Web Page Ranking Based on Events

Ajay Gupta, Manish Bhide, and Mukesh Mohania

IBM India Research Laboratory,
Block-1 IIT Delhi, Hauz Khas, New Delhi-110016
{agupta,abmanish,mkmukesh}@in.ibm.com

Abstract. Search results of web search engines are displayed according to their ranking function, which is a function of the in-links and the out-links of the web page. Users are generally overwhelmed by the thousands of results retrieved by the search engine, few of which are useful. Most of the search engine users are interested in the latest information about the searched keywords. Such pages containing the latest information about an event (or incident) are not always ranked high by the search engines due to the lack of sufficient in-links pointing to these web pages. For example, if a search query “DaWaK” is given to Google, the users are generally interested in the latest information i.e., the home page of DaWaK 2004 conference, which is not ranked in the top 10 results returned by Google. In this paper we address the problem of ranking the web pages based on events (or incidents) related to the searched keywords. We provide a method for finding patterns that constitute events in the search results returned by a conventional search engine and then rank the web pages based on these event patterns. We also describe a mechanism whereby our technique can be used inside the ranking function of a conventional search engine such as Google. We provide experimental results that validate the efficiency and use of our technique in capturing the user intentions.

1 Introduction

Many a times when a user is searching for information about events that are repeatable in nature, the user is interested in information about the latest events. Search engines today, cannot capture these user intentions and the results returned by the search engine are oblivious of this requirement of the user. The main reason for this is that the ranking functions of the search engines depends on the in-links and out-links of the web pages and the web page that has the latest information is generally not pointed to by many other web pages. Further, the ranking function of the search engines does not consider the semantic similarity between web pages and the user query while ranking. Hence there is a need to identify the users’ intentions while ranking the web pages and displaying them to the user. In the paper, we outline a technique that is able to capture such user intentions, when the user is interested in searching information about recurring events. Our technique ranks the web pages based on the events related to the searched keywords. We make use of the search results of a conventional search engine (like Google) and discover patterns that constitute a periodic event in these search result. We then re-rank these results, based on discovered patterns and their relevance to the searched keywords.
1.1 Related Work

There has been a lot of work in improving the web search results. A meta-search agent based methodology has been proposed in [3] that captures the semantics of a user’s search intent, transforms the semantic query into target queries for existing search engines, and ranks resulting page hits according to a user-specified weighted-rating scheme. In [2], the authors have described a meta-search architecture that allows users to provide preferences in the form of an information need category. This extra information is used to direct the search process, providing more valuable results than by considering only the query. The difference in these approaches and the one proposed in this paper is that the former approaches improve the search results based on some extra user-specified information, whereas our approach does not require any extra information from the user. [4,5,6] organize the search results into hierarchical categories whereas [7,8] organize search results into clusters. However all of these approaches do not have a concept of events and recentness of information.

1.2 Contributions

In this paper, we focus on displaying the search results in accordance with the users’ intentions, when the user is interested in information about recurring events. We address the following important problems:
1) How to find the user intentions when the user is searching for information related to a recurring event?
2) How to find the web pages that contain information about recurring events?
3) How to rank the web-pages based on their relevance to the recurring event and based on the semantic similarity of web-pages to the searched keywords?

We have laid out the rest of the paper as follows. In the next section, we give an overview of our technique when it is implemented outside a conventional search engine. Section 3 gives an outline of our algorithm that can be implemented inside a search engine, thereby making the process faster. The performance of our technique is shown in section 4 and we conclude the paper in section 4.

2 Architecture of the System

Our technique can be implemented both inside as well as outside a conventional search engine. This section gives a brief overview of the tool when it is implemented outside a conventional search engine. At a high level, our technique tries to identify web pages that constitute a pattern related to the searched keywords. For example, if the user query is “DaWaK” then there will be a set of pages that will have “DaWaK 2002”, “DaWaK 2003”, “DaWaK 2004” etc. in the title of the web page. We identify such web pages and rank them higher as the user is generally interested in the latest information which will be part of such a pattern. Figure 1 shows the architecture of our system when it is implemented outside a conventional search engine. The details of the architecture are given below:
1) The input to the system is the user query (i.e. keywords). In this technique we re-rank the search results based on the user intentions. To get the relevant web-pages related to the searched keywords, the user query is sent to a conventional search engine like Google. The results (URL’s) returned by the search engine along with their snippet are sent to the Query Characterizer.