An Adaptive Agent for Web Exploration Based on Concept Hierarchies

Scott Parent, Bamshad Mobasher*, Steve Lytinen

School of Computer Science, Telecommunication and Information Systems
DePaul University
243 S. Wabash Ave., Chicago, Illinois, 60604, USA

ABSTRACT

In this paper we present the design of a client-side agent, named ARCH, for assisting users in one of the most difficult information retrieval tasks, i.e., that of formulating an effective search query. In contrast to traditional methods based on relevance feedback, ARCH assists users in query modification *prior* to the search task. The initial user query is (semi-)automatically modified based on the user's interaction with an embedded, but modular, concept hierarchy. This allows for the generation of richer queries than is possible, for example, based on simple query expansion according to lexical variants such as synonyms. In addition, ARCH passively learns a user profile by observing the user's past browsing behavior. The profiles are used to provide additional context to the user's information need represented by the initial query.

1. INTRODUCTION

The World Wide Web is a vast resource of information and services that continues to grow rapidly. The heterogeneity and the lack of structure that permeates much of the ever expanding information sources on the World Wide Web, such as hypertext documents, makes automated discovery, organization, and management of Web-based information difficult. Traditional search and indexing tools of the Internet and the World Wide Web such as Lycos, Alta Vista, Google, and others provide some comfort to users, but they do not generally provide structural information nor categorize, filter, or interpret documents. Often users' poorly designed keyword queries return inconsistent search results, with document referrals that meet the search criteria but are of no interest to the user. Studies have shown that users consistently find the tasks of formulating the right query and managing or filtering the results quite difficult.

In recent years these factors have prompted researchers to design more intelligent tools for information retrieval, such as intelligent Web agents (Boley, et. al. 1999; Bollacker, et. al. 1998; Craven, et. al. 2000; Joachims, et. al. 1997; Lieberman 1997) and mechanisms for incrementally refining users queries (Allan 1996; Eguchi 2000). Traditional approaches to *query reformulation* (also called *query expansion*), generally perform one or both of two tasks: re-weighting the terms in the query and expanding the query using additional terms. These tasks are usually performed using relevance feedback from the search results (Buckley, et. al. 1994) or by expanding the query with lexical variants of keywords such as synonyms (Miller, 1997). Relevance feedback mechanisms allow users to refine search parameters based on explicit judgments on the relevance or non-relevance of results from the initial search. However, these approaches do not allow for the creation of an effective query *prior* to an initial search, and furthermore, they are not adaptive in that they assume user's information needs remain constant at different stages of the search process (Mizzaro, 1997).

In this paper, we describe a client-side agent which is designed to assist users in searching large collections of documents, such as the World Wide Web. The agent utilizes a hierarchically-organized semantic knowledge base in aggregate form, as well as an automatically learned user profile, to enhance user queries. In contrast to traditional relevance feedback mechanisms, our approach allows for semi-automatic reformulation of the user's initial query prior to the search phase. Specifically, our intent is to assist the user in generating richer queries by (a) asking the user to identify portions of the hierarchy which are relevant and irrelevant to the query; (b) using pre-computed term vectors associated with each node in the hierarchy to enhance the original query; and (c) identifying any relevant portions of the user profile to provide additional context for the user's information need. Since there are many hierarchical knowledge bases from which semantically related concepts can be drawn, the design of the system is intentionally modular and not specific to a particular knowledge base. This allows the user to switch among the representations of different domain-specific hierarchies depending on the goals of the search.

* Please direct correspondence to Bamshad Mobasher (*mobasher@cs.depaul.edu*).

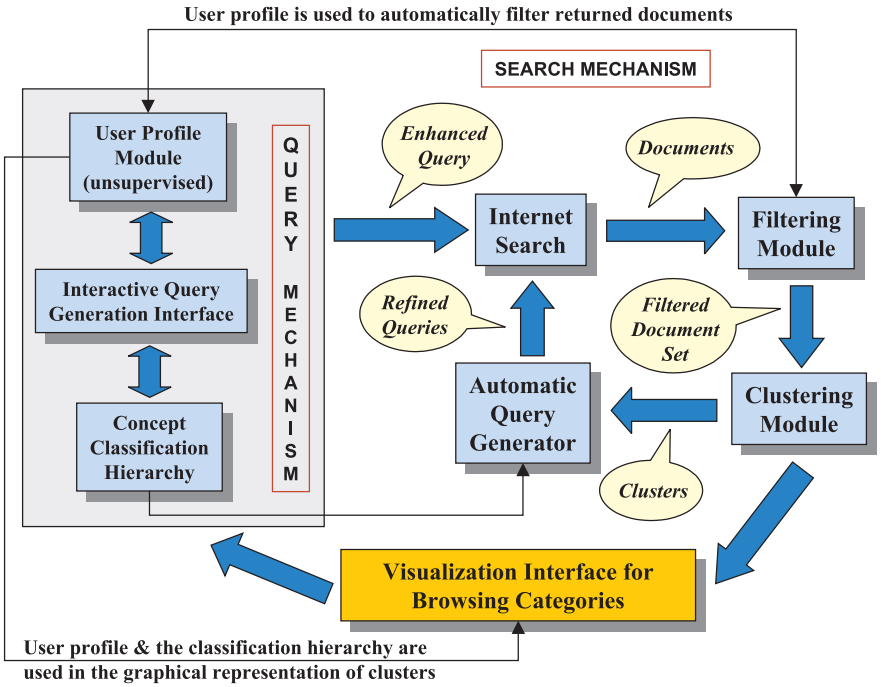

Figure 1. The architecture for ARCH - An Adaptive agent for Retrieval based on Concept Hierarchies

For example, a user interested in performing a general search on the Web can "plug in" the aggregate representation of the Yahoo concept hierarchy. On the other hand, a user interested in searching the Web for specific medical information can, instead, use a medical concept classification hierarchy such as Medical Subject Headings (http://www.nlm.nih.gov/mesh/). It is also possible to use lexical knowlegebases, such as WORDNET (Miller, 1997), in order to capture linguistic relationships among concepts in the search process.

We envision the query enhancement agent to be part of a larger system, as depicted in Figure 1. The system, called ARCH (Adaptive agent for Retrieval based on Concept Hierarchies), combines several capabilities: A highly interactive interface for guiding the user in formulating queries by providing visual tools to graphically view and manipulate the query in the context of the user profile and the concept classification hierarchy; mechanisms for learning a user profile; mechanisms for automatically clustering the search results and automatically generating refined queries according to cluster summaries and the hierarchical knowledge base; and a visual interface for assisting the user in displaying and navigating the results of a query in automatically created semantic categories.

In the rest of this paper we focus on the query generation mechanism for ARCH. We first present our approach for automatically reformulating user's initial query based on the selected portions of the concept hierarchy. We then discuss how the generated query can be further enhanced based on a learned user profile. Throughout the paper, we illustrated the functionality of the query generation mechanism by an example with results based on an experimental data set and the Yahoo concept hierarchy. The clustering and filtering mechanisms of ARCH will be discussed in a future paper.

2. QUERY ENHANCEMENT BASED ON CONCEPT HIERARCHIES

In ARCH, user queries are generated in several stages. In the first stage, the user formulates an initial query, Q_1, which is simply a list of keywords. The system responds to this list by displaying those portions of the hierarchy which are most relevant to the keywords. Once the relevant portions of the hierarchy are displayed, the user is asked to select those categories which are relevant to the intended query, and to "deselect" those categories which are not relevant. The system maintains an aggregate representation of the concept hierarchy. This is to allow for efficient storage and manipulation of the concept hierarchy and to facilitate matching user keywords to nodes in the hierarchy. This aggregate representation is obtained by pre-computing a weighted term vector for each node in the hierarchy which represents the centroid of all documents and subcategories indexed under that node. Specifically, given a node n in the concept hierarchy containing a collection D_n of individual documents and a set S_n of subcategories, the term vector for the node n is computed as follows:

$$T_n = \left(\left(\sum_{d \in D_n} T_d \right) / |D_n| + \sum_{s \in S_n} T_s \right) / (|S_n| + 1),$$

where T_d is the weighted term vector which represents document d (indexed under node n in the hierarchy), and T_s is the term vector which represents subcategory s of node n. Note that the individual documents indexed under n are collectively weighted the same as any of the subcategories under n. Standard information retrieval text preprocessing techniques are performed to initially create a global dictionary of terms from which the features in term vectors are drawn. These techniques include stemming (Porter, 1980) and the removal of stop words. Furthermore, for computing term weights extracted from text we use a standard function of the term frequency and inverse document frequency (tf.idf) as commonly used in information retrieval (Salton & McGill, 1983; Frakes & Baeza-Yates, 1992).

As an example, consider the scenario in which the user may start with a single keyword query "music," using the Yahoo hierarchy. The system may then display an appropriate portion of the hierarchy to the user, containing parents, children and siblings of the node corresponding to the initial query. The user can now select (or deselect) various nodes. In this case we assume that the user is interested in finding information about different types of music, thus selecting the node corresponding to "Genres" in the hierarchy. A portion of the Yahoo hierarchy corresponding to this scenario, as well as the term vector for this node, are depicted in Figure 2. Low scoring terms (stems) appearing in the term vector are not shown in the figure. The term vector which represents the node "Genres" in the Yahoo hierarchy is computed from a combination of those documents indexed under "Genres," as well as the term vectors representing its subcategories (e.g., "Jazz," "Blues," and "New Age").

Once the system has matched the term vectors representing each node in the hierarchy with the list of keywords typed by the user, those nodes which exceed a similarity threshold are displayed to the user, along with other adjacent nodes. An ambiguous keyword might cause the system to display several different portions of the hierarchy. The user is asked to select those categories which are relevant to the intended query, and to deselect those categories which are not relevant. Our approach to the generation of an enhanced query based on nodes in the concept hierarchy is an adaptation of Rocchio's method for relevance feedback (Rocchio, 1971). Using the selected and deselected nodes, the system produces a refined query Q_2, as follows:

$$Q_2 = \alpha \cdot Q_1 + \beta \cdot \sum T_{sel} - \gamma \cdot \sum T_{desel},$$

where each T_{sel} is a term vector for one of the nodes selected by the user, and T_{desel} is a term vector for one of the deselected nodes. The factors α, β, and γ are tuning parameters representing the relative weights associated with each component with the condition that $\alpha + \beta - \gamma = 1$.

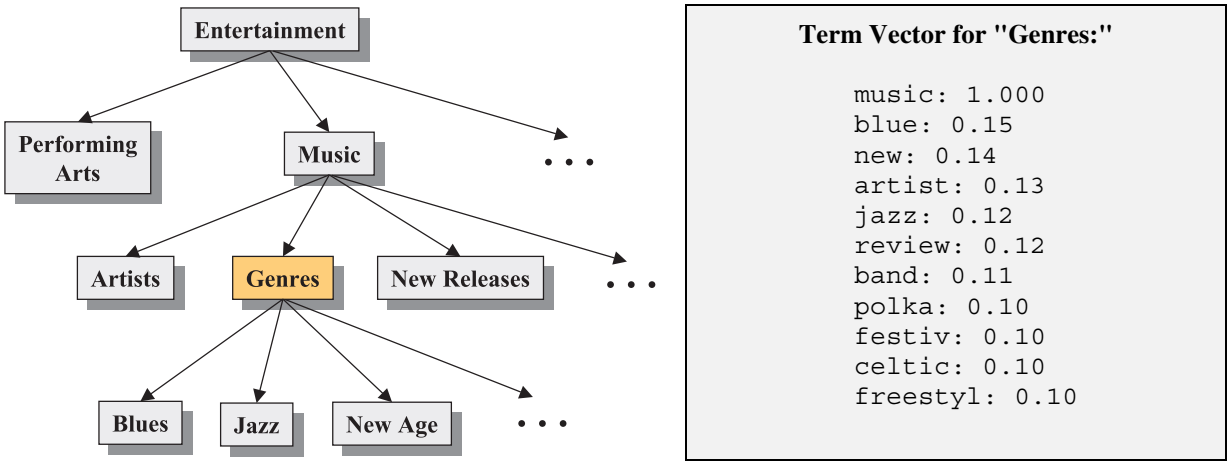

Figure 2. Portion of the Yahoo hierarchy corresponding to the query "music" and the selection of the node "Genres" is depicted on the left. The partial term vector for "Genres" which provides and aggregate representation of the node is given on the right.

Continuing with our example, suppose that the user now selects "Jazz" and deselects "Blues." Furthermore, suppose the user also selects "Dixieland Jazz" as a subcategory of "Jazz." This will result in a modified query, more accurately reflecting the user's information need. In particular, the top scoring terms (and their weights) based on the above scenario will be as follows:

music:1.00, jazz:0.44, dixieland:0.20, tradition:0.11, band:0.10, inform:0.10

3. DERIVATION OF USER PROFILES FROM DOCUMENT CLUSTERS

The user profile generation component of ARCH employs several heuristics to automatically (and without user intervention) determine the topics of "interest" to the user. This component works by passively observing the user's browsing behavior over time and collecting and analyzing documents in which the user has shown interest. The heuristics used by the system are based on several factors, including the frequency of visits to a page or a site, the amount of time spent on the page (or related pages within a site), and other actions of the user such as bookmarking a page. Once enough documents have been collected as part of the profile generation, the system clusters the documents into semantically related categories.

As in the case of nodes within the concept hierarchy, each document in the profile is represented as a term vector, with term weights derived using the tf.idf measure. Clustering algorithms, such as k-means, partition a document set into groups of similar documents based on a measure of vector similarity. A common measure used in information retrieval, also employed in ARCH, computes the similarity of two vectors based on the normalized inner product of the those vectors. Our method for the generation of topical profiles is similar to the "concept indexing" method described in Karypis & Han (2000). Individual profiles, each representing a topic category, are computed based on the centroid of the document clusters. Specifically, given the document collection, D and a document cluster $c \subseteq D$, we construct a profile pr_c as a set of term-weight pairs:

$$pr_c = \{\langle t, weight(t, pr_c)\rangle \mid weight(t, pr_c) \geq \mu\},$$

where the significance weight, $weight(t, pr_c)$, of the term t within the profile pr_c is given by

$$weight(t, pr_c) = \frac{1}{|c|} \cdot \sum_{d \in D} w(t, d)$$

and $w(t, d)$ is the weight of term t in the document vector d. The threshold μ is used to filter out insignificant terms within each profile. Each profile, in turn, can be represented as a vector in the original n-dimensional space of terms, where n is the number of unique terms in the global dictionary.

Once the enhance query, Q_2, is derived from the concept hierarchy, each profile can be compared to the query vector for similarity. Those profiles which satisfy a similarity threshold, are then used to further expand the query, resulting in a new query, Q_3. Note that, as in the case of query expansion based on the concept hierarchy, the new query is computed as a weighted sum of the term vector representing Q_2 and the normalized sum of term vectors representing the matching profiles. To continue our example, let us assume that, through her past browsing behavior, the user has expressed interest in a number of documents related to intellectual property and copyright in music (among other topics). Clustering on our experimental data set based on this scenario, resulted in the following matching profile with respect to the enhanced query described earlier:

music:1.00	intellectu:0.48	inform:0.28	peopl:0.20
jazz:0.76	law:0.47	tap:0.21	band:0.20
copyright:0.69	record:0.35	arrang:0.20	musician:0.20
properti:0.60	protect:0.29	product:0.20	author:0.20

This profile was obtained by computing the centroid vector of one of the document clusters produced by the k-means algorithm (in this case those documents relating to intellectual property in music).

Finally, the enhanced query vector generated by the system based on concept hierarchy and the user profiles (i.e., Q_3) is as follows:

music:1.00, jazz:0.71, band:0.40, copyright:0.22, properti:0.19, intellectu:0.15, law:0.15, artist:0.14, perform:0.14, dixieland:0.13, inform:0.12, protect:0.09, record:0.11, tradition:0.09, musician:0.06

Performing searches based on this and other queries have shown improved retrieval effectiveness. Due to space constraints, these results are not shown here. A detailed evaluation of the system based on standard measures such as precision and recall will be provided in a future paper.

4. CONCLUSIONS AND OUTLOOK

We have presented the work-in-progress on the design of ARCH, an adaptive agent which can assist users in formulating an effective search query based on a modular concept classification hierarchy and a learned user profile. Preliminary experiments have shown that the agent can substantially improve the effectiveness of information retrieval both in the general context of the Web, as well as for search against domain-specific document indexes. The full system also incorporates mechanisms for categorizing and filtering the search results, and using these categories for performing refined searches in the background. Our future work in this area is focused on a detailed evaluation of the system based on standard information retrieval measures such as precision and recall. We also plan on extending the query generation component by incorporating mechanisms for incremental modification of user profiles and incremental updating of the concept hierarchy based on document categories generated during the search process.

REFERENCES

Allan, J. (1996), "Incremental Relevance Feedback for Information Filtering". In Proceedings of the ACM SIGIR '96, pp. 270-278, ACM Press.

Boley, D., Gini, M., Gross, R., Han, E-H., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., and Moore, J. (1999). "Document Categorization and Query Generation on the World Wide Web Using WebACE". Artificial Intelligence Review, Vol. 13, No. 5-6, pp. 365-391.

Bollacker, K., Lawrence, S., Lee Giles, C. (1998), "CiteSeer: An Autonomous Web Agent for Automatic Retrieval and Identification of Interesting Publications". In Proceeding of the 2nd International Conference on Autonomous Agents, Minneapolis, Minnesota, pp. 116-123, ACM Press.

Buckley, C., Salton, G., and Allan, J. (1994), "The Effect of Adding Relevance Information in a Relevance Feedback Environment". In Proceedings of the ACM SIGIR'94, pp. 292-298, ACM Press.

Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, A., and Slattery, S. (2000), "Learning to Construct Knowledge Bases from the World Wide Web". Artificial Intelligence, 118(1-2), pp. 69-113.

Eguchi, K. (2000), "Incremental Query Expansion Using Local Information of Clusters". In Proceedings of the 4th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2000), Vol. 2, pp.310-316.

Frakes, W. B., and Baeza-Yates, R. (1992). Information Retrieval: Data Structures and Algorithms, Prentice Hall.

Joachims, T., Freitag, D., and Mitchell, T. (1997), "WebWatcher: A Tour Guide for the World Wide Web". In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, pages 770-775, Nagoya, Japan, Morgan Kaufmann.

Karypis, G., Han, E-H. (2000), "Concept Indexing: A Fast Dimensionality Reduction Algorithm with Applications to Document Retrieval and Categorization". Technical Report #00-016, Department of Computer Science and Engineering, University of Minnesota.

Lieberman, H (1997), "Autonomous Interface Agents". In Proceedings of the ACM Conference on Computers and Human Interface, CHI-97, Atlanta, Georgia, pp. 67-74, ACM Press.

Miller, G., (1997), "WORDNET: An Online Lexical Database". International Journal of Lexicography, 3 (4).

Mizzaro, S. (1997), "Relevance: The Whole Story". Journal of the American Society of Information Sciences, 48(9), pp. 810-832.

Porter, M. F. (1980), "An Algorithm for Suffix Stripping". Program, 14(3), pp. 130-137.

Rocchio, J. (1971), "Relevance Feedback in Information Retrieval". In Salton, G. (ed.), The SMART Retrieval System: Experiments in Automatic Document Processing, pp. 313-323, Prentice Hall.

Salton, G., and McGill, M. J. (1983). Introduction to Modern Information Retrieval, McGraw-Hill.