Web Search via Learning from Relevance Feedback

Xiannong Meng
Bucknell University, USA

Zhixiang Chen
University of Texas-Pan Americana, USA

INTRODUCTION

Recently, three general approaches have been taken to increase Web search accuracy and performance. One is the development of meta-search engines (e.g., MetaCrawler, *www.metacrawler.com*) that forward user queries to multiple search engines at the same time in order to increase the coverage and hope to include what the user wants in a short list of top-ranked results. Another approach is the development of topic-specific search engines that are specialized in particular topics. These topics range from vacation guides (*www.vocations.com*) to kids' health (*www.kidshealth.com*). The third approach is to use some group or personal profiles to personalize the Web search. Examples of such efforts include *GroupLens* (Konstan et al., 1997).

Those meta-search engines suffer to a certain extent the inherited problem of information overflow that it is difficult for users to pin down specific information for which they are searching. Specialized search engines typically contain much more accurate and narrowly focused information. However, it is not easy for a novice user to know where and which specialized engine to use. Most personalized Web search projects reported so far involve collecting users' behavior at a centralized server or a proxy server. While it is effective for the purpose of e-commerce, where vendors can collectively learn consumer behaviors, this approach does present the privacy problem.

The clustering, user profiling and other advanced techniques used by those search engines and other projects such as Bollacker et al. (1998) are *static* in the sense that they are built before the search begins. They cannot be changed dynamically during the real-time search process. Thus they do not reflect the changing interests of the user at different time, at different location or on different subjects. Intelligent Web search systems that dynamically learn the users' information needs in real-time must be built to advance the state of the art in Web search. Machine learning techniques can be used to improve Web search, because machine learning algorithms are able to adjust the search process dynamically so as to satisfy users' information needs.

BACKGROUND

There have been great research efforts on applications of machine learning to automatic extraction, clustering and classification of information from the Web. Some earlier research includes *WebWatcher* (Armstrong et al., 1995), which interactively helps users locate desired information by employing learned knowledge about which hyperlinks are likely to lead to the target information; *Syskill and Webert* (Pazzani et al., 1996), a system that uses a Bayesian classifier to learn about interesting Web pages for the user; and *NewsWeeder* (Lang, 1995), a news-filtering system that allows the users to rate each news article being read and learns a user profile based on those ratings. Some research is aimed at providing adaptive Web service through learning. For example, *Ahoy! The Homepage Finder* (Shakes et al., 1997) performs dynamic reference shifting; and *Adaptive Web Sites* (Perkowitz & Etzioni, 2000) automatically improves their organization and presentation based on user access data.

A series of work in Chen et al. (1999, 2000, 2001, 2002) and Meng and Chen (2004) study intelligent Web search as an adaptive learning process, where the search engine acts as a learner and the user as a teacher. The user sends a query to the engine, and the engine uses the query to search the index database and returns a list of URLs that are ranked according to a ranking function. Then the user provides the engine relevance feedback, and the engine uses the feedback to improve its next search and returns a refined list of URLs. The learning (or search) process ends when the engine finds the desired documents for the user. Conceptually, a query entered by the user can be understood as the logical expression of the collection of the documents wanted by the user. A list of URLs returned by the engine can be interpreted as an approximation to the collection of the desired documents.

ADAPTIVE LEARNING FOR WEB SEARCH

Let ***X*** denote the set of all index keywords for the whole Web (or, practically, a portion of the whole Web). Given any Web document *d*, let *I(d)* denote the set of all index keywords in ***X*** that are used to index *d* with non-zero values. Then, the following two properties hold. (1) The size of *I(d)* is substantially smaller than the size of ***X***. Practically, *I(d)* can be bounded by a constant. The rationale behind this is that in the simplest case only a few of the keywords in *d* are needed to index it. (2) For any search process related to the search query *q*, let *D(q)* denote the collection of all the documents that match *q*; then the set of index keywords relevant to *q*, denoted by *F(q)*, is $F(q) = \cup_{d \in D(q)} I(d)$. Although the size of *F(q)* varies from different queries, it is still substantially smaller than the size of *X*, and might be bounded by a few hundred or a few thousand in practice.

Definition 1 - *Given any search query q, F(q), which is given in the previous paragraph, is defined as the set of dynamic features relevant to the search query q.*

Definition 2 - *Given any search query q, the dynamic vector space V(q) relevant to q is defined as the vector space that is constructed with all the documents in D(q) such that each of those documents is indexed by the dynamic features in F(q).*

Adaptive learning for intelligent Web search as studied in Chen et al. (1999, 2000, 2001, 2002) and Meng and Chen (2004) is formulated as follows. Let *S* be a Web search system. For any query *q*, *S* first finds the set of documents *D(q)* that match the query *q*. It finds *D(q)* with the help of a general-purpose search strategy through searching its internal database, or through external search engine such as AltaVista (*www.altavista.com*) when no matches are found within its internal database. It then finds the set of dynamic features *F(q)*, and later constructs the dynamic vector space *V(q)*. Once *D(q)*, *F(q)* and *V(q)* have been found, *S* starts its adaptive learning process with the help of the learning algorithm that is to be presented in the following subsections. More precisely, let $F(q) = \{K_{1,...,} K_n\}$ such that each K_i denotes a dynamic feature (i.e., an index keyword). *S* maintains a common weight vector $w = (w_{1,...,} w_n)$ for dynamic features in *F(q)*. The components of *w* have non-negative real values. The learning algorithm uses *w* to extract and learn the most relevant features and to classify documents in *D(q)* as relevant or irrelevant.

Practically efficient adaptive learning algorithms such as TW2 have been developed in (Chen et al. (1999, 2000,

Figure 1. Algorithm TW2

Step 0: Set $w_0 = (w_{01},..., w_{0n}) = (0,...,0)$.

Step $i > 0$: Classify documents in $D(q)$ with $\sum_{j=1}^{n} w_{ij}x_j > \theta$.

While (user judged a document $x = (x_1,..., x_n)$) do
  If (x is relevant) do //promotion
    For ($j = 1; j \le n; j++$)
      If ($x_j = 0$) set $w_{i+1,\,j} = w_{ij}$
      If ($x_j \neq 0$ & $w_{ij} = 0$) set $w_{i+1,\,j} = \alpha$
      If ($x_j \neq 0$ & $w_{ij} \neq 0$) set $w_{i+1,\,j} = \alpha w_{ij}$
  If (x is irrelevant) do //demotion
    For ($j = 1; j \le n; j++$)
      Set $w_{i+1,\,j} = w_{ij} / \alpha$
If (user has not judged any document) stop
Else set $i = i + 1$ and go to step $i + 1$.