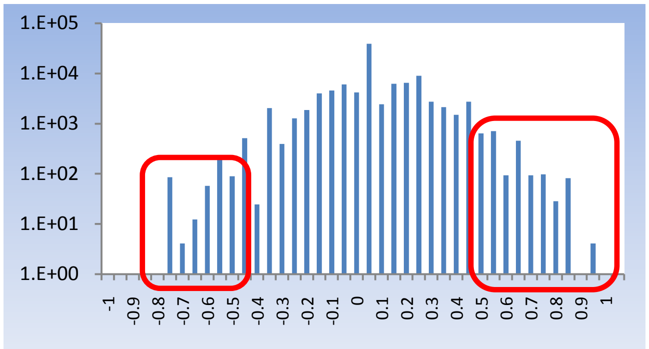

Figure 7. Alphabetical score.

5.2.3 Spam Score

Spam bots submit spam words to a search engine such as the queries shown in Table 1. Consequently, a feature which estimates the amount of spam words in the search queries can be useful for detecting queries from spam bots. We compute a spam score as a feature using a bag of $< spam\ word, weight >$ pairs for all queries for each user Id. The weight assigns a probability that a given keyword is spam. For example, the term "Viagra" has a higher probability of being spam than the term "coffee." In Figure 8 we show a normalized histogram of the spam score for queries received from individual cookies. The circled region in the histogram indicates user Ids submitting queries containing large numbers of spam terms. Per user scores are generated by summing the keyword spam score for their queries.

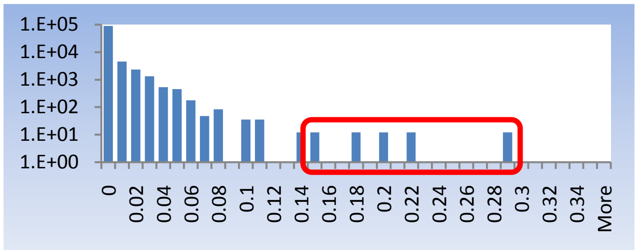

Figure 8. Spam score.

5.2.4 Adult Content Score

The adult entertainment industry has taken to the web with vigor. Many in this industry attempt to attract new customers by directing users to web sites containing pornography. Adult content enterprises may employ bots to measure the ranking of their website or try to boost their website's rank in the search engine. Although it is also a common human query space, there is lift in relative adult query counts. Thus, bot generated queries often contain words associated with adult content. As with the spam score, we use another bag of $< adult\ word, weight >$ pairs to compute an adult score for each user Id. A normalized histogram is presented in Figure 9 where we have circled the region in the figure which offers significant lift for bot detection. Examples of discovered bots for this feature are omitted due to space constraints.

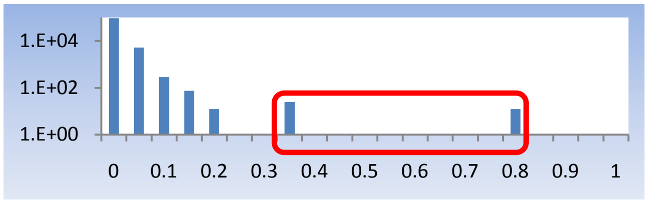

Figure 9. Adult content score.

5.2.5 *Query Keyword Entropy*

Many bots enter queries that are extremely redundant; as a result, bot queries tend to have keyword entropies which fall outside normal usage patterns. We calculate a map of $< word, count >$ pairs for each user Id. We then use traditional informational entropy, *H(k)*, to assign a score to each user

$$H(k) = E\big(I(k)\big) = -\sum_{i}\sum_{j} p\big(k_{ij}\big)\log_2 \mathrm{p}\big(k_{ij}\big)$$

where k_{ij} is the *j*th keyword (i.e. query term) in the *i*th query submitted a single user Id. In Figure 10, we plot the distribution of the entropy of keywords in the set of queries issued by users. In one example of a low keyword entropy bot, a user queried "*mynet*" 10,497 times, generating an entropy of zero. One could also consider the entropy of each query, without parsing it into keywords.

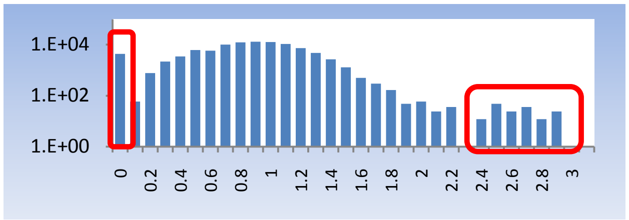

Figure 10. Query term entropy.

5.2.6 *Query Word Length Entropy*

Typical query terms have a natural word length entropy distribution, as does the length of a typical query. Some bots query for specific classes of words which are outliers of this distribution. For example, the word length entropy for the stock bot shown in Table 4 will have a lower word length entropy compared to that for a typical human. The word length entropy *WLE* is calculated as

$$WLE(l_{ij}) = -\sum_{i}\sum_{j} l_{ij}\log\,(l_{ij})$$

where *i* is the index for each separate query submitted to the search engine by a single user Id and l_{ij} is the length of the individual query term *j* in the *i*th query. The word length entropy is shown in Figure 11 (normalized to 1M users). One could also have as a feature the longest query in the session.

5.2.7 *Query Time Periodicity*

It is not uncommon for a bot to generate traffic at regular intervals, such as every 15 minutes [4]. To capture this property, we sort requests by request time for each user, and calculate the difference in time between successive entries. For each observed delta, we record the number of occurrences for each user. We then calculate the informational entropy of the deltas (a second option would be to calculate an FFT score for each user). This can be done at a variety of granularities for time deltas (seconds, 10 seconds, minutes, etc). The distribution for seconds can be seen in Figure 12. This feature can be used to investigate dwell time [1]. When combined with other features, such as the number of requests, it has the potential to provide significant lift. For example, a user Id with 30 queries may not appear automated based on request count alone, but if the entropy for time deltas is zero, it is much more likely to be a bot.

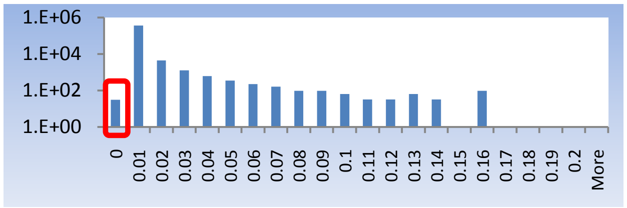

Figure 11. Keyword length entropy.

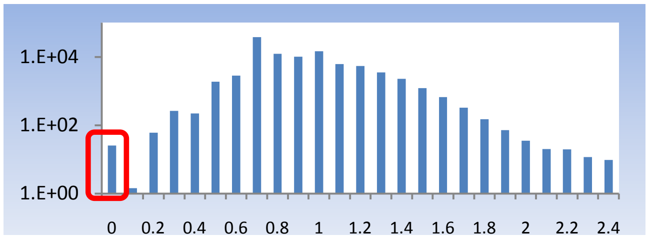

Figure 12. Period entropy.

5.2.8 *Advanced Query Syntax*

Some bots use advanced syntax to probe particular features of the index. For example, prefixing a query with "intitle:" for many search engines will force results to have the listed keywords as part of the title of the web page.

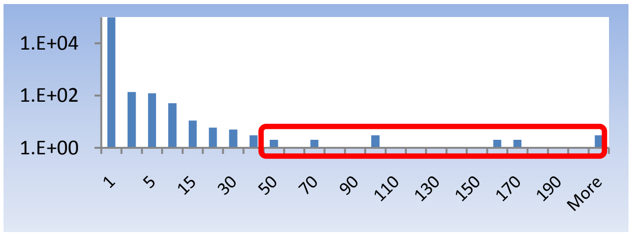

Figure 13. Advanced query terms.

Similarly, "inURL:" will restrict results to those URLs which have the keywords embedded into the URL. To discover bots which use advanced query syntax, we keep a total count of all advanced terms for each user throughout the day. A histogram is shown in Figure 13. Less than 1/10th of one percent of users use more than 5 advanced terms in the sample. As an example of a bot, one user had 110 queries, all of which requested the terms appear in the title of the web page.

5.2.9 *Category Entropy*

As a generalization of both adult content score and spam score, we can define a feature which captures the number of distinct categories associated with a user Id. We use a category hierarchy to assign a category to each query. We then track the category entropy for each user Id.

5.2.10 Reputations and Trends

There are several fields in the query logs that can directly identify known bot activity. Examples include blacklisted IP addresses, blacklisted user agents, and particular country codes. Tables are built for each property using domain expertise. For these cases, we simply perform a lookup into these tables at runtime. In less direct cases, query and query-click probability lists are used. For example, some bots search rare queries inordinately often. We often see sessions where each query is nonsensical. To detect these bots, a table of query-frequency pairs can be used to evaluate the popularity of the session's queries. Finally, a table of query-url click pairs can be stored to evaluate the probability that the user will click on a particular page. Users who often click on very low probability pairs are then deemed suspect. A potential weakness of these last two features is that a separate process is required to update the tables on a regular basis, and the tables can be somewhat large.

6. PRELIMINARY CLASSIFICATION

We now discuss preliminary results towards using the proposed feature set for classifying search traffic. We labeled 320 different user sessions, of which 189 were normal user sessions and 131 were automated sessions. This distribution is artificially skewed towards an equal distribution because we employed an active learner to choose which sessions to label (this active learner attempts to select sessions with the largest and smallest class margins.) It may be the case that we will need to train for the true prior for production purposes. Also, a larger set of labeled sessions would improve confidence.

We report classification results provided by the publicly available *Weka* toolset [12], as shown in Table 8. In all cases, we used 10-fold cross validation. We consider automated traffic labeled as automated traffic to be a true positive, noted as TP. Most of the classifiers chosen afforded greater than 90% accuracy on this small label set.

Classifier	TP	TN	FP	FN	%
Bayes Net	183	120	11	6	95
Naïve Bayes	185	106	25	4	91
AdaBoost	179	119	10	12	93
Bagging	185	115	16	4	94
ADTree	182	121	10	7	95
PART	184	120	11	5	95

Table 7. Classification results using proposed feature set (320 labeled data points).

We also used *Weka's* attribute evaluator to gain insight into the relative benefits of each feature, namely Information Gain using the Ranker search method. The top four features in order were query count, query entropy, max requests per 10 seconds, click through rate, and spam score, with ranks of 0.70, 0.39. 0.36, 0.32, and 0.29. As suspected, volume is a key indicator of present-day automated activity.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we provide an investigation of web query traffic received by a large-scale production search engine. Separating automated traffic from human traffic is useful from both a relevance as well as a performance perspective. A set of repurposeable features has been proposed to model the physical interaction of a user as well as the behavior of current day automated traffic. An analysis of the distributions of these features indicates they can be used as a basis for labeling user sessions accordingly. We are in the process of further assessing these features via classification, and are continuing to hand-label a large number of user Ids in the search query logs. For example, we are analyzing the lift for each feature when classifying a larger set of labeled sessions. Finally, we are investigating several avenues which may improve the proposed feature set, for example analysis over a longer time range (one month) and the evolution of user Id behavior.

8. ACKNOWLEDGMENTS

The authors would like to thank Zijian Zheng, Fritz Behr, and Toby Walker for their comprehensive logging and analysis tools.

9. REFERENCES

[1] E. Agichtein, E. Brill, S. Dumais and R. Ragno. Learning User Interaction Models for Predicting Web Search Result Preferences, in *Proceedings of the ACM Conference on Research and Development on Information Retrieval (SIGIR),* 2006.

[2] P. Anick. Using Terminological Feedback for Web Search Refinement - A Log-based Study. In *Proceedings of the SIGIR Conference on Information Retrieval* (Toronto, Canada, July 28 - August 1, 2003). SIGIR '03. ACM Press, New York, NY, 88-95.

[3] Click Quality Team. How Fictitious Clicks Occur in Third-Party Click Fraud Audit Reports, Google, Inc, 2006.

[4] N. Daswani, M. Stoppelman, and the Google Click Quality and Security Teams. The Anatomy of Clickbot.A, In *Proceedings of the USENIX HOTBOTS Workshop*, 2007.

[5] D. Fetterly, M. Manasse, and M. Najork. Spam, Damn Spam, and Statistics: Using Statistical Analysis to Locate Spam Web Pages. In 7th *International Workshop on the Web and Databases,* Paris, France, June 2004, pages 1-6.

[6] T Joachims, Optimizing Search Engines Using Clickthrough Data, In *Proceedings of the ACM Conference on Kowledge Discovery and Data Mining (SIGKDD),* 2002.

[7] T. Joachims, L. Granka, B. Pang, H. Hembrooke and G. Gay, Accurately Interpreting Clickthrough Data as Implicit Feedback, in *Proceedings of the ACM Conference on Research and Development on Information Retrieval (SIGIR),* 2005. Ls -l

[8] M. Kamvar and S. Baluja. A Large Scale Study of Wireless Search Behavior: Google Mobile Search. In Proceedings of the CHI Conference on Human Factors in Computing Systems (Toronto, Canada, April 22- 27, 2006). CHI 2006. ACM Press, New York, NY, 701-709.

[9] A. Karasaridis, B. Rexroad, D. Hoeflin. Wide-scale Botnet Detection and Characterization, In *Proceedings of the USENIX HOTBOTS Workshop*, 2007.

[10] T. Schluessler, S. Goglin, and E. Johnson. Is a Bot at the Controls? Detecting Input Data Attacks, in *NetGames*, 2007.

[11] A. Tuzhilin. The Lane's Gifts v. Google Report.

[12] Weka. http://www.cs.waikato.ac.nz/~ml/weka/

[13] K.-L. Wu, P. S. Yu, and A. Ballman. SpeedTracer: A Web usage mining and analysis tool. In *IBM Systems Journal*, Volume 37, Number 1, 1998