中央研究院
資訊科學研究所
Institute of Information Science, Academia Sinica • Taipei, Taiwan, ROC

TR-IIS-05-013

Active Feedback for Effective Web Search

Ray-I Chang, Jan-Ming Ho

September 2005 || Technical Report No. TR-IIS-05-013

http://www.iis.sinica.edu.tw/LIB/TechReport/tr2005/tr05.html

Active Feedback for Effective Web Search

Ray-I Chang

Department of Engineering Science

National Taiwan University

Taipei, Taiwan, ROC.

Jan-Ming Ho

Institute of Information Science

Academia Sinica

Taipei, Taiwan, ROC.

Abstract

Current search engines with arrays of servers can provide efficient web content services. However, as their returned results are usually enormous mass, finding target information is still time-consuming. To increase the correctness of returned results, different page ranking methods were introduced. Some of them also try to use users' feedback to increase their precision in ranking. However, as the traditional approaches are passive in feedback and greedy in search, experiments show that the average error ratio is over 20%. Their returned results are usually too large to satisfy users' needs. In this paper, an active feedback technology is introduced. It bases on the concept of balanced tree to present some critical questions for guiding users to have the proper feedback in further searching. The same idea can be applied to assist either distributed or P2P (peer-to-peer) search engines to balance workloads and speed responses.

Keywords— active feedback, balanced tree, distributed, peer-to-peer, search engine

1. Introduction

The number of web sites and pages are highly increasing. The *Web Server Survey* published by Netcraft on June 2003 found over 40 millions of Web sites on the Internet [1]. The number of Web pages is zillion. It becomes more and more difficult to retrieve target content from such an information ocean. Users really need a good scheme to reduce the response time in searching target content [2]. Nowadays, the search industry has evolved two dominant ways to retrieve information: human-powered *directories* and crawler-based *search engines*. These two types of methods gather their presentation lists of Web pages in radically different ways [4].

A *directory* categorizes knowledge into some structures and classifies individual Web pages with respect to the pre-designed structure. The most prominent directory in commercial is Yahoo. An example of the directory of Web sites shown in www.yahoo.com is as follows.

As there are more and more pages on the Web, the classification of web pages becomes a labor-intensive activity (there are much more "publishers" on the Web than "classifiers"). By the way, if the pre-designed directory does not reflect the information you seek, you are out of luck [3]. An alternate of the human-powered directory is the crawler-based search engine. Google (www.google.com) is the most famous crawler-based search engine in worldwide. It will *crawl* (or *spider*) different Web sites to create a database of data pages automatically. Then, people search through what the crawlers have found by presenting a query statement (usually, a single keyword) [4]. Currently, Google also provides the service to do the keyword search for other portal sites (including Yahoo). An example of the crawler-based search service shown in www.yahoo.com is as follows.

Based on the query statement and a pre-defined measurement function (usually the query statement's importance in a Web page), the search engine returns a set of relevant pages as the result. Users tend to get valuable results at the first couple of retrieved pages [3, 6]. However, as there are so many pages on the Web and few of them are valuable, it always takes a user lots of time in finding his real interest. A good search engine needs to show the most relevance results on the top one or two pages.

To further focus on the real interest of a user, Current researchers apply the measurement of a relevance score to rank pages. The PageRank technology [8] of Google and the PolyRank technology [9] of Openfind are known as two of the most important page ranking methods. To further rank pages by the relation of Web sites, the SSP (Subject-Specific Popularity) technology

[10] of Teoma and the Prisma technology [11] of AltaVista are introduced. They analyze the relationship of Web sites within a community based on the number of same-subject pages referred. For example, the number of citations to a page can be an evidence of its importance. We can display the target results by listing pages from high to low citation. Although there are a few differences between these technologies, the main concern is the degree of relationship and significance of Web pages. Its correctness is highly dependent on the accuracy of user input and page relation. Unfortunately, both of them are non-guaranteed.

Another possible way of target emphasizing is to ask users to give a more explicit query statement. Like the Advanced Web Search in Yahoo shown as follows. It is usually a boolean function of the page's keywords, updated time and site domain.

However, a user may have the difficulty in explicitly specifying his interest. It is not easy to specify an explicit query statement, not only the keywords but also their boolean function, for searching the target Web pages. Moreover, a user has no idea about the characteristics of Web

pages stored in the database. A query statement may be explicit but redundant in information and inefficient for searching.

How to provide an efficient method to find target information is an important research issue. Some researches try to use users' feedback to increase the precision of page ranking. Before doing the feedback, the search results are first ranked by following their significance to the query statement. Every time the user is clicking to see a Web page. This page provides new information to adjust the scores of significance for ranking pages. Therefore, a new result is presented to try to get approaching user's target. In Yahoo, it also provides the category of each page and three related high score categories to users.

In Google, the most related keyword in the database is also provided to users to adjust their query statements. Notably, all these current approaches assume that the feedback page is an analogue of the target. However, as users can't explicitly specify their targets, this assumption is not always correct. Even a user is so lucky and it is available to find the related pages. A greedy scheme that takes only the related pages into the consideration of feedback usually leads to a large overlap in the search result. The search space is still very large. Besides, as the feedback is passive, the system never tries to understand what kind of impacts the feedback will introduce. Therefore, the item clicked for feedback may not benefit the system's performance. If a user searches the web pages only from the passive feedback [13,14], it usually returns too many results. Experiments show that the retrieval error ratio is over 20% [12]. It needs a more effective assistant scheme for

speeding search.

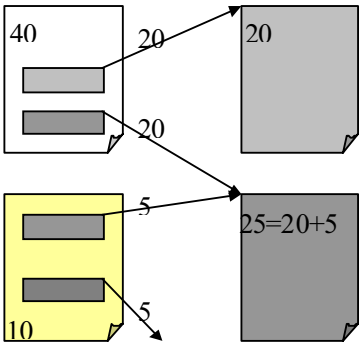
Let's move the scenario from the Web content search to the daily life search. Think about "who will be the best assistance in searching" and "what will he do." Imagining a scenario of the library reference situation, you may request directions to get a collection by your own query statement. The librarian, being experienced with such situations, is not going to serve you by following your statement passively. Instead, based on a measurement of the collections and your request, he usually asks you some questions actively to get a better understand of your target and to narrow down your search space. The degrees of significance and relationship (they have been applied in many feedback methods) are two good parameters in this measurement. However, the amount of return results should be also considered to maximize the system performance (on the worst or the average cases, not only the best case).

Not like a librarian, presented search engines don't have the ability to ask good questions to focus their search. They concern only the similarity of web pages. However, pages with high degrees of significance and relationship may not focus users' requests. In this paper, we base on the heuristic of library reference to propose an active feedback technology for Web content search. By analyzing the significance and relationship of result pages, we can estimate the effects for different keywords in feedback. Adopting the concept of balanced tree, a set of keywords can be identified to guide users to have a better search results. Our goal is not to replace search engine, but to provide a new assistant method. The same idea can be extended to assist either distributed or P2P (peer-to-peer) search engines to try to balance workloads and speed responses. It makes the original search engine more efficient.

2. Related Works

2.1 Page Ranking

Earlier page ranking methods are based on the boolean and vector models. They use the term-weighting (the frequency of terms within the pages) and the similarity function (the similarity between each page and a query) to rank pages [15, 16]. The most well known method is the PageRank algorithm [17]. It is proposed by Google to provide a more sophisticated scheme for citation counting. Usually, the number of citations to a page (the link structure of the page) is evidence of its importance. A page has a high rank if the sum of its back links' ranks is high.



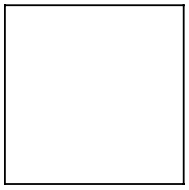
This method covers both the case when a page has many back links and when a page has a few highly ranked back links [8]. However, it leads some shortcomings, *e.g.*, new page not good as old page, small page less than large page, and professional page less than general page. To solve these shortcomings, Teoma [18] uses pages' communities to determine which pages are most relevant. It ranks a page based on the number of same-subject pages that reference it, not just general popularity [10]. The similar idea is applied in AltaVista [20] and WiseNut [19].

The accuracy of pages' ranking depends on the correctness on user's query and his feedback. Usually, the user's input is unalterable to be extremely short. It perhaps consisting of one or two terms (1.3 terms on the average) [6,7,21], and usually a short phrase or even a single word. In

many cases, even users have input several terms for searching, they still get bad results. Reasons for such bad results come from the passive human-interface in searching.

2.2 Greedy and Passive Feedback

A passive system never tells users what would be an explicit feedback for the next step in searching. It only provides the most similar results in ranking and waits users to give a feedback. While a user selects one of the result pages as the feedback, a greedy scheme is applied for ranking. It simply assumes that the feedback is explicit and the scores of the related pages are adjusted by following the feedback. Actually, users don't know what information in the selected page will be returned to the search engine. While a user's feedback is beyond the assumption (it always happens), the returned results of such a greedy search scheme would be inaccurate and irrelevant. By the way, even the feedback is explicit, strong relationship usually lead to large overlap in search results.



2.3 Search Expertise

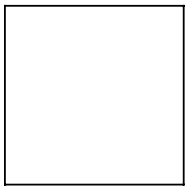
A librarian is known as an expertise to get requested collections from users' query statements. These statements are usually inexplicit as we presented during the Web search. Instead of passively waiting, a librarian will actively ask the user some questions to get a better understand of the target. Based on a measurement of the collections and the request, he tries to estimate the effect of feedback for different keywords related to the request. Using a balanced tree of Web pages in the database, a set of keywords can be identified to guide the user to fast narrow down the search space. Notably, the degrees of significance and relationship that have been applied in

current search engines are just two of parameters considered by a librarian.

3. Web Content Search with Active Feedback

Conventional approaches use the degrees of similarity and popularity for page ranking. It follows a greedy rule in searching. However, the greedy scheme is work only when the quality of user's query and its related feedback are explicit. It is not always work. In this paper, we introduce a novel method that tries to improve the performance of Web search by active feedback. No doubt, the user is the one and the only one that understand most what he want. But, actually holding the web pages and knowing how to deal with them is the search engine. If the system doesn't provide accurate information in searching, it is flatly impossible to expect users to find their requested pages. For example, the amount of return results should also be considered for minimizing the average search time.

An active feedback approach analyzes the distribution of Web pages and provide users the suggestion for proper feedback. Users are guided for fast searching, and don't need to think or type too much. Besides, as the system has tried to understand what kind of impacts the feedback will introduce, the item clicked for feedback would highly benefit the system's performance. It provides an auxiliary method on search engines nowadays to focus user demand efficiently and quickly. A simple example is shown as follows.



Different from the greedy rules what existing search engines are doing, we can consider the "weighted average height" of a balanced tree regarding keywords' significance and relationship.

The results is a factor to influence the system performance. Let Q_0 be the original query term from a user. Keywords list is a pre-built table that records the significant keywords and their significance and relationship. The MetaSearch system is an intermediary search mechanism. It could combine query terms with select keywords from user, and send them to search engines for searching. Finally, we show the results and the suggestive keywords for users. The function of each component is described in more detail below.

First, we built up the keywords list in advance that provides a series keywords for effective searching. It records important keywords, and corresponds to the significance and relationship of each keyword. The keywords influence the ranking; deduce the effect of the next status of system, and finally choose the suggestive keywords for effective searching. The principal keywords are all pre-built by the ordinary search engines. This function records the search results. It can help determine the influence that the effect of next status of system, depends on the results and each of keywords. To measure the degree of influence, we use the standard IR metrics, precision and recall. On the other hand, to evaluate the potency whether the keywords could reduce the amount of irrelevance information. In order to compute the degree of influence exactly, we must know the significance of each keyword, the relationship among keywords, and the connection between keywords and results. These tasks can be done by search engines.

Consider the search term Q_j. It is composed of original search term Q_0 and a series of keywords from user feedback.

$$Q_j = Q_{j-1} \bullet K_j = Q_0 \bullet K_1 \bullet K_2 \bullet \ldots \bullet K_j \quad (1)$$

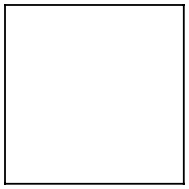
Where ● is the composite operator that indicates the boolean operator of information retrieval, could be + (AND) or – (OR). We consider the useful keywords else, it does not care that given Q_0, K_1, K_2, ... , K_j, to determine the proper keywords. There are a number of applications that use the tree structure to manage their data. The most popular application is the fast search [22]. The efficiency depends upon the balance of the tree. We use the balanced tree for information retrieval.

The balance means not only the efficient balance but also structural balance..

Let m_j is the number of results for search terms Q_j . The records of one page is m , the total pages N is then represented by m_j/m, and N_i is the number of pages related with keyword K_i (related records/m). Let $P_i=P(K_i|Q_j)$ denotes the probability that results include the keyword K_i on existing search term Q_j. R_i represent the significance and relationship that new keyword K_i regard the original search term Q_j. The degree of influence of average efficiency of search engine W_{ji} is given by (2).

$$W_{ji} = P_i*log(N_i*R_i) + (1-P_i)*log((N-N_i)*(1-R_i)) \quad (2)$$

This value, W_{ji} , represents the average height of balanced tree. Smaller value means the keyword is more efficiently. Applying the decision tree for balanced effectively, users could differentiate between what they want and don't want. In addition, it can focus on the users demand fast.



We now choose several keywords, the number of keywords depend up the setting of system. Consider the design form of web page, we get the eight keywords, and then integrate the keywords into result pages. In order to search the information quickly, the system prompts suggestive keywords, include both positive lists (AND) and negative lists (NOT), for users for advanced filtering capabilities.

4. Results

In this section, we evaluate our mechanism for balanced tree to determine what improvement can be

achieved. Due to we have not created the search engine by ourselves. Instead, we have developed a MetaSearch system to retrieve data from Google search engine. The search terms of metasearch system use boolean query model (using AND or NOT) that composing original search term (Q_0) that user input and the keywords list ($K_1, K_2, ...K_n$) that we established in advance. Then, adopting the equation (2) mentioned above to compute their retrieval efficiency for different keywords. Finally, we choose the keywords that make the balanced tree becoming balance, and show the keywords to users for advanced select.

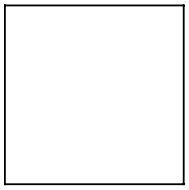
Suppose the original query term (Q_0) is "computer" (in Chinese). The numbers of results are 852,000 records. Now we combine Q_0 with each of keyword in the keywords list, to implement the "AND" operator (the numbers of results for "NOT" operator are the subtraction of total records and the records of "AND" operator). The partial results are listed in Table 1. (The implementation time of example was at 7:10~11:00 on May. 30, 2003)

The keyword list built up by manual in advance. For the generation, we set the significance and relationship of every keyword is all the same. Furthermore, the connections between keywords are same. On the other hand, users don't special like for any keywords. Therefore, we could ignore those factors while computing W_{ji}. The results are then ranked according to the value, from small to large. It means they could reduce the average height of the tree, attaining the balance of tree. It's the best way, on the other hand, to eliminate amount of data from irrelevance data.

Keywords	Records	Degree of influence
online	385000	5.63142
software	371000	5.63304
problem	482000	5.63317
management	367000	5.63359

safe	363000	5.63418
technology	490000	5.63433
center	495000	5.63513
network	501000	5.63618
Chinese	339000	5.63853
global	325000	5.64173
technique	322000	5.64248
forum	314000	5.64460
news	552000	5.64869
…	…	…
852000/2 = **426000**		

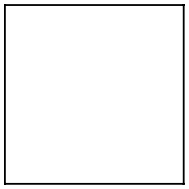
We could choose the top *M* keywords to become the suggestive terms, the value *M* depend on demand or layout of system. Consider the layout of web site, we choose eight keywords, and then combine the keywords with result pages. The keywords show on page top. Besides the keywords, we also place two boolean operator, “AND” and “NOT”. User could select any one to refine the results.

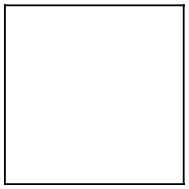


Give a description of working processes. First, user input the query term, Q_0 = “computer”. The metasearch system send the query term to Google and got the results, show original ranked results from Google and suggestive keywords include “online”, “software”, …, etc. When user chooses a

suggestive term "online" (K_x), new query term (Q_1) is "computer online" (Q_0+K_x). Repeat the work above. Showing the new ranked results and new suggestive terms, include "news" "network"…etc. It is obvious that the suggestive keywords differ from the first stage.

The best way, using the balanced tree and active feedback, user selects a keyword that provided by system every time, it could efficient reduces a half of data. The second stage of searching could reduce another half of data. Hence, this method provides efficient assist for search engine. To integrate with similarity of keyword of search engine, it could get more efficiency. In the worst case, in another word, users don't choose any term; the search effect is the same as the original search engine.





5. CONCLUSION

In this paper, we presented an efficient method that used balanced tree and active feedback to assist in Internet search. By suggesting a list of more benefit keywords actively, user could find the data required quickly and the originally vague and broad concept becomes clear and focus gradually. It is clear that our method is superiority in theory, even though the system is insufficient for scope and preciseness. There are still many issues regarding efficient search that

deserve further study. First, improving the keywords list further is necessary. Second, we have developed the one tier balanced tree. Next, we will extend the algorithm to multi-tier balanced tree. The other concepts such as the game tree (versus the balanced tree), the possibility (versus the probability) and the miss rate (versus the hit rate) in searching should also be considered. Furthermore, the user interface, for suggestive keywords and ranked results, will show and advance in accord with feedback. Finally, the active feedback technology can apply to provide personalization in P2P environment. And for loading balance in a distributed environment the concept of balanced tree is useful.

REFERENCES

[1] NetCraft, "May 2003 Web Server Survey," available at: http://news.netcraft.com

[2] H. Liu, H. Lieberman and T. Selker, "GOOSE: A Goal-Oriented Search Engine With Commonsense," *Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web Based Systems. (AH2002)*, Malaga, Spain, 2002

[3] R. E. Filman and S. Pant, "Searching The Internet," *IEEE Internet Computing*, Vol. 2(4), 1998

[4] D. Sullivan, "How Search Engines Work," SearchEngineWatch.com, Oct. 2002, available at: http://searchenginewatch.com/webmasters/
article.php/2168031

[5] D. Sullivan, "How Search Engines Rank Web Pages," SearchEngineWatch.com, Oct. 2002, Available at: http://searchenginewatch.com/
webmasters/article.php/2167961

[6] C. Silverstein, M. Henzinger, H. Marais and M. Moricz, "Analysis of a very large altavista query log," *Technical Report SRC 1998-014*, Digital Systems Research Center, 1998. See also SIGIR Forum 33(1), pp. 6-12.

[7] B. J. Jansen and U. Pooch. "A review of web searching studies and a framework for future

research," *Journal of the American Society of Information Science and Technology*, Vol. 53(3), 2000, pp. 235-246

[8] L. Page, S. Brin, R. Motwani and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," *Stanford Digital Library working paper SIDL-WP-1999-0120*, Available at: http://abpubs.standord.edu:8090/pub/1999-66, 1999

[9] Metamend web site, "The Search Engine Openfind," Available at: www.metamend.com/ openfind.html, 2003

[10] Teoma Search Engine, "Adding a New Dimension to Search: The Teoma Difference is Authority," Available at: http://sp.teoma.com/ docs/teoma/about/searchwithauthority.html, 2003

[11] http://www.altavista.com/help/search/pp

[12] C. J. Wu, "Distributed Metadata System and Retrieval Error Ratio," *Pacific Neighborhood Consortium Annual conference*, 1999

[13] C. J. Wu, "A Primary Investigation of Subject Frequency in Library Literature and Liisa," *Bulletin Of NCL Taiwan Branch*, Vol. 8(3), 2002 (in Chinese)

[14] C. J. Wu, "Effectiveness analysis of Subject Subdivision: Using NBINet," *Bulletin Of Library And Information Science*, Vol. 43 , 2002 (in Chinese)

[15] B. Y. Ricardo and R. N. Berthier, *Modern Information Retrieval*, Addison-Wesley, 1999

[16] B. Y. Ricardo and B. F. William, *Information Retrieval Data Structures and Algorithms*, Prentice Hall, 1992

[17] http://www.google.com

[18] http://www.teoma.com

[19] http://www.wisenut.com

[20] http://www.altavista.com

[21] C. L. Clarke, G. V. Cormack and E. A. Tudhope, "Relevance ranking for one to three term

queries," *Information Processing and Management*, Vol 36(2), 2000, pp. 291-311

[22] W. Mark, Data Structures and *Algorithm Analysis in C*, Addison-Wesley, 1999