Web Mining in Thematic Search Engines

Massimiliano Caramia
Istituto per le Applicazioni del Calcolo IAC-CNR, Italy

Giovanni Felici
Istituto di Analisi dei Sistemi ed Informatica (IASI-CNR), Italy

INTRODUCTION

The recent improvements of search engine technologies have made available to Internet users an enormous amount of knowledge that can be accessed in many different ways. The most popular search engines now provide search facilities for databases containing billions of Web pages, where queries are executed instantly. The focus is switching from *quantity* (maintaining and indexing large databases of Web pages and quickly selecting pages matching some criterion) to *quality* (identifying pages with a high quality for the user). Such a trend is motivated by the natural evolution of Internet users who are now more selective in their choice of the search tool and may be willing to pay the price of providing extra feedback to the system and to wait more time for their queries to be better matched. In this framework, several have considered the use of data-mining and optimization techniques, which are often referred to as *Web mining* (for a recent bibliography on this topic, see, e.g., Getoor, Senator, Domingos & Faloutsos, 2003), and Zaïane, Srivastava, Spiliopoulou, & Masand, 2002). Here, we describe a method for improving standard search results in a thematic search engine, where the documents and the pages made available are restricted to a finite number of topics, and the users are considered to belong to a finite number of user profiles. The method uses clustering techniques to identify, in the set of pages resulting from a simple query, subsets that are homogeneous with respect to a vectorization based on *context* or *profile;* then we construct a number of small and potentially good subsets of pages, extracting from each cluster the pages with higher scores. Operating on these subsets with a genetic algorithm, we identify the subset with a good overall score and a high internal dissimilarity. This provides the user with a few nonduplicated pages that represent more correctly the structure of the initial set of pages. Because pages are seen by the algorithms as vectors of fixed dimension, the role of the context- or profile-based vectorization is central and specific to the thematic approach of this method.

BACKGROUND

Let P be a set of Web pages, with $p \in P$ indicating a page in that set. Now assume that P is the result of a standard query to a database of pages, and thus represents a set of pages that satisfy some conditions expressed by the user. Each page $p \in P$ is associated with a score based on the query that generated P, which would determine the order that the pages are presented to the person submitting the query. The role of this ordering is crucial for the quality of the search: In fact, if the dimension of P is relevant, the probability that the user considers a page p strongly decreases as the position of p in the order increases. This may lead to two major drawbacks: The pages in the first positions may be very similar (or even equal) to each other; pages that do not have a very high score but are representative of some aspect of set P may appear in a very low position in the ordering, with a negligible chance of being seen by the user.

Our method tries to overcome both drawbacks, focusing on the selection from the initial set P of a small set of pages with a high score and sufficiently different from each other. A condition needed to apply our approach is the availability of additional information from the user, who indicates a search context (a general topic to which the search is referred to, not necessarily linked with the search keywords that generated the set P), and a user profile (a subjective identification of the user, which may either be provided directly by choosing amongst a set of predefined profiles or extracted from the pages that have been visited more recently by that user).

MAIN THRUST

The basic idea of the method is to use the information conveyed by the search context or the user profile to analyze the structure of P and determine in it an optimal small subset that better represents all the information available. This is done in three steps. First, the search context and the user profile are used to extract a finite set

of significant words or page characteristics that is then used to create, from all pages in *P,* a vector of characteristics (page vectorization). Such vectorization represents a particular way of looking at the page, specific to each context/profile, and constitutes the ground on which the following steps are based.

Second, the vectorized pages are analyzed by a *clustering algorithm* that partitions them into subsets of similar pages. This induces a two-dimensional ordering on the pages, as each page *p* can now be ordered according to the original score within its cluster. At this point the objective is to provide the user with a reduced list that takes into account the structure identified by the clusters and the original score function.

This is done in the third step, where a *genetic algorithm* works on the pages that have a higher score in each cluster to produce a subset of those pages that are sufficiently heterogeneous and of good values for the original score. In the following sections, we describe the three steps in detail.

Page Vectorization

The first step of the method is the representation of each page that has been acquired by a vector of finite dimension *m*, where each component represents a measure of some characteristic of the page (page vectorization). Clearly, such representation is crucial for the success of the method; all the information of a page that is not maintained in this step will be lost for further treatment. For this reason we must stress the thematic nature of the vectorization process, where only the information that appears to be relevant for a context or a profile is effectively kept for future use. In the most plain setting, each component of the vector is the number of occurrences of a particular word; you may also consider other measurable characteristics that are not specifically linked with the words that are contained in the page, such as the presence of pictures, tables, banners, and so on. As mentioned previously, the vectorization is based on one context, or one profile, chosen by the user. You may then assume that for each of the contexts/profiles that have been implemented in the search engine, a list of words that are relevant to that context/profile is available, and a related vectorization of the page is stored. Many refinements to this simple approach, may and should be considered. The dimension *m* of the vector (i.e., the number of relevant words associated with a context) is not theoretically limited to be particularly small, but you must keep in mind that in order to apply this method over a significant number of pages, it is reasonable to consider $m \leq 100$. We propose two methods to determine such a list of words:

- The words are determined in a setup phase, when the search engine managers decide which contexts/profiles are supported and what words are representative of that context/profile. This operation may be accomplished together with the users of a thematic engine devoted to a specific environment (such as an association of companies, a large corporation, or a community of users)
- The words are identified starting from an initial set of pages that are used as training sample for a context/profile. When user profiles are used, you may consider as a training sample for a profile the pages that have been visited more recently by the user(s) that belong to that profile so that the words associated with the profile evolve with the behavior of the users in a smooth way.

Page Clustering

Extensive research has been done on how to improve retrieval results by employing clustering techniques. In several studies the strategy was to build a clustering of the entire document collection and then match the query to the cluster centroids (see, e.g., Willet, 1988). More recently, clustering has been used for helping the user in browsing a collection of documents and in organizing the results returned by a search engine (Leuski, 2001; Zamir, Etzioni, Madani, & Karp, 1997) or by a metasearch engine (Zamir & Etzioni, 1999) in response to a user query. In Koller and Sahami (1997) document clustering has also been used to automatically generate hierarchical clusters of documents.

Document clustering in information retrieval usually deals with agglomerative hierarchical clustering algorithms (see, e.g., Jain, Murty & Flynn, 1999) or *k*-means algorithm (see Dubes & Jain, 1998). Although agglomerative hierarchical clustering algorithms are very slow when applied to large document databases (Zamir & Etzioni, 1998) (single link and group average methods take $O(|P|^2)$ time, complete link methods take $O(|P|^3)$ time), *k*-means is much faster (its execution time is $O(k \cdot |P|)$). Measuring clustering effectiveness and comparing performance of different algorithms is a complex task, and there are no completely satisfactory methods for comparing the quality of the results of a clustering algorithm. A largely used measure of clustering quality that behaves satisfactorily is the Calinski-Harabasz (*C-H*) *pseudo-F* statistic; the higher the index value, the better the cluster quality. For a given clustering, the mathematical expression of the *pseudo-F* statistic is $C\text{-}H = \frac{R^2}{(k-1)} / \frac{(1-R^2)}{(n-k)}$, where $R^2=(SST\text{-}SSE) / SST$, *SST* is the sum of the squared distances of each object