RJ 10199 (95073) November 15, 2000
Computer Science

01A 00019

Research Report

WEB QUERY CHARACTERISTICS AND THEIR IMPLICATIONS ON SEARCH ENGINES

Jason Y. Zien
Jorg Meyer
John Tomlin

IBM Research Division
Almaden Research Center
650 Harry Road
San Jose, CA 95120-6099

Joy Liu

U.C. Berkeley
Berkeley, California

LIMITED DISTRIBUTION NOTICE

This report has been submitted for publication outside of IBM and will probably be copyrighted if accepted for publication. It has been issued as a Research Report for early dissemination of its contents. In view of the transfer of copyright to the outside publisher, its distribution outside of IBM prior to publication should be limited to peer communications and specific requests. After outside publication, requests should be filled only by reprints or legally obtained copies of the article (e.g., payment of royalties). Copies may be requested from IBM T. J. Watson Research Center, P. O. Box 218, Yorktown Heights, NY 10598 USA (email: reports@us.ibm.com). Some reports are available on the internet at http://domino.watson.ibm.com/library.CyberDig.nsf/home.

IBM **Research Division**
Almaden • Austin • Beijing • Haifa • T. J. Watson • Tokyo • Zurich

RJ 10199 (95073) November 15, 2000
Computer Science

WEB QUERY CHARACTERISTICS AND THEIR IMPLICATIONS ON SEARCH ENGINES

Jason Y. Zien
Jorg Meyer
John Tomlin

IBM Research Division
Almaden Research Center
650 Harry Road
San Jose, CA 95120-6099

Joy Liu

U.C. Berkeley
Berkeley, California

Abstract

The rapid growth of the World Wide Web (WWW, Web) presents significant challenges to the implementers of high performance Web search engines. There are two dimensions to this growth. The first challenge manifests itself in the rapidly growing number of web pages, which leads to the problem of indexing terabytes of data. The second problem is the growing population of web users which will place high demands on the query load for a popular search engine. The most effective way of dealing with high query loads is through efficient caching. In addition, it is clear that understanding the search behavior of users is critical to caching strategies and to the overall design of a web search engine.

In this paper, we will study the characteristics of a large query log and perform an analysis of the web queries. We then assess the impact of these queries on search engine caching and performance. In particular, we will look at the properties of vocabulary growth, term occurrences and term frequencies, and illustrate how these properties should be taken into consideration when designing a search engine. We also analyze the temporal qualities of search queries and classify them into three categories: hot, popular, and unpopular.

Keywords: query, logs, search, engine, caching, Web, performance

Web Query Characteristics and their Implications on Search Engines

Jason Y. Zien, Jörg Meyer, John Tomlin, Joy Liu *
IBM Research Division
Almaden Research Center
650 Harry Road
San Jose, CA, 95120-6099
{jasonz, jmeyer, tomlin}@almaden.ibm.com

November 11, 2000

Abstract

The rapid growth of the World Wide Web (WWW, Web) presents significant challenges to the implementers of high performance Web search engines. There are two dimensions to this growth. The first challenge manifests itself in the rapidly growing number of web pages, which leads to the problem of indexing terabytes of data. The second problem is the growing population of web users which place high demands on the query load for a popular search engine. The most effective way of dealing with high query loads is through efficient caching. In addition, it is clear that understanding the search behavior of users is critical to caching strategies and to the overall design of a web search engine.

In this paper, we will study the characteristics of a large query log and perform an analysis of the web queries. We then assess the impact of these queries on search engine caching and performance. In particular, we will look at the properties of vocabulary growth, term occurrences and term frequencies, and illustrate how these properties should be taken into consideration when designing a search engine. We also analyze the temporal qualities of search queries and classify them into three categories: hot, popular, and unpopular.

Keywords: query, logs, search, engine, caching, Web, performance
Approximate word count for the abstract: 200
Approximate word count for entire paper: 5000

1 Introduction

The rapid growth of the World Wide Web (WWW, Web) poses significant problems to the designers and implementers of high performance Web search engines. There are two major challenges. First, the rapidly growing number of web pages requires search engines to index large amounts of information- currently up to several terabytes of data. Search engines are still catching up with the growth of the web. Figure 1 show that search engine sizes (number of web pages indexed) are growing at a rate of over 100% a year. The figures show statistics on the estimated size of search engines from [9]. Each point in the graph gives the average size of four major search engines (Fast, Google, Northern Light, and AltaVista) over an 18 month period. The second dimension of the Web growth is the rapidly increasing number of web users which place high demands on the query load for a popular search engine. From 1997 to 1999, in the U.S. and Canada, the population of web users grew at 46% a year [4]. Currently, there are an estimated 286.9 million total Internet users in the world [8]. The most effective way of dealing with high query loads is through efficient caching

*U.C. Berkeley

that provides a high hit rate. The importance of understanding the search behavior of users is critical to the design of a web search engine and its caching strategies.

In this paper, we will study the characteristics of a large query log and perform an analysis of the web queries. We then assess the impact of these queries on search engine caching and performance. In particular, we will look at the properties of vocabulary growth, term occurrences and term frequencies, and illustrate how these properties should be taken into consideration when designing a search engine. We also analyze the temporal qualities of search queries and classify them into three categories: hot, popular, and unpopular.

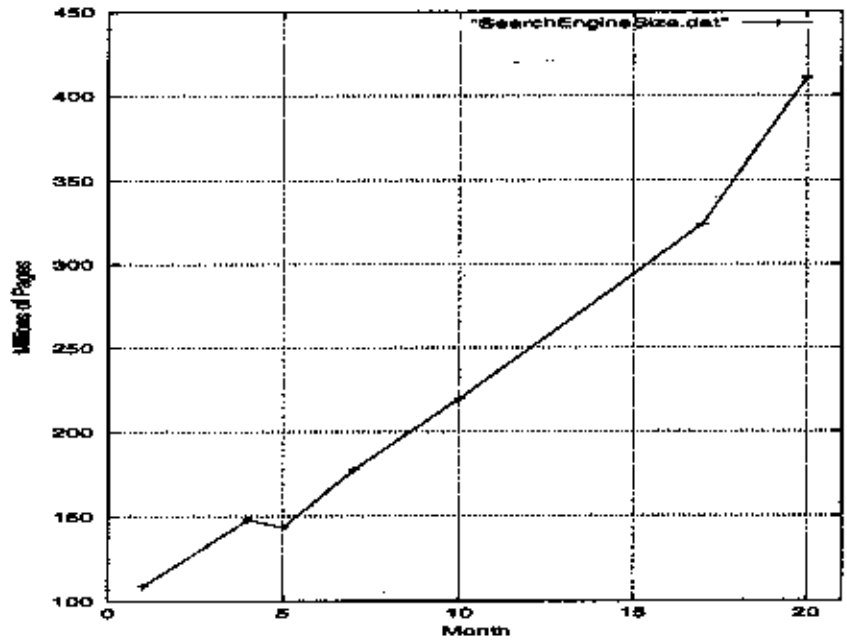

Figure 1: Growth of search engine sizes from May 1999 (month 1) through October 2000 (month 18).

1.1 Prior Work on Query Analysis

There have been several major results in the literature related to web query log characteristics and query caching. Spink et al. report on a study of 357 Excite users in [12]. This survey was designed to study the search strategies and habits of web search users. They found that the average search query contained 3.34 terms (not including stop words). Also of particular interest, most search users had no plans to use any of the more sophisticated features of the search engine (such as booleans, the + or - query modifiers, or phrases. They conclude that users lack the motivation to compose complex queries, or learn correct syntax and rules for querying.

The topic of query *result caching* is studied by Markatos in [7]. The author obtained a trace of one million search queries from the Excite search engine from September 16, 1997. A study of several cache replacement policies found that the *Least Recently Used* (LRU) method was consistently the worst method because it does not take frequency of access into account—unlike the other methods examined. The main result of this study was that there was very little locality in queries. Using a trace-driven cache simulator, the author found that even with large caches (several hundred megabytes to several gigabytes), the maximum hit rate would reach around 25% due to the poor query locality. Markatos does note, however, that this may be a partly due to the short one-day time duration of their query logs.

Silverstein et al[11], analyzed a large query log containing one billion queries gathered over six weeks from the

AltaVista search engine. They found that 63.7% of all the distinct queries occurred once, 16.2% occurred twice, 6.5% occurred tree times, and 13.6% of the queries occurred more than three times. They ignored word ordering and query operators but did use a case-sensitive comparison. The fact that 63.7% of all queries over that time period were distinct certainly demonstrates the diversity and lack of locality of web queries. Another important finding in this work is that for 85% of the queries, users only viewed the first query result page and that 77% of the time, users only entered one query for the session. Most users did not have a very long attention span or try to optimize and/or modify their queries. A final interesting observation was that in a two-way correlation test, the most highly correlated items were common phrase constituents (such as names- "cindy crawford" or "pamela anderson" for example) yet searchers did not use the quoted phrase operators in their queries.

Lu [6] has conducted a very interesting study of two search engine query logs. These were the Excite query logs from September 16, 1997, and the THOMAS query logs from July 14 to September 13, 1998. The THOMAS logs were provided by the Library of Congress. They are the queries from a search engine which indexes the full text of the Congressional Records and all bills introduced in the 101st - 105th Congress. The THOMAS logs averaged 5928 queries per day and had an average of 1.9 terms per query (for the 1000 most popular unique queries). This study examined query locality by grouping queries into groups, called topics. All queries whose top 20 document results overlapped were considered to be in the same topic. In the THOMAS logs, the top 2.5% (top 100) topics accounted for 21.2% of the queries and the top 12% (top 2000) of topics accounted for 41.5% of the queries indicating a high degree of locality. In the Excite logs, the top 2% (top 500) queries accounted for 12.3% of the queries while the top 10% (top 20000)topics accounted for 42% of the queries, also indicating a high degree of locality. Lu also studied daily and weekly overlaps of the THOMAS logs. On average, 43.1% of all queries for a particular day overlapped with the previous day while 57.8% of the queries overlapped with the previous week. Lu concludes that a good query topic caching scheme could have a 20% to 80% hit rate.

2 Query Log Details

We define a web query as the exact string typed in by a user searching for data using a search engine. A query may contain one or more terms as well as special query operators. In this section, we describe how we gathered the web query logs used in this study. In addition, we present a high-level analysis of the important characteristics of these web queries.

2.1 WebCrawler Queries

Our queries were gathered from www.webcrawler.com from March 22, 2000 to May 26, 2000 using their SearchTicker (http://www.webcrawler.com/cgi-bin/SearchTicker). We gathered the queries by obtaining snapshots at one minute intervals and timestamped each snapshot. Each snapshot could have from tens to hundreds of queries, and we checked for and removed the overlaps of the tail of a snapshot with the head of the next snapshot since there might be duplicates. We have no user tracking information and no way to tie queries to particular users or sessions at all- only the raw queries and approximate timestamps.

2.2 Query Characteristics

There were a total of 50,538,653 queries in the log and 165,763,490 terms. Table 1 shows the number of terms and the number of queries having that many terms. Each query had an average of 3.3 terms (we did not filter out stop words) which is roughly the same as the 3.34 found by [12] (though they did filter out stop words) though significantly higher than the 2.2 term average reported by Kirsch [5]. The reason for the large average number of query terms is the heavy influence of sites which encourage natural language questions, such as AskJeeves.com. In fact, www.webcrawler.com is one of the search engines used by the AskJeeves

meta-search engine, which likely accounts for the large number of natural language questions observed in the query log. We found that there were 8.97 million queries (17.9%) beginning with the terms: who, what, where, when, why, and how. This presents a challenge to search engines since the intent of the query must be inferred (without the benefit of special search query operators).

A significant number of queries (35.6%) included the use of special query operators: 11,635,328 queries (22.9%) contained the '+' operator, 1,189,358 queries (2.3%) included the '-' operator, and 5,272,111 queries (10.4%) included quotation marks for phrase searches. This is a surprisingly large number, as Kirsch [5] reported that only 10% of queries made use of special operators. The extensive use of query operators is not due to more sophisticated users, but rather, due to a more flexible search engine user interface. During the time period these queries were gathered, the user interface contained a pull-down menu allowing users to require all the terms or search for the exact phrase. The importance of machine-assisted query formulation could play an even more important role in improving the quality of search engine results in the future. Simple user interface changes may dramatically alter the final query received by the search engine and thus have a significant effect on the results.

Terms	Queries	Terms	Queries
1	11354335	17	51756
2	13056192	18	27930
3	8853696	19	16478
4	4154563	20	10644
5	2916638	21	8187
6	2005013	22	6036
7	2191455	23	4769
8	1739357	24	3397
9	1434374	25	3302
10	965651	26	2356
11	634401	27	2325
12	373585	28	2249
13	242518	29	1690
14	141037	30	1731
15	93128	31+	12752
16	55382	-	-

Table 1: Terms per Query

3 Performance Issues

We examine a number of performance issues in Web search based on the user query logs. Many factors contribute to search performance, including the index size, index postings list distributions, query term patterns, and document result set distributions.

3.1 Vocabulary Growth

The vocabulary used by web searchers has profound implications on the optimization of a search engine. There are two major optimizations that can be performed based on studying user search vocabulary. The first is the reduction of the index size by pruning useless terms in the index and the second optimization is the caching of frequently used terms or queries to improve performance. It would be ideal if the vocabulary (either the terms or the exact queries) used by search engine users stayed within some small subset or grew very slowly. Previous studies have looked at the vocabulary growth of terms in document collections. Let

V be the vocabulary size, n be the number of vocabulary elements (tokens, terms, query strings, etc.) in a data set, and K and β be constants which are dependent on the particular text of the data set. It has been generally observed that vocabulary size grows according to the following formula: $V = Kn^{\beta}$ where β is typically between 0.4 and 0.6, so vocabulary in a document collection typically grows proportionally to the square root of the words in the documents. This is known as Heaps' Law [1]. In [3], β was found to be 0.66 for the TIPSTER document collection.

Whether the vocabulary of web queries obeys Heaps' law or not is not obvious, since queries are typically very short, and there tend to be many popular queries that are often repeated. We analyzed our web query logs to determine the growth of query vocabulary. All 50 million queries were gathered in one large file and data points on vocabulary growth were gathered by processing successively larger prefixes of that file. We considered two cases:

1. Each complete user query was taken to be a single word in the vocabulary. This case is useful to consider because it is desirable to cache the results for an exact query with the hope that it will be repeated in the future, completely avoiding the query evaluation phase.
2. Each term in the query (ignoring capitalization and ignoring modifiers) was considered to be a single word in the vocabulary. This case is useful to consider because of the implications it has on search engine I/O performance. Typically, the postings for each term in the query must be fetched from disk during the query evaluation phase. If postings for terms are cached, then we can avoid the overhead of doing disk I/O during query evaluation. In addition, if we are able to determine which terms in the index are never used, we may be able to prune the dictionary and postings lists of an index with minimal impact on the user.

Figure 2 shows a plot of the vocabulary growth on a log-log scale. The upper curve is a plot of the total number queries versus the unique queries. This curve is based on the exact queries that users typed in, using a case-sensitive, exact string comparison. Each query is considered to be one vocabulary word. The curve is almost linear (a curve fit gives the equation $V = 1.53n^{0.95}$). This is very discouraging news, as it means that almost all of the queries are unique in our query logs and that even an optimal infinite cache could only hope to achieve a 42% hit rate. This agrees with the observations of [7] that query caching hit rates should be between 25% to 75% depending on the size of the query traces being examined. This result is not surprising given the extensive natural language queries and special query operators in the data set. The number of distinct queries made up 58% of the total queries asked. Clearly, caching of exact query results will be only a minor performance win due to the low hit rates.

The lower curve plots the total number of query terms versus the number of unique query terms. In this case, all operators were stripped from the terms and a case-insensitive comparison was done. This curve is much more promising- we see that it has a much lower slope (a curve fit gives the equation $V = 6.63n^{0.69}$), growing just slightly faster than the square root of the text size. The number of distinct terms made up 1.8% of the total terms so an optimal, infinite cache could achieve a 98.2% hit rate. There is clearly great potential for the caching of postings lists for individual query terms in a search engine, though the effectiveness of any particular algorithm will depend upon the distribution of the query terms. For instance, if query terms all occurred with uniform frequency, uniformly spaced in time, an LRU cache would perform poorly. We will later see that this is not the case.

One disappointing fact learned from the curve, however, is that there does not seem to be a plateau on the growth of the curve. The vocabulary used in query terms is not restricted to a small set, but rather continues to grow as the number of queries grows. Although this might indicate that pruning terms from the search engine dictionary and index would be detrimental, in the next section we will see that a more detailed examination of the query occurrence frequencies shows otherwise.

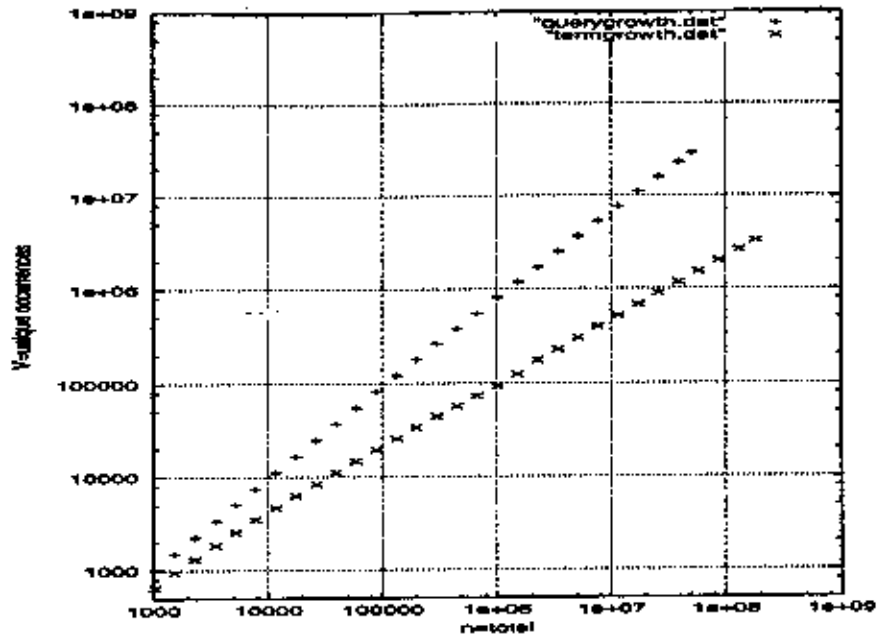

Figure 2: Query vocabulary growth (upper curve) and term vocabulary growth (lower curve).

3.2 Postings Size Distribution

The vocabulary of web query terms provides us with a good picture of the diversity of query terms, however, if we examine the query term occurrence frequencies and their corresponding document occurrence frequencies, we can gain additional insight into the relative importance of different query terms.

For instance, do people most frequently ask about the terms that are the most written about on the web? Are terms that occur infrequently on the web also terms that are infrequently asked in queries? Table 3 shows the importance of the most frequently occurring queries. The first column in each row gives the total number of the top most frequently occurring terms being considered, the second column shows the total percentage of the query terms that they make up, and the third column shows the percentage of the unique terms they make up. Clearly, the most frequently occurring constitute an extremely important subset of the vocabulary. The top 3.2% (100,000) terms make up over 95.1% of the total terms in the queries. The implication is, dropping a substantial number of terms in a search engine's dictionary and index should have relatively little effect on the user. Although the list of most frequently occurring query terms tells us which terms to keep, picking which terms should be dropped is more difficult, as some terms may be intermittently popular.

Top X Terms	% of Total	% of Distinct
10	20.5	.00032
100	35.9	.0032
1,000	58.1	.032
10,000	83.5	.32
100,000	95.1	3.2
1,000,000	98.7	32.0

Table 2: Percentage of terms accounted for by the most frequent query terms.

With such a small percentage of query terms accounting for such a large percentage of the total queries,

the possibility of caching postings lists for query terms looks promising. A Least Recently Used (LRU) caching algorithm would seem to be a good way of exploiting the frequent occurrences of query terms. We implemented an LRU caching scheme and experimented with various cache sizes. A cache size of x is used to indicate that the cache can the most recently encountered x terms. This is a simplification of the actual problem because in a real system, you would cache postings lists associated with a term. The sizes of the postings associated with terms may vary dramatically. The results however, show that an LRU cache can potentially have hit rates above 97% given a sufficiently large cache. Since postings are fetched from disk, we must note that typically, the operating system already performs LRU caching of disk blocks through a buffer cache. Thus, even without explicit caching of postings, we expect good performance in the retrieval of postings lists due to the operating system buffer cache.

Cache Size	Hit Rate
10	5.2%
100	26.6%
1,000	49.4%
10,000	78.8%
100,000	93.1%
1,000,000	97.2%

Now let us consider the issue of caching postings lists in more detail. A typical index is usually implemented using an inverted file. Such an inverted file will list all of the term occurrences in a postings list. This postings list includes not only the documents containing a term, but also intra-document location and attribute information as well. The overall distribution of postings list sizes has been shown to be Zipfian [3]. This tells us that there are a few postings that account for most of the index size. In [3], 95% of the inverted file size was accounted for by less than 5% of the postings. But what does the distribution of postings list sizes associated with the most frequently occurring query terms look like? After all, those most frequently occurring terms account for the bulk of the queries. We studied this question by gathering statistics on the query terms and estimated postings list sizes. We compiled a list of the occurrence frequencies of the 10,000 most frequently asked query terms (ignoring capitalization and query operators). For each term, we performed a query against the search engine AllTheWeb in July 2000 (which at the time indexed about 340 million unique pages) and extracted the total number of documents found. The number of documents found is approximately an indicator of the size of the postings list for a file.

A scatter plot of the term occurrences versus document occurrences is shown in Figure 3. There is an overall trend for terms that occur frequently in queries to also be terms that occur frequently in documents. The bulk of the terms are associated with postings lists containing 10,000 to 1,000,000 documents. Because of the large postings sizes, caching of only hundreds to thousands of the top query terms would be feasible. However, because of the Zipfian distribution of the term frequencies, even small caches may lead to significant improvements in performance as shown in Table 3. Returning to the question of pruning the index size, we observe that because there is a direct correspondence between query term occurrences and document occurrences, it would be reasonable to drop terms from the index that occur very infrequently in documents. Although some queries would remain unsatisfied, in a resource-limited environment such a tradeoff may be acceptable. In fact, this is already what some search engines already do [2].

We also plotted the distributions of the query term occurrences versus its rank in Figure 4. On the same graph, we plotted the distribution of the number of documents containing a term versus the document rank. In this context, the highest rank (=1) refers to the term or document that occurs in the most frequently. The term distribution curve is Zipfian. The document distribution curve is for the most part Zipfian, except there is a steep drop off on the right hand side. This is due to the fact that there are a some frequently occurring query terms that have very few matches. These terms may be the result of new topics which suddenly appear (such as due to a major news event) but which there are few documents in the index because the search engine is out of date.

Top X Terms	% of Total	% of Distinct
10	20.5	.00032
100	35.9	.0032
1,000	58.1	.032
10,000	83.5	.32
100,000	95.1	3.2
1,000,000	98.7	32.0

Table 3: Percentage of terms accounted for by the most frequent query terms.

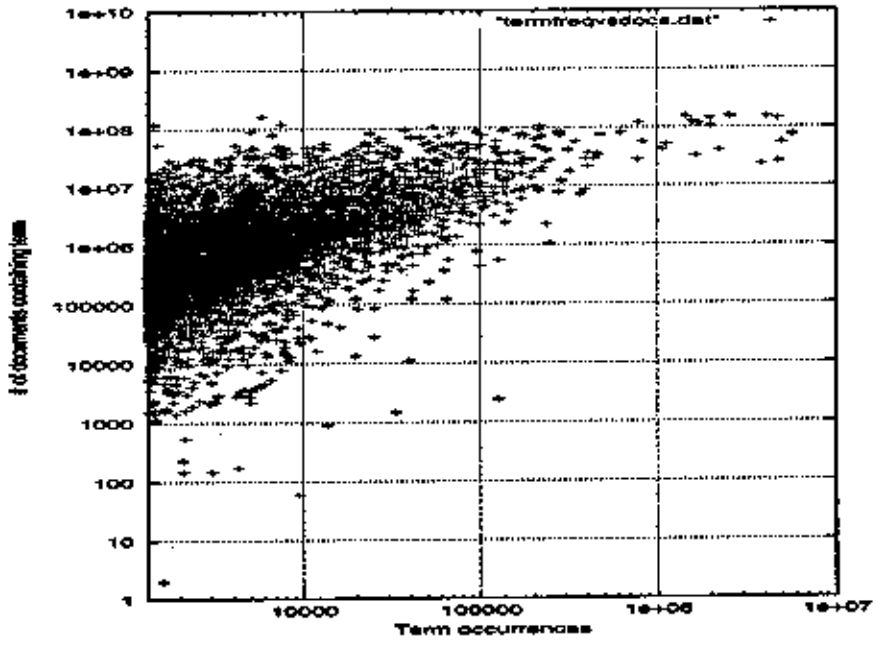

Figure 3: Term and Document Distributions

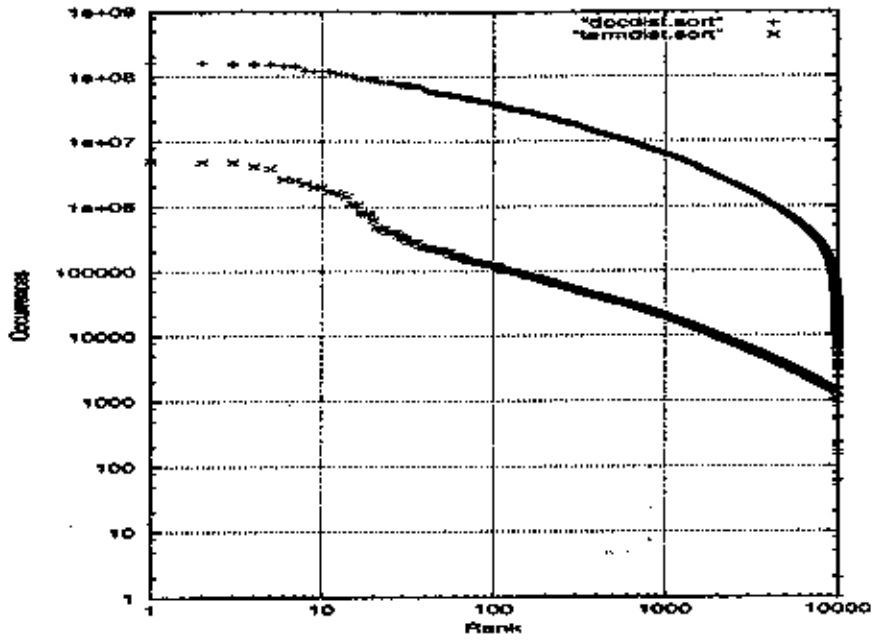

Figure 4: Terms and Documents. Bottom curve is the query term distribution, top curve is the document occurrence distribution.

4 Classifying Terms

In the previous section, we found that the distribution of term occurrences was Zipfian. We implicitly classified terms into two categories: those that occurred frequently (the ones that were ranked in the top X) during the entire query time period and those that did not (unpopular). However, looking at the query term distribution with a finer time granularity reveals an interesting new result. We believe that there are two distinct subclasses of terms with high frequency. First, there are the "popular" terms, that consistently occur with high frequency. Second, there are "hot" terms, which occur with high frequency for a short time period. To demonstrate that distinction, we need to do a time analysis of term occurrences over a long period of time. We break down our entire time period into smaller time intervals and identify the frequently occurring query terms in each time interval. Then we produce an overall view which shows a graph of the number of frequently occurring terms that occurred in 1 through T time intervals. Those terms that occur frequently in only a few intervals are hot terms because they are very actively used for only a short period of time, while those that occur in many intervals are popular because they consistently occur frequently throughout many time intervals.

We may describe the analysis mathematically as follows. Define:

$i = 1, ..., I$	The set of terms
$t = 1, ..., T$	The time intervals
f	The minimum occurrences for high frequency terms
a_{it}	The number of times term i is queried in period t

Further, let

$$b_{it} = \begin{cases} i & \text{if } a_{it} > f \\ 0 & \text{otherwise} \end{cases}$$

and

$$c_i = \sum_{t=1}^{T} b_{it} \quad \forall i$$

Then c_i is the number of periods term i occurred frequently, and we may write

$$s_x = \sum_{\{i|c_i=x\}} 1 \quad for\ x = 1, ..., T$$

as the number of terms which were popular for exactly x time intervals.

4.1 Experimental Results

For the purposes of this paper, we chose to break our 64 day time period into four hour intervals, so $T = 384$. We chose $f = 101$ as the minimum cut-off frequency of the terms. A term was counted for an interval if it appeared at least 101 times during that interval. Figure 5 shows the distribution of frequently occurring terms. The y axis plots s_x, which indicates the total number of distinct terms which were popular for x time intervals. The curve is a bathtub curve. The lower end of the x axis shows that there are a significant number of terms that were popular for brief durations of time while the upper portion of the x axis shows that there were a significant number of terms that occurred frequently during almost every time interval. In the context of our classification this means that you will find the hot terms on the left side of the graph and and the popular terms on the right side of the graph.

4.2 The 80/20 Rule

From Figure 5 we find that approximately 80% of the frequently occurring terms occur in the 20% of the range (the lower and upper 10%). The lower 10% of the curve, those occurring frequently in up to 38 time intervals, made up 67% of the number of frequently occurring terms. The upper 10% of the curve, those terms that occurred frequently for more than 346 time periods made up 14% of the frequently occurring terms.

A similar result is evident when choosing $f = 10$, as shown in Figure 6. The lower 10% of this curve accounts for 70.7% of the total frequently occurring terms while the upper 10% of the curve accounts for 10.9% of the frequently occurring terms.

4.3 Caching Hot and Popular Terms

The evidence of two distinct subclasses of frequently occurring terms has implications on the design of an effective caching system. Extracting a static list of the most popular terms from a query log is insufficient. A good caching algorithm must not only capture the popular terms, but also must dynamically react quickly and capture terms that suddenly become popular and also react quickly enough to throw out terms that just as quickly lose their popularity. An LRU cache should do a reasonably good job of caching, though other new algorithms might be devised to take advantage of the combination of frequency and recency of query terms.

5 Conclusions and Future Work

We have provided a detailed analysis of the characteristics of web queries. Through our studies of vocabulary growth, we have shown that caching of exact user queries provides some potential for improving search engine performance while caching of individual query terms provides considerable potential for improving the query evaluation phase of a search engine. The reasons for these conclusions are that the number of unique queries (58% of all queries) grows much faster than the number of unique query terms (1.8% of all query terms),

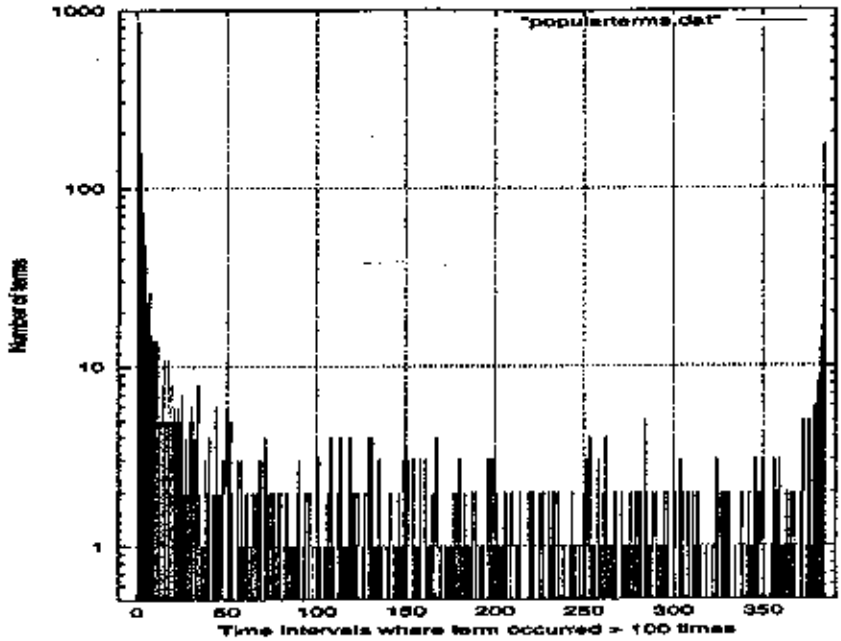

Figure 5: Hot terms are on the left and Popular Terms are on the right

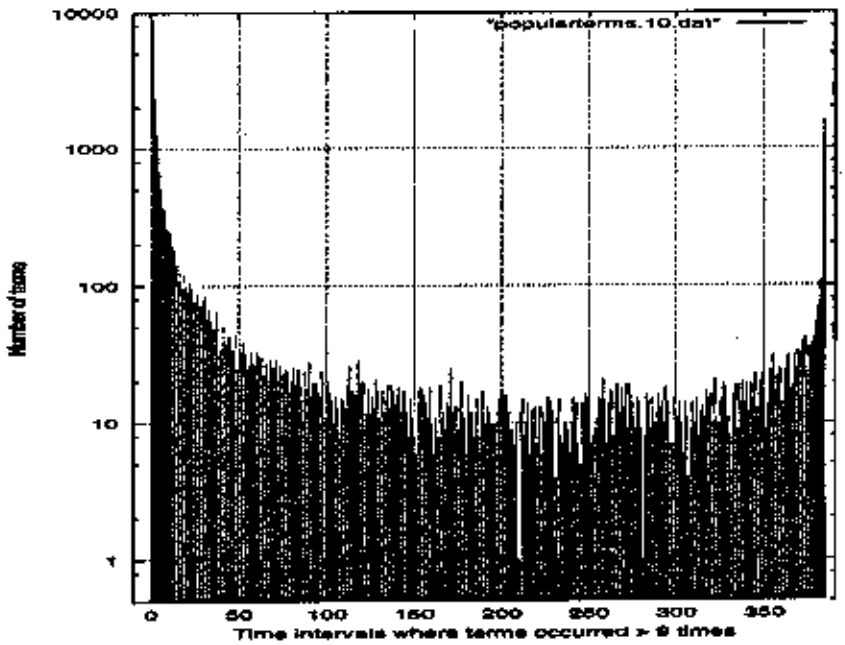

Figure 6: Hot terms are on the left and Popular Terms are on the right

and that very few (3.2%) query terms make up over 95% of all the terms in the queries. This makes a simple LRU cache look very promising. We note that operating systems may already perform LRU caching of disk blocks, which should provide a reasonable, automatic caching of postings, though there may be better specialized algorithms. A better caching strategy should incorporate additional parameters, such as query term frequencies. This parameter looks very promising because of its Zipfian distribution. Thus, even small caches can lead to significant improvements in performance.

To better understand web queries and their temporal behavior, we studied the query occurrence frequencies. Queries were classified as unpopular if they never exceeded a threshold occurrence frequency for any time period. Queries that did exceed the frequency threshold were found to naturally fall into one of two categories: hot and popular. The top 10% hot and top 10% popular terms together made up approximately 80% of the total frequently occurring queries.

There are a few more aspects of Web queries that we would like to study in the future. We have looked at the number of occurrences of the most frequently used terms and found a rough correspondence to the the number of documents containing the term (an approximation of the size of the postings for that term). In the future, we would like to perform a more precise analysis of the relation between query term frequencies and postings list sizes in the search index. This information will allow search engine designers to make intelligent tradeoffs in the pruning of index and dictionary entries.

A logical conclusion to the present investigation is the study of predictive and dynamic algorithms for caching. In this approach, based on the history of queries and term frequencies, a predictive caching algorithm would calculate the "optimum" amount of each postings list to keep in cache to minimize response time. Pitkow and Recker [10] describe a dynamic caching algorithm for Web servers. Their algorithm is based on psychological research of human memory and tries to estimate the probability of future document access based on prior document access. Such an algorithm may be very suitable for predicting query term occurrences based on term frequencies, as well as term recency and previous user queries. We are actively pursuing research in this area.

References

[1] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval.* Addison Wesley, New York, NY, 1999.

[2] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Seventh International World Wide Web Conference (WWW7)*, April 1998.

[3] Eric William Brown. Execution performance issues in full-text information retrieval. Technical report, University of Massacusetts, Amherst, MA, February 1996. Ph.D. Thesis.

[4] CommerceNet. Internet population. *http://www.commerce.net/research/stats/wwwpop.html*, Spring 1999.

[5] Steve Kirsch. Searching the internet, sigir '98 keynote. *http://www.skirsch.com/presentations/sigir.ppt*, August 1998.

[6] Zhihong Lu. Scalable distributed architectures for information retrieval. Technical report, University of Massacusetts, Amherst, MA, May 1999. Ph.D. Thesis.

[7] Evangelos Markatos. On caching search engine query results. In *Proceedings of the 5th International Web Caching and Content Delivery Workshop*, May 2000.

[8] Nielsen//Netratings. Nielsen//netratings expands global internet index to 17 countries. *http://63.140.238.20/press_releases/pr_001031_global.htm*, *http://www.nielsennetratings.com*, October 2000.

[9] Greg R. Notess. Search engine statistics: Database total size estimates. *http://www.searchengineshowdown.com/stats/sizeest.shtml*, October 2000.

[10] James E. Pitkow and Margaret M. Recker. A simple yet robust caching algorithm based on dynamic access patterns. In *Proceedings of the 2nd World Wide Web Conference, Chicago, USA*, October 1994.

[11] Craig Silverstein, Monika Henzinger, Hannes Marais, and Michael Moricz. Analysis of a very large altavista query log. Technical Report 1998-014, Digital, Systems Research Center, Palo Alto, CA, October 1998.

[12] Amanda Spink, Judy Bateman, and Major Bernard J. Jansen. Searching heterogeneous collections on the web: Behavior of excite users. *Information Research*, 4(2), October 1998.