Analysing Users WWW Search Behaviour

Clare Bradford and Ian Marshall, BT Advanced Communications Research.

In a recent study [1], Internet users ranked search as their most important activity, awarding it a 9.1 on a 10-point scale. The next most important activity ranked only 6.3. Internet search engines are continually updating their indexes, and scaling up their parallel processors to keep up with the growth of the WWW. It is estimated that there are 800 million indexable pages in the WWW [2], and the number is growing at a rate of a million pages per day [3]. Existing search engines cover around 15% of these pages, with the six largest public search engines collectively covering only 60%. The coverage is decreasing rapidly. User experience also shows that the frequency of out of date and broken links is increasing. In addition the traffic generated by indexing spiders and metasearch agents is adding significantly to congestion and delay. Perhaps most importantly the users valuation of search engines is declining, since the engines supply too much information.

Previous work [4] has indicated that a cache hierarchy could provide an improved data-set for WWW search engines. A cache hierarchy would cover around 50%[5] of the publicly indexable pages, without generating spider traffic, and avoiding the need for the multiple requests generated by metasearch tools. This proposal therefore solves some of the operational problems, but does not address the users need for more effective filtering. Currently a typical simple query will generate in excess of 105 responses, and users can take many attempts to make their query more specific before successfully reducing the response to a more manageable level. Experience at AltaVista [6] has shown that few users request pages after the first results are listed. They either refine their search or leave the search site. We are attempting to address this problem on two fronts. Firstly we have suggested [7] that cataloguing the contents of the cache will enable users to choose the directory most closely matching their topics before firing their search. The categories are chosen by monitoring the users activity, and adapting to improve the response. An alternative version of the same type of approach [8] allows users to create their own catalogues and federate them. Secondly since the search engines are hindered by their inability to scale [9], we are attempting to reduce the scale by removing duplication.

Unfortunately very little systematic work has been published on the real behaviour of web searches and current attempts to optimise search engine responses are somewhat ad hoc. In order to more fully understand how best to optimise the search engine response we are analysing logs and gathering statistics from a small search engine to confirm how users are using the engine, and particularly how they are refining their queries. If we can capture the most successful user behaviours, then we can write rules to help automate query refinement. Some search engines [10] can already suggest refined queries when a huge number of responses are returned. We believe this is the right way to approach helping users to search more efficiently. However, the rules determined from our log analysis approach will complement the ad hoc, sometimes weak, mathematical formulas already used. We intend to apply the results in an active (user programmable) front end to our cache based distributed search engine.

To tackle the problem of duplicated results we propose the use of URNs by the search engine. Indexers currently list files from different sources by URL, and yet the files are often direct copies of information already listed. If the search engine was made URN-aware, results would only include single copies of documents or images. Duplicate filenames from different sites will have been recognised as being the same information, and thus would only take-up one position in the crucial first-10 search results. We are currently analysing search engine logs to identify how large the benefit of duplicate removal would be.

Our paper shows the results of log analysis and search duplication tests. We hope to suggest rules and patterns to provide search engine developers with information to empower them to help users refine queries more accurately. Only by capturing and analysing what the users actually do, can we predict what needs to be done to improve the query suggestion mechanism. In future we will be testing our proposals on external sites, and encouraging their adoption by major search engines.

Tests Performed

To determine the extent of duplication in search results, broad subject terms were entered into a meta search tool and then into two well known search engines. Specific search terms where then entered. By broad terms, we mean “animals”, “history” and so forth. A specific term could be “Lambourghini Diablo”, for instance. During this investigation it became clear that few of these 'duplicates' are direct duplicates, they were files with similar titles or URL's, as the search engines generally didn't return the same URL twice. However, despite the fact that these may not be identical 'duplicates', many of them could be considered to be duplicates by users (such as pages that the search engine need not have returned as they are closely linked with other pages within the results set). Often, they were part of the same site or the same document spread over more than one page. This is what we mean by duplicate - results which are effectively covered in other pages, which are considered to be repeated information by the users.

The behaviour of users was analysed using search engine logs. 100Mbytes of information were analysed from an intranet search engine [11]. The initial findings will be confirmed on external logs before major conclusions are drawn. However, the collection of intranet users, being a succinct group of highly educated engineers, provides a valuable insight into user behaviour. We can assume that the ‘general public’ will show similar (or exaggerated) traits.

Search terms can appear to be popular, whereas further analysis can show that it’s the same individual making them so. We have analysed search terms in association with the number of users requesting the data. Search session information is key to finding user behaviour. Hence we noted how many search sessions involved single entries, how many searches required two entries, three and so forth. Query length distribution was logged. Refinement strategies of the multiple query sessions were then charted for further analysis and discussion.

Results

The first set of results, with broad search terms, showed that 14% of returned results were duplicates. The second set, with more specific search terms, showed that nearly a third of all the returned results were duplicates. A third set of results, taken from another search engine, gave similar results. Specific queries produce a greater number of duplicates to be returned, and the number of duplicates is around one third of results.

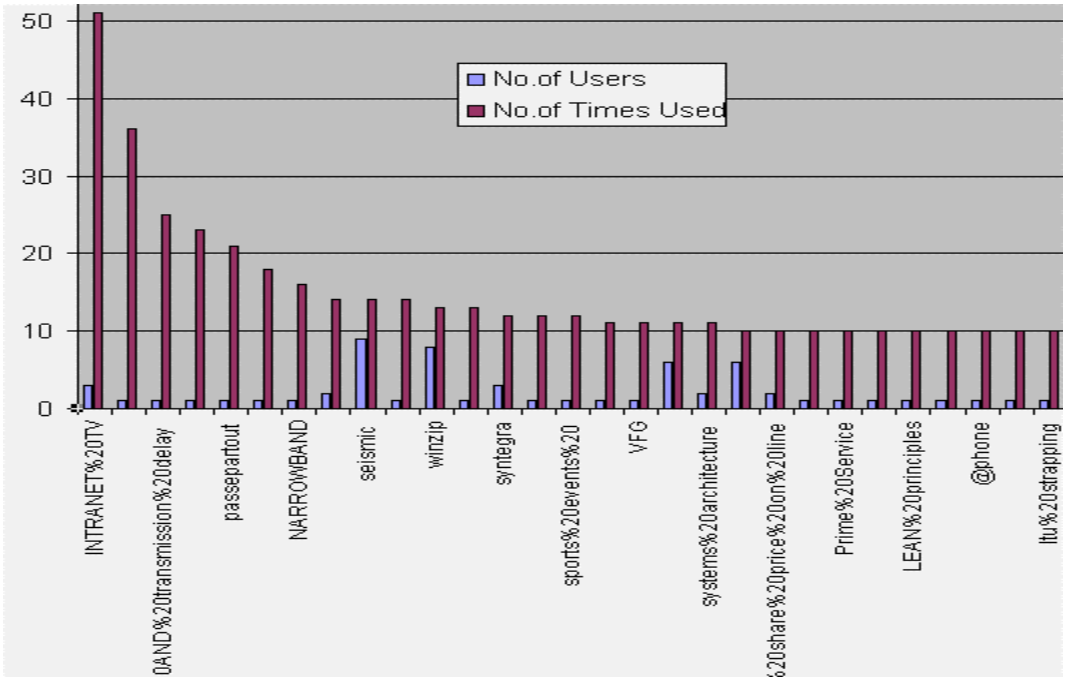

Figure 1 – Common Query Terms

As we can see from Figure 1, most of our queries (46%) were single queries. Whilst they are important for telling us about subjects of interest, such queries do not give much information about query refinement. Figure 2 shows the query length distribution. This refers to the number of queries entered from the client to the server per session and not the quantity of search words comprising a query. A session was defined as an uninterrupted sequence of related queries made by a single client. The sessions were manually checked, and sequences where the query terms appeared unrelated were split into multiple sessions.

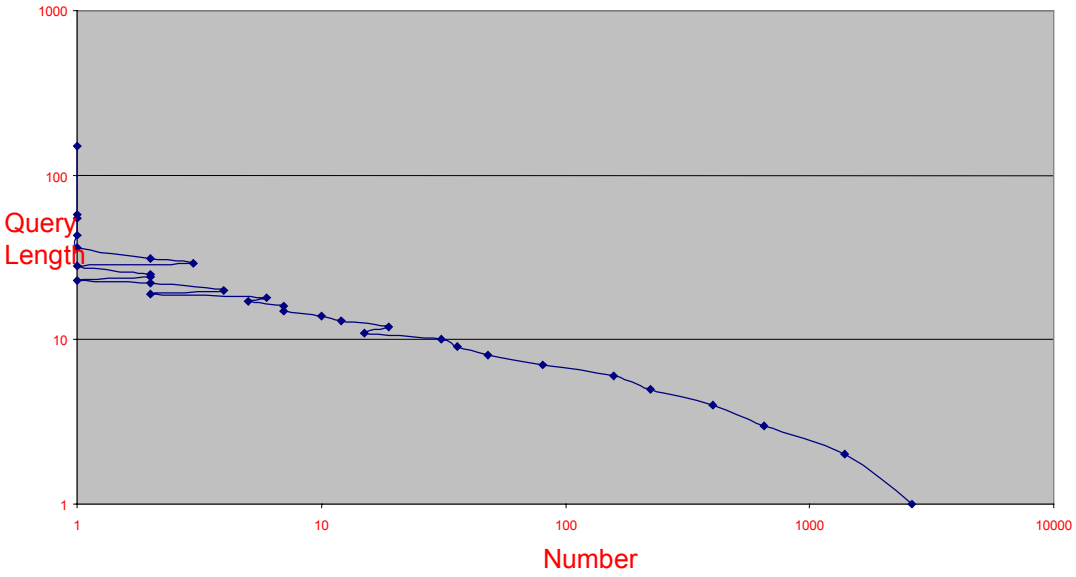

Figure 2 – Query Length Distribution

To map user behaviour, it's important to note what actions are taken between successive search entries. Figure 3 shows the refinement strategies found.

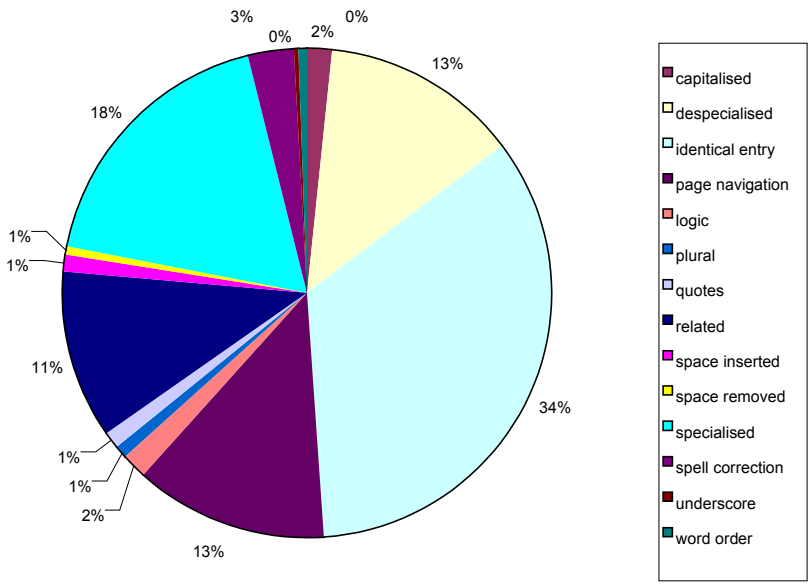

Figure 3 – Refinement Strategies

Some sequences could be categorised in greater depth. For example if the first entry was 'Excel' and the second 'Microsoft Excel', the category could be 'specialised/word-added'. However, the current

categorisation illustrates some interesting trends. The results show a large percentage of identical search terms (47%). Further analysis indicated that just under a third of these are generated by users viewing the next page of results. This leaves a considerable amount of users (34%) who refine their search terms by entering identical search terms into the search engines. Equally surprising is the number of users who are entering the same query (by adding/deleting spaces, changing order of terms outside quotes, etc). Less than 10% of the users were using logic and complex specialisations. Clearly, even in a technically literate community many users do not understand how to get the best from a search engine. It is therefore unlikely that intelligent agents (used for filtering or categorisation etc.) will learn much from observing user behaviour as captured by current search engine logs. A mechanism that encourages users to specify what they want in more detail would deliver more useful information and enable improved optimisation. Possibilities include a more natural speech interface, a dialog based interface, and encouraging users to supply their own filters and categories. We are currently concentrating on the third possibility.

Conclusions

Duplication is a problem for users of search engines and more effective information filtering is required if their valuation of search engines is to improve. Working to remove duplication would allow more relevant search results to be returned, perhaps reducing the quantity of unsuccessful searches. Users interests differ widely from one search to the next and their behaviour does too, so the use of learning agents based on past behaviour analysis does not seem the best way forward. In fact user behaviour can often be highly deterministic, thus any predictive adaptations, based on a probabilistic model, will not always work well. Mechanisms that respond to a more detailed input, obtained through some low effort means such as a speech interface are more likely to succeed. User emphasis is often effort-oriented as opposed to accuracy oriented. Usually simple, easily typed search terms are entered, without a great deal of thought by the user. If 'right' results are not returned then another term may be entered (which might be equally unsuitable or inaccurate). Alternatively, the user frequently gives up after the first poor response. Clearly any improvements must be easy to use and must provide good motivation for ongoing involvement by the user. Perhaps an immediate improvement would be for an intelligent thesaurus to be built into search tools, with spell/grammar correctors and hints about related terms that could be supplied before submitting a query and waiting for a response. It is clear from the data that 50% of the traffic is generated by the longest 10% of the queries. These queries are generated by a small, but highly committed, proportion of the user community. These users will benefit considerably from being able to propose categories that reflect their interests and make their searches faster and easier. The engine operator would also benefit from reduced traffic, and less committed users could also benefit by opting to use some of the proposed categories too.

References

[1] Jupiter Research, 1999 (via Infoseek Press Release)

[2] Steve Lawrence and C.Lee Giles "Accessibility of Information on the Web" Nature/Vol400/8 July 1999/www.nature.com

[3] Scientific American, June 1999 "Hypersearching the Web", IBM's Clever Project team

[4] I. Marshall, Kiam, Bradford "Building a scalable distributed multimedia directory". Proceedings of ACM '98, Bristol, short paper poster presentation

[5] Ramon Caceres, Fred Douglis, et al "Web Proxy Caching:The Devil is in the Details". Proceedings of 1998 Internet Server Performance Workshop June 23 1998, pp 111.

[6] ftp://ftp.digital.com/pub/DEC/SRC/technical-notes/SRC-1998-014.pdf Analysis of a Very Large AltaVista Query Log

[7] Clare Bradford and Ian.W.Marshall, IEEE Multimedia Systems '99 International conference on multimedia computing and systems, June 7-11, 1999 "A Bandwidth Friendly Search Engine", Vol 2 pp720

[8] Mikhail Bessonov, Heuser et al "Open Architecture for Distributed Search Systems" 6th International Conference on Intelligence and Services in Networks, IS&N'99 April 1999, pp55.

[9] http://www.forbes.com/forbes/98/1130/6212336a.htm "Inktomi Inside"

[10] http://searchenginewatch.internet.com/subscribers/articles/9901-altavista.html, "AltaVista Adds Related Search Prompter"

[11] BT Index Server – unofficial search engine of BT Intranet