SRC Technical Note
1998 - 014

October 26, 1998

Analysis of a Very Large AltaVista Query Log

Craig Silverstein, Monika Henzinger,
Hannes Marais, and Michael Moricz

digital

Systems Research Center
130 Lytton Avenue
Palo Alto, California 94301

http://www.research.digital.com/SRC/

Craig Silverstein is at the Department of Computer Science, Stanford University, Stanford, CA. His electronic mail address is: csilver@cs.stanford.edu.

Michael Moricz is at AltaVista Search Service, 1825 S. Grant Street, Suite 410, San Mateo, CA 94402. His electronic mail adress is: moricz@pa.dec.com.

Abstract

In this paper we present an analysis of a 280 GB AltaVista Search Engine query log consisting of approximately 1 billion entries for search requests over a period of six weeks. This represents approximately 285 million user sessions, each an attempt to fill a single information need.

We present an analysis of individual queries, query duplication, and query sessions. Furthermore we present results of a correlation analysis of the log entries, studying the interaction of terms within queries.

Our data supports the conjecture that web users differ significantly from the user assumed in the standard information retrieval literature. Specifically, we show that web users type in short queries, mostly look at the first 10 results only, and seldom modify the query. This suggests that traditional information retrieval techniques might not work well for answering web search requests.

The correlation analysis showed that the most highly correlated items are constituents of phrases. This result indicates it may be useful for search engines to consider search terms as parts of phrases even if the user did not explicitly specify them as such.

1 Introduction

There is a large interest in finding patterns in and computing statistics of search engine query logs. However there are very few results that study the query logs of commercial search engines. One that does, by Jansen et al. [Jansen et al., 1998], studies only 51,000 queries, which is a small percentage of the number of queries asked on search engines each day.

Our goal is to study a larger number of search requests, collected over the course of several weeks. Such a large data set has several advantages. One, because it covers a large time range, it is less likely to be affected by ephemeral trends in querying (such as searches related to a new movie being released). For another, the extended time range allows us to identify individuals' patterns of use. Finally, by including all times of day as well as many days, we ensure that input from the entire world, and not just that portion which is awake during the period the log covers, is included in our results.[1]

Most of the questions about web use we choose to answer do not differ significantly from questions others have asked about query logs. We are interesting in counting statistics such as which queries are most common, what is the average number of words per query, how many queries are included in the average user session, and so on. In addition, however, we study correlations between query terms,

[1]Our results, however, cover only the logs of the main AltaVista engine and not of mirror sites outside the U.S.

and also among other field values. Thus, we seek to answer questions such as, “Do users from the U.S. ask disproportionately about basketball?” or, “Do people view more result pages when the query is about sex than when it’s about other topics?” We believe this is a fertile area for research, and present first steps in this direction.

The paper is structured as follows: Section 2 describes the AltaVista search environment and query log. Section 3 and 4 contain the analysis involving single items and the correlation analysis respectively. We conclude in Section 5.

2 The AltaVista Search Environment

The AltaVista search environment consists of several components that are important for analyzing queries: the engine itself, which is accessible from the web page `www.altavista.com`, and the query logs, which store information about what queries are made to the engine.

2.1 The AltaVista Search Engine

AltaVista is based on weighted boolean search. There are two major search modes: simple querying and advanced querying. A simple query consists of a collection of words, which are ORed together. A pair of operators allows for other boolean operations: `-word` instructs AltaVista to ignore documents containing `word` and is thus a NOT operator; `+word` instructs AltaVista to ignore documents not containing `word` and is thus an AND operator. The `"` operator is a proximity operator: words within double quotes must be adjacent in a document for the document to match the query. Quotes are therefore used to enclose phrases.

We use the term *query term* to denote a word or a quotation-mark-enclosed phrase.

An advanced query is more explicitly boolean. In advanced query mode, `and`, `or`, and `not` are interpreted as boolean operators rather than as search terms. Advanced queries may also include the boolean operator `near`, which is a relaxed form of the `"` operator: the words on either side of `near` must be close — but not necessarily adjacent — in a document for the document to match the query.

Both simple and advanced queries support web-specific operators. For instance, the query `host:xx.yy.zz` returns all documents found on the machine xx.yy.zz. To simplify our analysis, we ignore these operators and treat the query as a four-word query containing the words `host`, `xx`, `yy`, and `zz`.

The user can control aspects of the search in ways other than modifying the query. For instance, a pull-down menu allows the user to restrict result pages only to pages in a particular language. In the advanced search, an input box allows the user to restrict the results to pages last modified on a certain date, or within a range of dates.

After a query is entered and the various other restrictions processed, AltaVista returns a screen consisting of 10 URLs, with information about each URL such as the title. These URLs are ranked in order of relevance to the query, as determined by AltaVista's internal relevance function. The user may click on any URL to explore the associated web page. In addition, the user may click on navigation buttons to explore other screens of URLs. By clicking on "3", for instance, the user would be taken to result screen 3, which has the the 21st–30th most relevant URLs. Note this differs from some other search engines, where the user can jump to only the next 10 or previous 10 relevant URLs at any time.

2.2 The Altavista Query Log

The AltaVista query log has many components, only some of which concern us here. The query log is a text file consisting of a series of *requests*. A request may consist of a new query or a new result screen for a previously submitted query. Each request includes the following fields:

- A **timestamp** indicating when the query was submitted. The timestamp is measured in milliseconds since 1 January 1970.
- A **cookie**, which can be used to say whether two queries come from the same user (this field is blank if the user has disabled cookies);
- The **query terms**, exactly as submitted;
- The **result screen**, that is the requested range of search results;
- Other **user-specified modifiers**, such as a restriction on the result pages' language or date of last modification;
- **Submission information**, such as whether the query is a simple or advanced query; and
- **Submitter information**, such as the browser the submitter is using and the IP address of the submitting host.

2.3 Sessions

A *session* is a series of queries by a single user made within a small range of time. A session is meant to capture a single user's attempt to fill a single information need. By focusing on a single user, we attempt to separate two information needs being asked at the same time; by limiting the time gaps in a session, we attempt to separate two information needs by the same user asked at different times.

The use of cookies is integral to identifying sessions, because in theory every user has a unique cookie. Practice is of course somewhat crueller: different people using the same browser will share a cookie, and some users disallow cookies altogether. The former situation is not damaging for identifying sessions, since it is unlikely two separate users will be using the same browser simultaneously.

For those queries in which the user has disallowed cookies, we use the pair "domain IP/web browser used" as a substitute for the cookie. This is a poor substitute for cookies, particularly for large ISPs such as AOL, where tens of thousands of users can share a single IP address. Over 96% of all queries in our experiments have cookie information, however, so even a poor substitute discriminator of users is unlikely to bias the results. To ensure a lack of bias, we ran our experiments on two data sets: the full data set, and a data set restricted only to those queries that contain cookie information. The results in both cases were very similar, so we report only the experiments performed on the full data set.

In addition to identifying unique users, a good sessioning algorithm has to determine when a query starts a new information need. We use the heuristic that queries for a single information need come clustered in time, and then there is a gap before the user returns to the search engine. We used a cutoff of 5 minutes. This means that as a user is entering queries, as long as the latest query is submitted within five minutes of the previous query from that user, the new query is part of the same session. If the gap is more than 5 minutes, the new query starts a new session.

Since in reality a user may try to fill two information needs in one sitting, our session-identification heuristic almost certainly underestimates the true number of sessions.

2.4 The Query Log Data Set

Much of this paper is devoted to exploring the characteristics of the data set we studied; this introductory section is intended merely to give an introduction to the data.

The data set is summarized in Table 1. Already a few issues are apparent. For instance, fully 15% of all requests were empty requests, i.e. requests containing no query terms. Of the non-empty requests, 32% consisted of a request for a new result screen, while 68% (the 575,244,993 non-empty queries) consisted of a request for the first result screen of a new query. The queries were collected over 43 days, from 2 August 1998 to 13 September 1998. They include all the queries submitted to the AltaVista search engine over this time period.

Total number of bytes: 300,210,000,000	
Total number of requests:	993,208,159
Total number of non-empty requests:	843,445,731
Total number of non-empty queries:	575,244,993
Total number of unique, non-empty queries:	153,645,050
Total number of sessions:	285,474,117
Total number of exact-same-as-before requests:	41,922,802

Table 1: Statistics summarizing the query log contents used in our experiments. Empty requests had no query terms. A request consists of either a new query or a new requested result screen. Exact-same-as-before requests had the same query and requested result page as the previous request. The total number of unique, non-empty queries gives the cardinality of the set consisting of all queries.

2.5 Experimental Setup

Our experiments were performed on a Compaq AlphaServer 4100 with four 465 MHz. processors and 4 gigabytes of main memory, running OSF V4.0b. All code was compiled using `gcc 2.7.2` and compiled with the `-O6` optimization option. The experiments took about 22 hours of wall clock time to run. The duplicates analysis of Section 3.2, because of its large memory requirement, was done separately from the rest of the experiments; it took 18 additional hours of wall clock time to run.

3 First-order Analysis of Queries

First-order analysis of queries consists of all analysis that involves counting only single items, such as the frequency of query terms or the number of times a query has more than 20 result screens requested for it. This is in contrast to *second-order analysis*, which requires counting pairs of items, such as the number of times the terms `computer` and `free` occur in the same query.

The first-order analysis we performed falls into two categories: analysis of individual queries, and analysis of how queries are modified throughout sessions.

3.1 Analysis of Individual Queries

One contrast that was noted early in the history of web search is that searches on the web tend to have many fewer search terms than searches in more traditional information retrieval contexts [Jansen et al., 1998]. Though the gap has narrowed

0 terms in query:	20.6%	max terms in query:	393
1 term in query:	25.8%	avg terms in query:	2.35
2 terms in query:	26.0%	stddev of terms in query:	1.74
3 terms in query:	15.0%		
>3 terms in query:	12.6%		

Table 2: Statistics concerning the number of terms per query. Only distinct queries were used in the count; queries with many result screen requests were not upweighted. The mean and standard deviation are calculated only over queries with at least one term.

0 operators in query:	79.6%	max operators:	958
1 operator in query:	9.7%	avg operators:	0.41
2 operators in query:	6.0%	stddev of operators:	1.11
3 operators in query:	2.6%		
>3 operators in query:	2.1%		

Table 3: Statistics concerning the number of operators — `+`, `-`, `and`, `or`, `not`, and `near` — per query. Only distinct queries were used in the count; queries with many result screen requests were not upweighted.

over time, as we see in Table 2 the number of words per query is still small on average (2.35). The same average query length was found by [Jansen et al., 1998].

The number of operators used per query is also small, as is seen in Figure 3. One interesting note is that the maximum number of operators in a query is higher than the maximum number of terms found in a query. We attribute this to queries with a large number of dashes, which we interpret as operators in this context. Such a query could result from a user cutting text from another source and pasting it into the AltaVista search box.

3.2 Analysis of Query Duplicates

A different type of question concerning queries is how often an individual query is asked. We conjectured that a small set of queries is repeated many times over the course of a day. Indeed our data shows that the 25 most common queries form fully 1.5% of the total number of queries asked in a 43 day period, despite being only 0.00000016% of the unique queries. We give a listing of the 25 most frequent queries in Table 4. Note that uppercase and lowercase queries were not collapsed in the table.

Query	Frequency
sex	1551477
applet	1169031
porno	712790
mp3	613902
chat	406014
warez	398953
yahoo	377025
playboy	356556
xxx	324923
hotmail	321267
[non-ASCII query]	263760
pamela anderson	256559
p****	234037
sexo	226705
porn	212161
nude	190641
lolita	179629
games	166781
spice girls	162272
beastiality	152143
animal sex	150786
SEX	150699
gay	142761
titanic	140963
bestiality	136578

Table 4: The 25 most popular queries, and how often they were asked in the 43 day test period. Only distinct queries were used in the count; queries with many result screen requests were not upweighted. `p****` is a vulgarity.

Query occurs 1 time:	63.7%	max query frequency:	1,551,477
Query occurs 2 times:	16.2%	avg query frequency:	3.97
Query occurs 3 times:	6.5%	stddev of query freq:	221.31
Query occurs $>$ 3 times:	13.6%		

Table 5: Statistics concerning how often distinct queries are asked. Only distinct queries were used in the count; queries with many result screen requests were not upweighted. Percents are of the 154 million unique queries.

One surprising result is the frequency of the term `applet`. Examination of the logs shows that almost all queries containing the term were submitted by a robot.

In Table 5 we show part of a histogram of query duplication; that is, what percent of queries are asked only once, what percent are asked twice, and so forth. Counts are over all 43 days that we studied. We classified two queries as being the same if they had the same words with the same capitalization. (AltaVista is case sensitive.) We ignored word order and operators. That fact that almost two-thirds of all queries are asked only once in a 6 week period indicates that information needs on the web are quite diverse, or at least are specified in diverse ways.

Determining query duplication is expensive because it requires storing each unique query. For our analysis, with over a hundred million unique queries, it is infeasible to store each query in memory. Instead, we stored a one word *fingerprint* of each query, which is merely a hash value. We then stored the small fingerprint in a hashtable. This allowed us to count the frequency of each query but would not let us later say which queries were the most common. Therefore, when a query was seen more than 1000 times, we stored a map from the query to the fingerprint. Since almost all queries are asked only once, this reduced the memory requirement significantly.

Each fingerprint is 32 bits, so even with a hundred million queries it is unlikely two queries will have the same fingerprint value. Nevertheless, we possibly undercount slightly the number of unique queries, and overcount slightly the frequency of some queries, because of fingerprint collisions.

3.3 Analysis of Sessions

One of the most striking observation about sessions is most of them are very short. Fully 63.7% of all sessions consist of only one request: meaning only one query was entered and only one result screen was examined. Our analysis, unfortunately, does not enable us to determine how many of these simple sessions are so short because the information need was fulfilled so easily, because the user gave up in

1 query per session:	77.6%	max queries per session:	172325
2 queries per session:	13.5%	avg queries per session:	2.02
3 queries per session:	4.4%	stddev of queries/session:	123.40
$>$ 3 queries per session:	4.5%		

Table 6: Statistics concerning the characteristics of query modification in sessions.

1 screen per query:	85.2%	max screens per query:	78496
2 screens per query:	7.5%	(2nd most screens:	5108)
3 screens per query:	3.0%	avg screens per query:	1.39
$>$ 3 screens per query:	4.3%	stddev of screens/query:	3.74

Table 7: Statistics concerning the characteristics of result screen requests in sessions.

despair after seeing a single screen of results, or because the user was unaware of the usefulness of modifying the query or requesting further result screens.

Tables 6 and 7 reinforce the idea that sessions are often simple. However, on occasion a single query is modified hundreds of thousands of times, or thousands of result screens are requested. These numbers — like the large count for the term `applet` in Section 3.2 — are likely due to robots or other mechanized search agents. We found that the average number of queries per session is 2.02 and the average screens per query is 1.39. Jansen et al. [Jansen et al., 1998] report the average number of queries per session of 2.8, and average number of screens per query of 2.21. It is difficult to explain this discrepancy: either Excite users differ from AltaVista users, or Excite's relatively short query log was distorted.

For those situations where a query is modified, we see in Table 8 that in about 12% of the cases the query is modified by either adding or deleting terms or operators. In this situation, it is likely the user is satisfied with how the information need

Adding terms:	7.1%	(Adding *one* term: *5.4%*)
Deleting terms:	3.1%	(Deleting *one* term: *2.1%*)
Modifying operators only:	1.4%	(operators *everywhere*: *0.4%*)
Totally changing the query:	35.2%	
Otherwise modifying query terms:	53.2%	

Table 8: How queries are modified within a session.

is expressed, but the query is either too specific or not specific enough. Even when adding a word, a user may be trying to restrict the search space rather than broaden it, since often the word is added with a − operator, with the goal of removing non-relevant pages that share a common word.

In over half the cases, some query terms are deleted from the query and other terms added. In these cases it is likely the user is modifying the query not to change the scope of the query, but to restate the information need.

Over a third of the cases result in the a total change, where no word is shared between the two modifications. These may result in the information need being refined or even changed on the basis of information gleaned in the first search. In some cases, a total change may result from improperly identifying sessions. Unfortunately we don't know how many times this occurs.

4 Second-order Analysis of Queries

Second-order analysis of queries is analysis that requires joint counts, that is, counts for pairs of items in the same query. Unlike first-order analysis, which requires only linear space in the quantity being counted, second-order analysis requires quadratic space. Because of this, we limit our attention to query terms and fields for second-order analysis, and furthermore we limit ourselves to the 10,000 most common terms found in queries. (We include all field values, since there are only a small number). Because of these restrictions, the memory requirement does not grow with the size of the data set, and it is possible to do second-order analysis on large data.

Our second-order analysis consists of finding correlations between items, where an *item* is either a query term, such as `"new york"`, or a field value, such as `language=fr` (which indicates the user specified only pages written in French should be returned). In addition to studying 2-way correlation, we devise a method for augmenting our correlation analysis to find 3- and 4-way correlations, while keeping the space use quadratic.

Since we limit our attention to the 10,000 most common terms, many queries become empty because they contain only rarer terms. We limit our attention only to queries with at least one term in the top 10,000 — for this data set, 313,454,867 distinct queries.

4.1 The Chi-squared Test for Correlation

While in reality queries consist of items, for the purpose of discussing correlation we consider the dual formulation that items decide whether or not to be in queries.

We define $P[a]$ to be the probability that item a is in any given query. Likewise, $P[ab]$ is the probability a query will contain both items a and b.

Two items a and b are said to be *independent* if

$$P[ab] = P[a] \times P[b].$$

Two items are said to be *correlated* if they are not independent.

At times it is useful to use the terminology $P[\overline{a}] = 1 - P[a]$, the probability that a does *not* occur in a query.

The observational equivalent of $P[]$ is $O()$, the number of queries *observed* to contain a set of items. For any given argument to $O()$, the value can be easily calculated by iterating over the log. If there are a total of n queries in the log, it makes sense to estimate $P[a]$ by the maximum likelihood estimator

$$E(a) = O(a)/n.$$

Here we use the terminology $E()$ to mean "the maximum likelihood estimate of $P[]$ for the given log." Estimating $P[ab]$ is as straightforward as estimating $P[a]$: $E(ab) = O(ab)/n$. $P[\overline{a}]$ can be estimated by $O(\overline{a})/n$, the number of queries not containing a.

If a and b are independent, we can use the definition of independence to get another estimate of $P[ab]$, namely $E(ab) = E(a)E(b) = O(a)/n \times O(b)/n = O(a)O(b)/n^2$. Therefore, it makes sense to say that a and b are independent if and only if $O(a)O(b)/n^2 = O(ab)/n$.

Because observational data is only a sample of the presumed probability distribution $P[]$, it may be that $O()$ and $E()$ do not exactly match in the data set, even if the two variables are actually independent. The work of [de Moivre, 1733] and [de Laplace, 1878] found that there is a relationship between $O()$ and $E()$ that depends only on the size of the data and can be used to characterize the probability of independence. This relationship is characterized by the *chi-squared* statistic:

$$\begin{aligned}\chi^2(a,b) &= \frac{[E(ab) - O(ab)]^2}{E(ab)} + \frac{[E(\overline{a}b) - O(\overline{a}b)]^2}{E(\overline{a}b)} + \frac{[E(a\overline{b}) - O(a\overline{b})]^2}{E(a\overline{b})} \\ &+ \frac{[E(\overline{a}\overline{b}) - O(\overline{a}\overline{b})]^2}{E(\overline{a}\overline{b})}\end{aligned}$$

χ^2 measures a (normalized) deviation from expectation under the independence assumption, but it calculates the deviation in terms of non-occurrence as well as occurrence.

The χ^2 value has a chi-squared distribution. (In particular, for boolean data it has a chi-squared distribution with one degree of freedom.) Looking up values for the chi-squared distribution in a table, we see that only 5% of the time does the χ^2 value exceed 3.84 if the variables are actually independent. Therefore, if we see $\chi^2 > 3.84$, we say that a and b are correlated at the 95% confidence level.

4.2 The Correlation Coefficient

While the chi-squared test can detect the *presence* of correlation, it cannot judge its *strength*. For instance, if two items are actually correlated, their χ^2 value will grow as n, the size of the data, grows. The *correlation coefficient*, ρ, is typically used to measure the strength of correlation. It is defined as follows:

$$\rho(a, b) = \frac{\left(\sum_i (A_i - \mu_a)(B_i - \mu_b)\right)}{\sqrt{\sigma_a^2 \sigma_b^2}}$$

where μ_a is the mean value of item a (that is, $O(a)/n$), σ_a^2 is the variance of a, and A_i is 1 if and only if query i contains a. The denominator is a scaling factor that keeps ρ between -1 and 1. For boolean data, ρ will equal 0 if A and B are independent, 1 if they are perfectly correlated, and -1 if they are perfectly negatively correlated.

The correlation coefficient is important for data mining applications because there is often enough data to find weak but significant correlations. The correlation coefficient can help the analyst concentrate on those correlations that are strong enough to make an impact. For instance, suppose that once a year a user enters a query `volcano baklava`, but otherwise the two terms are independent. With enough data, the chi-squared test will (correctly) say there is a correlation, but the correlation coefficient will be low, indicating that the correlation is weak. An analyst would be wise not to change policy based on this correlation.

It is clear, then, that χ^2 and ρ are complementary measures, and one does not supersede the other. In particular, ρ can have an extreme value even if the χ^2 value is below the 95% confidence threshold, if the correlation is strong but there is not enough data to say it is statistically significant. Likewise, the χ^2 value could be high and the correlation coefficient near 0 if there is enough data to establish the correlation, even though it is weak.

Despite the differing roles of χ^2 and ρ, for boolean data such as we have for query log analysis, the two are related. In particular,

$$\chi^2 = n \times \rho^2.$$

This relationship means that for boolean data, the strongest correlations are also the most likely to be true correlations rather than statistical anomalies. For mining

Term A	Term B	$O(AB)$	ρ
`cindy`	`crawford`	118558	0.7098
`persian`	`kitty`	75716	0.6830
`pamela`	`anderson`	453467	0.6451
`visual`	`basic`	177971	0.6325
`www`	`http`	2355010	0.3862
`buffy`	`slayer`	12340	0.3989
`slayer`	`vampire`	13640	0.3088
`buffy`	`vampire`	12986	0.2766
`lang=ko`	`domain=kr`	1030416	0.7281
`date=restricted`	`applet`	1165565	0.7273
`referred=yes`	`sessmodlen=4+`	76257842	0.6388
`referred=yes`	`sessmod=restart`	46359290	0.5482
`the`	`qwords=6+`	2417838	0.1886

Table 9: Some highly correlated pairs of items. The data set consists of about 10,000 items and 313,000,000 queries.

query logs, this is an important fact, because with over 10,000 items there are more than 100 million statistical tests being done. Even at the 95% confidence level, there will likely be made millions of false judgements of correlation. By concentrating on items with correlation coefficient above 0.2, however, we bring the expected number of false judgements to less than one.

4.3 Results for Two-way Correlation

We calculated correlation between items, consisting of 10,000 terms and 662 field values. Field values are boolean items of the form `field=value`. For instance, `domain=nl` is an item in all queries emanating from the Netherlands. After throwing out trivial correlations, such as `domain=nl` being negatively correlated with `domain=kr`, we were left with 110 correlated pairs. Some of these are displayed in Table 9.

There are several interesting things to note about the highly correlated pairs. One is that the most highly correlated items are constituents of phrases. This indicates there are many situations where users are not using the `"` operator to tell AltaVista that the separate words constitute a single term.

One phrase that contributes to three of the entries in Table 9 is "Buffy the Vampire Slayer," a TV show. While presumably most queries containing one of these words contained all three, `buffy` and `slayer` are more highly correlated than pairs containing `vampire`. This is because terms like `buffy` and `slayer` are unlikely to occur in queries in other contexts, making the strength of correlation high. However, there are likely many queries concerning vampires but not the TV show, making the correlation less strong in this case.

One of the most frequent of the highly correlated queries is between `www` and `com`. These terms are likely to occur in two contexts: when the user is using an operator like `link:` that takes a URL, and when a user enters a URL in the search engine. The fact that `www` and `com` are highly correlated is hardly surprising: both often occur in a URL, and neither is likely to be a common query term when not searching for a URL. This is an extreme example of the general fact that text is naturally correlated. This makes data mining text more challenging than data mining other data sets.

The correlation between people from Korea querying and asking for result pages in Korean is hardly surprising. As mentioned before, the term applet is often queried by a robot, which explains the correlation between the word "applet" and requesting a date-restricted set of pages.

One interesting set of correlations concern "referred" users, which are users sent to AltaVista by an AltaVista partner, rather than those who typed in the URL explicitly or came from a bookmark. These users are correlated highly with sessions with many modifications, perhaps indicating these users spend more time than most modifying their query. They are also particularly likely to "restart" their queries, meaning they enter a query with no terms in common with their previous query. This may indicate users sent to AltaVista to fill one information need get distracted and try to fill others as well. Another possible explanation for both data points is that cookie information does not get forwarded from all the referring sites, meaning we are misidentifying sessions in this case.

A final, unsurprising result is that the term `the` is correlated with long queries. It is likely that queries containing such a contentless word are natural language queries.

4.4 Results for Three- and Four-way Correlation

If a pair of items is correlated, then by definition all supersets of that pair are correlated as well. Therefore, to find correlations involving more than two items, it is necessary to find pairs of items that are uncorrelated. In the case of text this is difficult, since text is naturally correlated. Therefore, typical methods of finding large correlated sets have proven unsuccessful for mining queries in query logs.

In order to find these larger correlations, we implemented a heuristic that "phrasifies" highly correlated terms. In this case, we took the 1000 item pairs with the largest (positive) correlation coefficient, as determined from the analysis in the

Term A	Term B	$O(AB)$	ρ
(links, kitty)	(persian, adult)	40375	0.9438
(www, http)	(http, com)	127783	0.9353
(harvard, business)	review	12554	0.9215
(used, car)	(used, prices)	24702	0.8988
(bluemountain, com)	(www, bluemountain)	36722	0.8871
(anderson, lee)	(pamela, lee)	105208	0.7375
(ibm, video)	(highlander, newsgroups)	5261	0.5037
(persian, kitty)	(persian, links)	45322	0.4687

Table 10: Some highly correlated items. By "phrasifying" highly correlated pairs — represented by "(word1, word2)," — we obtain sets of three and four highly correlated items. The data set consists of about 10,000 items and 244,000,000 queries.

Section 4.3. For each pair, we constructed a new term consisting of that pair of items. A query contains the new term only if it contains both items in the pair. Effectively, we are pretending the correlation coefficient of the pair of items is 1 — meaning they only occur together — so the correlation of the pair with a third item is equal to the correlation of the pair with either of the items individually. In reality, since the correlation of the pair is less than 1, we approximate but underestimate the true correlation between the three items.

By repeating the same analysis as was done in Section 4.3 with the 1000 new items, we can find triples of items that are correlated, or even sets of four if two pairs are correlated. The results are shown in Table 10. Most of the 3- and 4-way correlations involve phrases with three and four words in them, such as "Harvard Business Review" and "Persian Kitty adult links." A connection such as "IBM video" and "highlander newsgroups" seems more obscure, and indeed the low frequency indicates it is probably the result of a small number of users repeatedly modifying a query with these four words as the base.

For multi-word phrases, the way in which the words are grouped can affect the correlation score. For instance, (persian, kitty, adult, links) has a fairly weak correlation when one of the pairs is (persian, kitty). This is because the phrase persian kitty is often asked alone, as a single two-word query; it therefore correlated relatively weakly with adult and links. On the other hand, the pairs (kitty, links) and (persian, adult) obviously each only occur when the full query persian kitty adult links is entered.

Similary, (www, bluemountain, com) is only correlated when considering the pairs in Table 10. The correlation between bluemountain and (www, com) is very low.

Given these observations, it makes sense to consider a triple of words to be correlated only if all 3 pairwise partitions of the words yield strong correlations.

5 Conclusions and Further Research

In this paper we presented an analysis of an AltaVista query log containing almost 1 billion entries. We confirmed the conjecture that an average web user differs significantly from the user model assumed by the information retrieval community. Surprisely, for 85% of the queries only the first result screen is viewed, and 77% of the session only contain 1 query, i.e., the queries were not modified in these sessions.

In the correlation analysis we considered the queries of all users and found the strongest correlations resulted from short queries that were actually single-term phrase queries. For future work it might be interesting to restrict the correlation analysis to long queries, with for instance more than ten terms, in hopes of finding correlations between concepts rather than merely terms.

Our analysis did not distinguish between requests issued by humans and requests issued by robots. In some cases saw, robot-initiated queries in the AltaVista logs resulted in a number of seemingly bizarre results. It is impossible for us to judge how much automated search techniques skewed the results of this study. It would be interesting to repeat the analysis with the request from robots discarded, assuming a method could be found to distinguish requests by robots from requests by humans.

References

[de Laplace, 1878] de Laplace, P.S. 1878. Oeuvres complétes de Laplace publiées sous les auspices de l'Académie des Sciences par M.M. les secrétaires perpétuels, Gauthier-Villar, Paris, 1878/1912.

[de Moivre, 1733] de Moivre, A. 1733. Approximatio ad summam terminorum binomii $(a + b)^n$ in seriem expansi, Supplement to Miscellanea Analytica, London.

[Jansen et al., 1998] Jansen, B.J., Spink, A., Bateman, J., and Saracevic, T. 1998 Real Life Infomration Retrieval: A Study of User Queries on the Web, SIGIR FORUM, 32 (1):5–17.