Excalibur : A Personalized Meta Search Engine

Leo Yuen, Matthew Chang, Ying Kit Lai and Chung Keung Poon
Department of Computer Science, City University of Hong Kong
{leo, kcmchang, yklai, ckpoon}@cs.cityu.edu.hk

Abstract

General purpose Web search engines are becoming ineffective due to the rapid growth and changes in the contents of the World Wide Web. Meta-search engines help a bit by having a better coverage of the WWW. However, users are still overwhelmed by the large amount of irrelevant results returned by a search. A promising approach to tackle the problem is personalized search. Thus the problem of capturing users' personal information need and re-organizing the results has attracted a lot of attention. In this paper, we present a meta-search engine that extracts users' preference implicitly and provides immediate response by re-ranking the results. Re-ranking is done by using the Naive Bayesian classifier and the resemblance measure. Moreover, we show that the users' preference can be succinctly represented by a few keywords.

1 Introduction

The World Wide Web contains virtually all kinds of information needed by people from all walks of life. General purpose search engines [1] used to be an effective tool for retrieving information from this huge information repository. However, as the web technology is becoming popular, the information that is available on the web has been growing at a rate that is difficult for these search engines to keep up. The size and the growth of the web is simply too huge and fast for a search engine to keep its index up-to-date. Meta-search engines help the situation a bit by having a better coverage of the WWW. However, the size of search results returned by a (meta-)search engine per query is overwhelming.

A promising approach to combat this problem is personalization. By taking a user's personal interest into account, one can focus the crawling of pages on some specific topics [3, 4, 5, 8]. This not only reduces the data set size but also improves the precision, as the crawled pages are related to the target topics. Personalization also helps in filtering the result. As 85% of the queries made by users contained less than three terms [7], the query terms alone are usually not specific enough to accurately identify the required pages. Further filtering and ranking of the high recall but low precision results returned by a search engine helps improving the quality of the search results. Thus, how to capture the users' preferences and how to present the results according to the preferences have attracted a lot of attention recently.

Simply adding more keywords and re-doing a query is not very useful. Composing a query with the relevant keywords based on a user's information need is already challenging. This is especially true when the user does not know much about the topic (s)he searches. There are search engines that categorize the search results and allow users to select the categories they want. However, the categorization may not fit the users' need exactly. Thus (s)he may need to search for multiple categories, making the searching process cumbersome. Some search engines allow users to see pages similar to a page for each page in the search results. However, it is easy to get lost after a few such steps since the similarity of pages is not a transitive property. Thus, we may end up with pages which are totally irrelevant to what we wanted at the very beginning.

2 Highlights of Our Idea

In this paper, we introduce a method to extract user's personal interest implicitly based on the user's browsing actions and to provide immediate response to the user by re-ranking the search results. Such re-ranking greatly improves the result presentation and makes the web surfing more convenient. Moreover, we employ a server-based approach as opposed to the client-based approach used in many previous work on this topic. We use a server-side meta-search engine and put the user's profile in the client machine as a cookie [1]. The filtering process is done on the server-side. This is possible because our method is able to represent the user's preference succinctly using a few keywords.

[1] http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2109.html

3 Detailed Design

We built a prototype system named $Excalibur$ to demonstrate the feasibility of our idea. In addition to the text box for typing querying keywords, the user can also select a *Search Session* from a list. Each *Search Session* is identified by a group of query terms made by the user previously. Since a user may have different preferences for different information needs, we provide a list of groups of keywords for the user to identify the preference (s)he wanted.

The query is then issued to different general purpose search engines. Search results from each of them are then aggregated into one result list using Borda rule. [2] Taking the top 100 results from each search engine should be sufficient as over half of the users did not access results beyond the first page (≤ 20 results in most search engines). [7]

Before presenting them to the user, each page is fetched from the Internet. This will weed out dead links and make sure that the contents of the pages are current. Moreover, we will check the contents of the pages and rank them using either Naive Bayesian classifier or resemblance [2]. Due to space limitation, we will only describe the ranking based on Naive Bayesian.

Initially, all the pages are classified as "bad". We first consider the case in which the user has only specified a number of keywords for querying but not a *Search Session*. In this case, we wait for the user to click on a page. We assume this is a page of interest to the user and move it into the class of "good" pages. Then we use the Naive Bayesian classifier to assign a score to each page p in the result set as follows. After counting the frequency of each word appearing in "good" pages and "bad" pages, we know the probability of each term being good (P_{Good}) and bad (P_{Bad}). We then calculate $P_{Good}(p)$, the probability that page p is good, by multiplying the P_{Good}'s of each word in that page. Similarly, we calculate $P_{Bad}(p)$. Finally, page p will receive the score $P_{Good}(p) - P_{Bad}(p)$ and the search results are re-ranked based on this score. Then we wait for the user to click again and the above process is repeated. In this way, user's preference is implicitly detected and appropriate response is given immediately to the user.

All the intermediate statistics will be stored temporarily in the server within a session. Words with the largest difference between P_{Good} and P_{Bad} will be useful in identifying the user's preference in this search session and maybe useful for the user in future searches. Thus, these words together with their probabilities of being in the "good" class will be associated with this *Search Session* (represented by the list of user's query terms) and stored in the user's machine as a cookie. This information will be imported to the Naive Bayesian classifier when this *Search Session* is used again in later queries. Only the top 10 words per *Search Session* are stored to fit in the maximum size (4K Bytes) of a cookie. Experiments show that this is already quite sufficient to identify a user's preference.

$Excalibur$ also provides a function to rank results with respect to a specific result using full text resemblance with shingling. This allows an instant re-ranking of the search result so that pages will be ranked in the order of their similarity to the preferred ones, without the need for training the classifier.

[2] vwww10.hrz.tu-darmstadt.de/vwl5/mitarbeiter/thomas/deborda1784.pdf

4 Conclusion and Future Work

We have built a prototype system that demonstrated the feasibility of capturing user's preference and re-ranking of results by Naive Bayesian classifier and resemblance measure. Rudimentary experiments show that the speed and precision of both methods are quite promising. While Naive Bayesian classifier is regarded as one of the best text classifiers, our results seemed to indicate that resemblance measure is also rather competitive. Both of them are suitable for ranking web pages on the fly. More systematic experiments will be done to compare their performances.

Since our method is able to represent a user's profile succinctly in a few keywords, it opens up the possibility of collaboration of customization information among users [6].

References

[1] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[2] Broder. On the resemblance and containment of documents. In *SEQS: Sequences '91*, 1998.

[3] S. Chakrabarti, M. van den Berg, and B. Dom. Focused crawling: a new approach to topic-specific Web resource discovery. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1623–1640, 1999.

[4] M. Chau, D. Zeng, and H. Chen. Personalized spiders for web search and analysis. In *ACM/IEEE Joint Conference on Digital Libraries*, pages 79–87, 2001.

[5] L. Chen and K. Sycara. WebMate: A personal agent for browsing and searching. In K. P. Sycara and M. Wooldridge, editors, *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*, pages 132–139, New York, 9–13, 1998. ACM Press.

[6] B. Chidlovskii. Collaborative re-ranking of search results.

[7] B. J. Jansen, A. Spink, J. Bateman, and T. Saracevic. Real life information retrieval: a study of user queries on the web. *SIGIR Forum*, 32(1):5–17, 1998.

[8] I. Martin and J. M. Jose. A personalised information retrieval tool. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 423–424. ACM Press, 2003.