COLLABORATIVE WEB SEARCH
BASED ON USER INTEREST SIMILARITY

DANIELA GODOY* AND ANALÍA AMANDI
ISISTAN Research Institute, Facultad de Ciencias Exactas, UNCPBA
Campus Universitario, CP B7001BBO, Tandil, Bs. As., Argentina
Also at CQNIECET, Consejo Nacional de Investigaciones Científicas y Técnicas, Bs. As., Argentina

The motivation behind personal information agents resides in the enormous amount of information available on the Web, which has created a pressing need for effective personalized techniques. In order to assist Web search these agents rely on user profiles modeling information preferences, interests and habits that help to contextualize user queries. In communities of people with similar interests, collaboration among agents fosters knowledge sharing and, consequently, potentially improves the results of individual agents by taking advantage of the knowledge acquired by other agents. In this paper we propose an agent-based recommender system for supporting collaborative Web search in groups of users with partial similarity of interests. Empirical evaluation shown that the interaction among personal agents increases the performance of the overall recommender system, demonstrating the potential of the approach for reducing the burden of finding information on the Web.

Keywords: collaborative Web search; recommender systems; user profiling.

1. Introduction

In the last years there has been a crescent interest for effective personalization techniques in order to help users to cope with information search in vast spaces such as the Web. Personal agents acting as Web search assistants rely on user profiles intended to either disambiguate query terms or filtering search results. In this context, cooperative agents actively searching and discovering relevant information on behalf of their users can enrich personalized search experiences by delivering more relevant results according to the knowledge acquired by a community of users.

The idea of personal agents acting on behalf of users and sharing knowledge has been generally hampered by the assumption that agents begin with a predefined, common ontology instead of personalized, diverse ontologies. In the context of an agent-based system, a common ontology serves as a knowledge-level specification of the ontological commitments of the participating agents. In open and heterogeneous environments such as the Web, however, a common ontology can hardly embrace the diverse interests users might have.

*Corresponding author: dgodoy@exa.unicen.edu.ar
In this paper, we propose an agent-based recommender system for collaborative Web search which involves agents that can meaningfully communicate using diverse personal ontologies consisting of conceptual representations of user interests. In this approach, personal agents learn individual user profiles and then look for commonalities in the profiles of other users for collaboration. By locating users with similar interests, agents overcome their lack of shared meaning and gain the ability of exchanging information to effectively assist users.

In contrast to traditional collaborative filtering, which measures the similarity among users considering their entire user profiles (i.e., behavior history or opinions), we propose a novel recommendation method in which a user receives recommendations from other users with partial similarity of interests. Users might have some similarities regarding certain aspects of their preferences, yet they might be completely dissimilar in others. In the proposed recommender system, agents focus cooperation on those interests that are common to the users they are assisting.

The rest of the paper is organized as follows. Section 2 reviews related research. Section 3 describes the system architecture and main components. Sections 4 and 5 are concerned with personalized and collaboration strategies for recommendation, respectively. The recommender system functionality is exemplified in Section 6. Experimental evaluation of the system is reported in Section 7. Concluding remarks and future lines of research are discussed in Section 8.

2. Related Works

Personalized Web search emerged as an alternative to help users to cope with the increase of resources available on the Web by incorporating their own information needs in the retrieval process. Pitkow et al. identified two approaches to Web search personalization: query augmentation and search result processing. In the first approach, a user query is augmented by adding terms to it based on some knowledge of the user context and/or preferences. The second approach consists in processing the search results by filtering or ranking them based on user preferences. Both approaches rely on user profiles intended to either disambiguate query terms or identify relevant documents within search results.

Browsing assistants or personal agents assisting users during Web search observe user browsing and build user profiles in order to anticipate information needs. FERRET, for example, is a Web search agent that is able to use an explicit, a priori knowledge about a user to add context information to the user queries. Similarly, QueryTracker learns user interests based on relevance feedback and daily resubmits user queries to a search engine, monitoring search results for changes. In spite of their capacity to provide personalize assistance with information-related tasks, these autonomous agents do not take advantage of the knowledge and expertise other agents in a community may have acquired.

User profiles combined with category hierarchies have been also explored for detecting the context of user queries. In ARCH, user profiles generated applying
an unsupervised document clustering technique over interesting documents are used to interact with a concept classification hierarchy (such as Yahoo!) and derive an explicit user context for each query session. Liu et al. proposed a technique to learn user profiles from user search histories and use it in combination with a general profile gleaned from the Open Directory Project (ODP) to map a user query into a set of categories. These categories denote the user search intention and are used for query disambiguation.

Collaborative Web search (CWS) exploits repetition and regularity within the query-space of a community of like-minded searchers in order to improve the quality of search results. This approach is implemented in I-SPY, a search engine that records and reuses search histories, thus learning a preference model for a community. I-SPY explicitly asks users to subscribe to a set of predefined communities. SearchTogether, a prototype that enables groups of users to collaborate when searching the Web, explores aspects of interface design for collaborative Web search. Other works have also addressed the problem of exploiting group behavior patterns based on Web search history data for improving Web search effectiveness.

Multi-agent approaches to collaborative information retrieval put forward personal agents acting on behalf of their users and collaborating with one another aiming to improve browsing and/or searching experiences of their users. For example, a multi-agent Web mining system called Collaborative Spiders, which implies across-user collaboration in Web search, is proposed in 3. In this system, there is an agent that performs profile matching to find information potentially interesting to users. Before searching, users have to specify the interest area, privacy or publicity of each search experience. Users then can explore similar already completed search sessions from the user community.

In 28, a multi-agent referral system in which personal agents provide users with answers to their questions is presented. From an agent point of view, other agents are classified as neighbors, which can provide answers to queries, or acquaintances, which are only reached through neighbors. This system uses pre-defined ontologies, which have to be shared by all agents, to facilitate knowledge exchanging among them. Implicit is a multi-agent recommender system based on the concept of Implicit Culture, a generalization of collaborative filtering in which a new member of a community is induced to behave similarly to the other members. Implicit recommends specific information exploiting previous observations about the behavior of other users as asked for similar queries.

In some of these systems, users are burdened with a significant load of work that might hinder the advantages of collaboration. For instance, users are asked to specify their interest areas explicitly in order to access shared search sessions or analyze the results of numerous similar search experiences gathered from the community. Other systems are restricted to use certain predefined knowledge or ontologies to both describe the interests of users and compare them. In environments such as the

http://dmoz.org
Web, however, users not only have diverse information interests which can hardly be covered by general-purpose ontologies, but also have changing interests which can appear or disappear over time.

3. Recommender System Overview

The agent-based recommender system presented in this paper accepts queries from users and exploits the knowledge about each user interests and behavior patterns as well as the interests of other users in order to make personalized recommendations about Web pages. In contrast to the common-ontology paradigm, agents in this system acquire personalized ontologies or conceptual descriptions of individual user interests as a starting point for creating communities with common interests. Figure 1 illustrates the overall recommender system.

Users have their own personal agents or assistants helping them to find relevant information according to their current needs. Each personal agent assisting a user in the system is mainly responsible for acquiring an accurate profile representing the user information preferences, retrieving pages from the Web upon a user query, assessing the level of correspondence of these pages with the user profile, contacting other agents for obtaining recommendations regarding some information need, interacting with the user to present recommendations and processing relevance feedback in order to adapt the user profile over time.

**PersonalSearcher**, a Web search agent that helps users to find interesting documents, plays the role of personal agent in this recommender system. In order to provide personalized assistance, this agent learns a hierarchical representation of a user interests by observing the user browsing behavior on the Web and then assists this user by filtering documents resulting from a traditional keyword-based search. Other personal agents can also be integrated into this framework provided that they use a compatible representation of user interests. For example, agents using the same user profiling approach for alternative purposes or obtaining hierarchical
representations of user interests by means of different strategies can take part of the recommender system and contribute in the information discovering process. These strategies include user profiling approaches which concentrate on obtaining semantically enriched profiles through the representation of hierarchical organized concepts a user is interested in, such as ontology-based user profiling \textsuperscript{13}\textsuperscript{,9} or profiling based on clustering algorithms \textsuperscript{8}\textsuperscript{,16}.

Personal agents, each associated with a different user in the system, are integrated into a multi-agent system of cooperative agents that help users to access, manage and exchange information. Hierarchical descriptions of user interests allow agents to detect partial similarities in the tastes of several users. Thus, agents can establish relationships in different contexts on the base of shared interests in certain topics so that collaboration is focused on the interests two users have in common, regardless of their dissimilarities in other concerns.

Each user group (e.g., all participants of a research project) or the entire set of community members shares one matchmaking agent which facilitates the exchanging of knowledge among the different personal agents. Matchmaker agents are designed to establish and maintain connections between users in the community exhibiting similar interests. In the recommender system, matchmaker agents are in charge of coordinating personal agents, receiving requests and disseminating relevant information to the interested users.

4. Content-Based Recommendation

Personal information agents’ main goal is to recommend Web documents to potentially interested users based on profile matching. In this content-based approach, recommendations are made based on comparison of the user interest areas described in the profile and the content of Web pages. A brief overview of PersonalSearcher, the approach used by this agent to acquire user profiles and the strategy applied to recommend pages based on their content is given in the following subsections.

4.1. PersonalSearcher Overview

Each instance of PersonalSearcher monitors the Web activity of its associated user in order to collect documents the user is interested in. For each article read in a standard browser this agent observes a set of implicit indicators to estimate the extent of the user interest in the displayed Web page (e.g., time spent reading the page, amount of scrolling, if it was bookmarked, etc.). Implicit indicators were used since explicit feedback, even though more reliable, burdens the user with an additional cognitive load caused by the necessity of evaluating each Web page. In the observation process, agents capture experiences exemplifying user interests, such as Web pages a user read or bookmarked for future reading, through non-intrusive observation of user behavior on the Web.

Each experience encapsulates both specific and contextual knowledge describing a particular situation denoting a user interest in a certain piece of information.
Experiences can be divided into three main parts: the description of the Web page content, the description of the associated contextual information, and the outcome of applying the experience in decision making. The first part enables agents to carry out content-based comparison for discovering and retrieving topical related information, whereas the second and third part allow agents to act based on both the current user context as well as an estimated level of confidence in each candidate recommendation.

Web page contents are represented using a bag-of-words approach. This is, each page is described by a feature vector in a space in which each dimension corresponds to a distinct term associated with a weight indicating its importance. The resulting representation of a Web page is, therefore, equivalent to a $t$-dimensional vector $e_i = \langle (t_1, w_1), \ldots, (t_t, w_t) \rangle$, where $w_j$ represent the weight of the term $t_j$ in the experience $e_i$. The page address, date and time on which the visit took place as well as other contextual data are also added to the experience in order to enable context-aware recommendation. Finally, experiences keep track of the patterns of received feedback regarding actions undertaken based on the knowledge they provide, starting from the level of interest the user showed in the page calculated according to the observed interest indicators.

Experiences are analyzed to learn a conceptual description of user interests and organized within user profiles to be applied in recommendation. Identification of categories or topics a user is interested in is based on clustering of similar past experiences. Thus, user profiles are built starting from scratch and constantly refined as new experiences representing some user interests become available, using relevance feedback as the main source of information.

Users interact with their PersonalSearcher agents by expressing information needs through a set of keywords as in traditional search engines. In turn, agents post these queries to some of the most popular search engines, receiving documents that cover a wide portion of the Web. Incoming Web pages are evaluated by computing their relevance degree regarding the user profile to determine whether to recommend them for future reading. Once agents present some recommendations, the behavior of users is again observed to perform profile adaptations based on the acceptance/rejection of agent actions.

4.2. User Profiling Approach

To provide effective assistance agents depend on the knowledge they have about individual users contained in their profiles, i.e. models of user interests, preferences and habits. PersonalSearcher employs a user profiling technique designed to support incremental learning and adaptation of user profiles. This technique addresses representational issues of profiles, modeling multiple user interests at several abstraction levels and allowing the extraction of comprehensible views of user interests.

In a user profile, each experience $e_i$ collected by a personal agent is attached with a relevance value denoted $rel_i$, i.e. the user profile consists of pairs $\langle e_i, rel_i \rangle$, where
\(e_i\) is the user experience encoding mainly the Web page the user found interesting and \(rel_i\) represents the evidence about the user interest in that experience. This relevance value is confined to the [0, 1] interval and is established according to the evidence collected by observation (i.e., the interest in a visited Web page estimated using implicit interest indicators and the subsequent feedback).

The relevance of experiences is used for both gaining confidence in experiences which are more representative of the user interests as well as forgetting old recommendations. Important experiences in the profile are those that lead to successful recommendations, so that pages which are similar to these experiences have more chances of being recommended. In contrast, irrelevant experiences that do not provide good recommendations gradually lose relevance, adapting the profile to recognize and eventually remove no longer interesting topics.

The user profiling technique is built upon a clustering algorithm, named WebDCC (Web Document Conceptual Clustering) \(^6\), that allows agents to acquire profiles starting from experiences without an a priori knowledge of user interest categories, so that the learning process is completely unsupervised. This algorithm belongs to the conceptual clustering paradigm \(^12\), which includes clustering and characterization, i.e., the formation of intentional concept descriptions for extensionally defined clusters. In contrast to other profiling approaches, the use of conceptual clustering results in more comprehensible and semantically enhanced user profiles, which enable collaboration with other agents at a conceptual level.

Hierarchies of concepts produced by WebDCC algorithm are classification trees in which internal nodes represent concepts and leaf nodes represent clusters of experiences. The root of the hierarchy corresponds to the most general concept, which comprises all the experiences the algorithm has seen, whereas inner concepts become increasingly specific as they are placed lower in the hierarchy, covering only subsets of experiences by themselves. Finally, terminal concepts are those with no child concepts but clusters.

In other words, a hierarchy is conformed by a set of concepts \(C = \{c_1, c_2, \ldots, c_n\}\), which are gradually discovered by the algorithm as new experiences become available. In order to automatically assign novel experiences to existing concepts, a text-like description given by a set of weighted terms \(c_i = \langle (t_1, w_1), \ldots, (t_m, w_m) \rangle\) is associated to each concept during the process of concept formation. This description constitutes a linear classifier for the category and emerges from observing the common features of experiences in the category and those a novel experience should have in order to belong to it.

Leaves in the hierarchy correspond to clusters of experiences belonging to all the ancestor concepts. Intuitively, clusters are groups of experiences whose members are more similar to one another than to members of other clusters, so that clusters group highly similar experiences observed by the algorithm. In general terms, a set of \(n_i\) experiences belonging to a concept \(c_i\) and denoted \(E_i = \{e_1, e_2, \ldots, e_{n_i}\}\), are organized into a collection of \(k\) clusters below \(c_i\), \(S_{ji} = \{s_{1i}, s_{2i}, \ldots, s_{ki}\}\), containing elements of \(E_i\) such that \(s_{li} \cap s_{pi} = \emptyset\), \(\forall l \neq p\).
WebDCC integrates classification and learning by sorting each experience through the concept hierarchy and simultaneously updating it. Upon encountering a new experience, the algorithm incorporates it below the root of the existing hierarchy and then recursively compares the experience to each child concept as it descends the tree. When the experience cannot be further classified down, the algorithm decides whether to incorporate the experience into a cluster or create a new singleton cluster or category.

The incorporation of experiences to the hierarchy is followed by an evaluation of the current hierarchical structure in order to determine whether novel concepts can be created or some restructuring is needed. In this step, meaningful concepts can be extracted to refine the hierarchy and previously discovered concepts may be reorganized (re-structuring operators such as merging, splitting and promotion are used) taking into account the recently acquired knowledge.

In this algorithm the formation of concepts is driven by the notion of conceptual cohesiveness or intra-cluster similarity. Highly cohesive clusters are assumed to contain similar experiences belonging to the same category, whereas clusters exhibiting low cohesiveness are assumed to contain experiences concerning to distinctive sub-categories. In the last case, concepts are extracted enabling a re-partitioning of experiences and the identification of sub-topics.

Ultimately, the incremental clustering of experiences outputs a hierarchical set of classifiers, each based on its own set of relevant features, as a combined result of a feature selection method for deciding on the appropriate set of terms at each node and a supervised learning algorithm for constructing a classifier for such node. Figure 2 shows an example of a hierarchical clustering solution achieved with the algorithm to represent a user profile (actual words are not showed, but their stems resulting of applying a stemming algorithm when processing Web pages).

The user in this example seems to be interested in two broad categories sports and politics, and several sub-categories. In politics, three distinctive interests are revealed in the hierarchy: Argentinean and U. S. governments and political science.
This user is also interested in several sports, such as basketball, ski and tennis, and some more specialized interests, such as certain tennis and basketball players and teams. For example, regarding tennis, the user is concerned about the activities of two Argentinean players as Mariano Zabaleta and Paola Suarez.

In this profile, it is possible to appreciate the advantages of a personalized hierarchy of interests over the use of a general-purpose ontology. In contrast to topics such as NBA and political science that can be easily found in any ontology, an ontology-based profile might not be able to represent the user interest in Emanuel Ginobili, Paola Suarez and Mariano Zabaleta or still less well-known local players. Moreover, experiences define categories in an extensional sense and at the same time provide a fine-grained description of user interests since actual experiences are at the bottom level of the hierarchy. In consequence, a profile not only represents the interest in a given basketball player within basketball and sports, but also the specific type of documents the user might be interested in within these categories (e.g., personal life, professional achievements, etc.).

4.3. Relevance-Based Recommendation

For personal agents, user profiles consist of a set of past user reading experiences organized in a conceptual hierarchy. To recommend a previously unseen Web page, an agent retrieves similar experiences from the user profile assuming that the user interest in some page will resemble the interest in similar past experiences. Hierarchical concepts or classifiers in the profile bias the search toward the most relevant experiences. This is, each candidate page has to be classified into the different categories until selecting the n best matching experiences.

The confidence in a Web page to be recommended, denoted $\text{conf}(r_i)$, is calculated by a function that aggregates the global similarity of an experience $e_k$ with the candidate page $r_i$ and the relevance of this experience. To assess the confidence in recommending $r_i$ given the experience $e_k$, a weighted sum of the confidence value of each similar retrieved experience is calculated as follows:

$$\text{conf}(r_i) = \frac{\sum_{k=1}^{n} w_k \cdot \text{rel}_k}{\sum_{k=1}^{n} w_k}$$

(4.1)

where $n$ is the number of similar experiences retrieved, $\text{rel}_k$ is the relevance of experience $e_k$ and $w_k$ is the contribution of each experience according to its similarity. This method is based on the well-known distance-weighted nearest neighbor algorithm \(^{14}\). Each experience has a weight $w_k$ according to the inverse square of its distance from $r_i$, i.e., $w_k = \frac{1}{(1 - \text{sim}(e_k, r_i))^2}$. where $\text{sim}(e_k, r_i)$ is the similarity between the item to be recommended $r_i$ and the experience $e_k$ which is the cosine similarity of their contents.

Finally, if the confidence in recommending $r_j$ is greater than a certain confidence threshold, the page is recommended. Items with confidence below this threshold are considered not interesting enough to be presented to users. If the result of a
recommendation is successful according to user feedback, the relevance $rel_k$ of the corresponding experiences in the profile is increased and, possibly, new experiences are added. If the result of a recommendation is a failure according to user feedback, agents learn from the mistake by decreasing the relevance of the experiences that have led to the unsuccessful recommendation.

5. Collaborative-Based Recommendation

Loosely coupled groups of users with similar interests, can take advantage of existing knowledge in the community they are immersed in through cooperation. The possibility of collaboration among agents fosters knowledge sharing and, consequently, potentially enriches the achievable results of individual agents by taking advantage of the experience acquired by other agents.

A well-known problem of pure content-based agents is their tendency to over-specialization, so that users only receive recommendation of documents highly similar to those already seen in the past, whereas collaboration brings diversification to the recommendation process. Hence, individual instances of PersonalSearcher can improve information retrieval quality and efficiency acting not just as isolated agents, but as a part of a multi-agent system of collaborative agents.

In order to enable knowledge sharing in a multi-agent system setting, issues such as how agents determine if two users know the same semantic concepts, how knowledge is effectively exchanged between them and how collaboration affects the group performance at the assigned collective task have to considered. The mechanisms to handle these issues, implemented by matchmaking agents in the proposed recommender system, are described in the following subsections.

5.1. Identifying Shared Interests

Instead of committing to a common ontology, agents in the recommender system base collaboration on the comparison of user profiles. Thus, the system provides recommendations to target users based on neighbors with partially similar long-term interests. If two agents agree on the meaning of two or more concepts via the comparison of the corresponding interest hierarchies, these concepts translate to each other in the system, which enables future information exchange.

Most methods for ontology comparison involve specific tools for constructing and merging complex ontologies requiring the intervention of human experts. Conversely, the Triple Matching-Model (MD3) adopted to compare hierarchies learned by personal agents in the proposed recommender system allows to automatize the process of evaluating the similarity among concepts across multiple personal ontologies or user profiles.

MD3 aims at finding quantitative values of similarity among concepts in two separated ontologies by comparing concept descriptors as well as concept interrelationships. Instead of relying on semantic labels, MD3 compares the representations...
Collaborative Web Search based on User Interest Similarity

Fig. 3. Example of interest hierarchies belonging to three different users

of concepts (e.g., associated words) and defines a similarity model in terms of common features of these representations. In turn, similarity relationships allow agents to create anchors between two personal ontologies or profiles while keeping each of them autonomous.

In other words, this method detects which sets of terms are similar and, therefore, which concepts are good candidates for establishing links across user profiles. Figure 3 shows the profiles of three users, the one described in Section 4.2 and two additional profiles of users sharing some interests. The comparison of hierarchies belonging to users $U_a$ and $U_b$ should result in a link between the concept Argentinean presidency as it is shown in the figure.

Links between profiles denoting shared interests can be considered a weak form of integration of personal ontologies, since no further inferences can be made about concept relationships. However, establishing this kind of relationships is particularly useful in dynamic user profiling in which profiles are expected to change over time, novel interests are prone to appear and old interests will be likely forgotten so that links between profiled should be adapted accordingly.

MD3 matching process is defined over a single hierarchy connecting two independent concept hierarchies via an imaginary root. Using the interconnected hierarchies, the matching process is applied in successive steps to different components: (1) lexicon matching, (2) feature matching, and (3) semantic-neighborhood matching. The global similarity value $S(a, b)$ between two concepts $a$ and $b$ belonging to two different profiles, $p$ and $q$, is the weighted sum of the similarity of each of the mentioned components:

$$S(a^p, b^q) = w_l * S_l(a^p, b^q) + w_f * S_f(a^p, b^q) + w_n * S_n(a^p, b^q)$$

where $S_l$, $S_f$, and $S_n$ denote the lexical (i.e., the similarity among names), feature and semantic neighborhood similarities, respectively, and $w_l$, $w_f$, and $w_n$ are their corresponding weights, that must add up to 1. The lexicon matching in MD3 refers to the number of common and different words in the concept labels. Hierarchies learned by personal agents, however, have not associated labels, so that
only the feature and semantic neighborhood similarities are considered for hierarchy comparison (i.e., \( w_3 = 0 \)).

The feature matching \( S_f(a^p, b^q) \) between two concepts \( a^p \) and \( b^q \), where \( A \) and \( B \) are the respective term sets, is determined by the cardinality of the intersection and the difference between \( A \) and \( B \) as follows:

\[
S_f(a^p, b^q) = \frac{|A \cap B|}{|A \cap B| + \alpha(a^p, b^q) \times |A - B| + (1 - \alpha(a^p, b^q)) \times |B - A|} \tag{5.3}
\]

The value of \( \alpha \) can be expressed as a function of the depth of concepts, which corresponds to the shortest distance from the concepts to the virtual root:

\[
\alpha(a^p, b^q) = \begin{cases} 
\frac{\text{depth}(a^p)}{\text{depth}(a^p) + \text{depth}(b^q)} & \text{if depth}(a^p) \leq \text{depth}(b^q) \\
1 - \frac{\text{depth}(a^p)}{\text{depth}(a^p) + \text{depth}(b^q)} & \text{if depth}(a^p) > \text{depth}(b^q)
\end{cases} \tag{5.4}
\]

In measuring the similarity between two concepts, MD3 also includes their relationships with other concepts in the hierarchies, assuming that two concepts are semantically similar if they are related to the same group of concepts denoted as its semantic neighborhood. Thus, comparing concept relationships is transformed into a comparison of corresponding semantic neighborhoods.

The notion of semantic neighborhood, that is the set of concepts whose distance to a given concept is within a specified radius, is used to calculate neighborhood similarity. The distance between two concepts in a hierarchy is measured as the shortest path formed by the smallest number of undirected arcs that connect these concepts. Since distance is a metric function that satisfies the property of minimality (i.e., the self-distance is equal to zero), the semantic neighborhood of a concept contains the concept itself. Figure 3 depicts the immediate semantic neighborhood of the concept "governments or presidencies."

Equation 5.5 gives a formal definition of the semantic neighborhood \( N \), where \( a^o \) and \( c^o_i \) are concepts in an ontology \( o \), \( r \) is the specified radius, and \( d(a^o, c^o_i) \) is the distance between the two concepts.

\[
N(a^o, r) = \{c^o_i\} \text{ such that } \forall i d(a^o, c^o_i) \leq r \tag{5.5}
\]

The calculus of semantic-neighborhood matching \( S_n \) is a recursive process, because comparing concepts in the semantic neighborhoods is also a similarity evaluation. This recursion stops when the specified radius is reached, at which point concepts can be compared based on feature matching. Given two concepts \( a^p \) and \( b^q \) from ontologies \( p \) and \( q \), where \( N(a^p, r) \) has \( n \) concepts and \( N(b^q, r) \) has \( m \) concepts, and the intersection of the two neighborhoods is denoted by \( a^p \cap b^q \), the value of the semantic-neighborhood matching \( S_n(a^p, b^q, r) \) is a function of the cardinality of the semantic neighborhoods \( N \) and the approximate intersection \( \cap_n \) between these semantic-neighborhoods, calculated by:
Collaborative Web Search based on User Interest Similarity

\[
\frac{a^p \cap_n b^q}{a^p \cap_n b^q + \alpha(a^p, b^q) \ast \delta(a^p, a^p \cap_n b^q, r) + (1 - \alpha(a^p, b^q)) \ast \delta(b^q, a^p \cap_n b^q, r)} \quad (5.6)
\]

with

\[
\delta(a^p, a^p \cap_n b^q, r) = \begin{cases} |N(a^p, r)| - |a^p \cap_n b^q| & \text{if } |N(a^p, r)| > |a^p \cap_n b^q| \\ 0 & \text{otherwise} \end{cases} \quad (5.7)
\]

The intersection over semantic neighborhoods is approximated by the similarity of concepts across neighborhoods:

\[
a^p \cap_n b^q = \left[ \sum_{i \leq n} \left( \max_{j \leq m} S(a^p_i, b^q_j) \right) \right] - \varphi \ast S(a^p, b^q) \quad (5.8)
\]

with

\[
\varphi = \begin{cases} 1 & \text{if } S(a^p, b^q) = \max_{j \leq m} S(a^p_i, b^q_j) \\ 0 & \text{otherwise} \end{cases} \quad (5.9)
\]

where \( S \) is the semantic similarity of concepts and, \( a^p_i \) and \( b^q_j \) are concepts in the semantic neighborhood of \( a^p \) and \( b^q \) respectively.

For each pair of users, a profile comparison algorithm based on MD3 model is run to identify the concepts they have in common. Essentially, this algorithm carries out breadth-first traversals of the conceptual hierarchies representing the interests of both users. It traverses the first profile, which is considered the reference profile, taking each concept along with its neighborhood, and comparing it with all the concepts in the second profile, which is considered the target profile. The algorithm outputs one or more links denoting similarity relationships between the profiles as the one shown in Figure 3.

5.2. Matchmaking and Social Recommendation

In order to enable information exchange between users, agents have to get to know other users and find the most appropriate ones for collaboration. Each instance of PersonalSearcher in the recommender system is registered to one or more matchmaker agents that accomplish this goal, i.e., matchmakers are in charge of making connections between agents that request recommendations and those that are able to provide them. Thus, agents are fully connected with their peers, but only through the corresponding matchmakers.

In the communication with matchmakers, personal agents provide the knowledge about the main interests of their users (i.e., interest hierarchy, but not individual experiences), resources discovered about shared interests (i.e., candidate recommendations) and, optionally, user current information needs (e.g., a given query...
or categories the user is looking information about). In this situation, matchmaker agents determine which agents are promising candidates for fruitful cooperation by taking advantage of the links discovered between personal ontologies.

Matchmakers determine and keep up to date the relationships between user profiles applying the comparison method described in the previous section for each pair of users periodically. If the comparison of profiles belonging to two users $U_a$ and $U_b$ is successful, agents become aware of which interests their users share with other members of the community. For instance, if $U_a$ and $U_b$ users are both interested in $c_x$ and $c_y$ concepts, the associated agents store social knowledge registering the existence of other agents in the system which might be interested in information regarding to these concepts. This social knowledge allows personal agents to notify matchmakers about the discovery of new relevant information when interesting Web pages belonging to the shared interests is found by their users.

Matchmaker agents store the knowledge gathered from the group of personal agents they are in charge of. In the previous example, after comparing the profiles of users $U_a$ and $U_b$ the matchmaker agent stores knowledge stating that both users are interested in concepts $c_x$ and $c_y$ with a certain degree of confidence $conf_x$ and $conf_y$, where $conf_x = S(c_a^x, c_b^x)$ and $conf_y = S(c_a^y, c_b^y)$. Hence, when the agent assisting $U_a$ delivers novel information about $c_x$ or $c_y$, the agent assisting $U_b$ is contacted to receive this information. The level of relevance of a candidate recommendation $r_i$ about $c_x$ for the producer user $U_a$, i.e., $conf_a(r_i)$, is multiplied by the confidence $conf_x$ in the relationship between $U_a$ and $U_b$, causing an increase or decrease of the overall recommendation relevance.

Ultimately, a recommendation is deemed relevant for a given user if the page overcomes certain relevance threshold, which can be customized according to the amount of information each user is willing to receive from others. Matchmaker agents use a temporary space to store incoming recommendations, which are automatically removed if they were not delivered to users after some time. Users can choose to handle incoming recommendations of all or some particular interest categories at any time. This is, personal agents can be configured to present recommendations matching active queries during Web search or matching certain interests either upon demand or proactively.

For a user $U_a$, member of some community in the system, looking for information about a set of concepts $C_n$ existing in his profile the collaborative search process proceeds as follows. The personal agent assisting this user can detect these information needs ($C_n$) by different methods: a query matches these concepts, documents in a search or browsing session match these concepts, the users explicitly ask for information in one or more topics, etc. If the user if performing a search, the agent will first post the query to multiple search engines on the Web, retrieving a set of candidate recommendations $R_s$. In parallel, the concepts in $C_n$ are used to query the community to produce another set of recommendations, $R_c$, judged relevant by other community members based on their past search or browsing beha-
Collaborative Web Search based on User Interest Similarity

Fig. 4. Example of a user profile visualization and Web search results

haviors. Finally, the fusion of both result sets $R_a \cup R_c$ is presented to the user for examination.

6. Example of User Assistance in Web search

To illustrate how collaboration affects the group performance in discovering information, the system functionality is exemplified using the three different users depicted in Figure 3. A visualization of this profile in PersonalSearcher as well as the result of a Web search using the query ‘NBA ginobili’ are shown in Figure 4, whereas a set of content-based recommendations the agent made while the user was searching the Web are depicted in Figure 4. For obtaining these recommendations PersonalSearcher fused the results of several search engines and identifies Web pages which are potentially relevant to the user by comparing them with the user profile.

The function of personal agents is to recommend novel Web pages to users as they are searching the Web, but also through collaboration with other agents. $U_b$ and $U_c$ have not interests in common, but both share one or more interests with $U_a$. $U_a$ can take advantage of the information discovered by $U_b$ and $U_c$, whereas they both can be benefited by the information that $U_a$ might discover related to
the shared interests. $U_b$ and $U_c$ will no profit from their interaction due to their orthogonal profiles.

From profile comparison, the matchmaker agent determines that $U_a$ and $U_c$ are both interested in *Emanuel Ginobili* basketball player and that users $U_a$ and $U_b$ are both interested in the same political subjects except for *political science*. The first conclusion is drawn from the similarity between the two concepts that is calculated as follows:

$$e_{ginobili}^a = \{\text{ginobili}, \text{spur}, \text{san}, \text{antonio}, \text{player}, \text{manu, emanuel}\}$$
$$e_{ginobili}^c = \{\text{ginobili, nba, basketball, spur, game, team, san, antonio, manu}\}$$

which are denoted $e_{ginobili}^a$ and $e_{ginobili}^c$. The MD3 model calculates the feature similarity between these two concepts are follows:

$$|A \cap C| = \{\text{ginobili, spur, san, antonio, manu}\} = 5$$
$$|A - C| = \{\text{player, emanuel}\} = 2$$
$$|C - A| = \{\text{nba, basketball, game, team}\} = 4$$
Collaborative Web Search based on User Interest Similarity

Fig. 6. Example of collaborative-based recommendations

\[
\alpha(e_{ginobili}^a, e_{ginobili}^c) = 1 - \frac{\text{depth}(e_{ginobili}^a)}{\text{depth}(e_{ginobili}^a) + \text{depth}(e_{ginobili}^c)} = 1 - \frac{3}{3 + 2} \approx 0.40
\]

then

\[
S_f(e_{ginobili}^a, e_{ginobili}^c) = \frac{5 + 0.40 \times 2 + 0.60 \times 4}{5} \approx 0.61
\]

To calculate the neighborhood similarity of both concepts, the semantic neighbors in a radius of one include only the parent concepts, since none of the concepts
have children.

\[ e_{\text{ginobili}}^a \cap e_{\text{ginobili}}^c = \max \left\{ S(e_{\text{nba}}^a, e_{\text{sports}}^c), S(e_{\text{nba}}^a, e_{\text{ginobili}}^c) \right\} + \]
\[ \max \left\{ S(e_{\text{nba}}^a, e_{\text{sports}}^c), S(e_{\text{nba}}^a, e_{\text{ginobili}}^c) \right\} - \varphi \ast S(a^p, b^q) \]
\[ = 0.62 + 0.61 - 1 * 0.61 \]
\[ = 0.62 + 0.61 - 1 * 0.61 \approx 0.62 \]

with \( \varphi = 1 \). The neighborhoods of both concepts is composed of two elements, then the \( \delta \) function results in \( 2 - 0.62 = 1.38 \).

\[ S_n(e_{\text{ginobili}}^a, e_{\text{ginobili}}^c, 1) = \frac{0.62}{0.62 + 0.4 \ast 1.38 + 0.60 \ast 1.38} \approx 0.31 \]

and

\[ S(e_{\text{ginobili}}^a, e_{\text{ginobili}}^c) = 0.5 \ast 0.61 + 0.5 \ast 0.31 \approx 0.46 \]

Figure 6(behind) shows the recommendations received by \( U_a \) starting from the information found by both \( U_b \) and \( U_c \) on the Web. The former provided some recommendations about politics belonging to the categories Bush and Kirchner presidents. The latter instead contributed with recommendations about Emanuel Ginobili basketball player, but did not send recommendations about Maradona as this is not an interest shared with \( U_a \). Figure 6(front) depicts the recommendations received by \( U_c \) from the other user in the system sharing the interest in Emanuel Ginobili which is \( U_a \).

Figures 7(a) and (b) depict the position of users in a ego-centric social network composed of the three users \( U_a, U_b \) and \( U_c \) integrated in the recommender system. The vertical distance of users in the landscape is given by the their similarity regarding either a given interest category or the overall profile correlation. From \( U_a \) perspective, \( U_b \) is closer than \( U_c \) regarding political issues, but user \( U_c \) is the closest regarding sports. By selecting a person in the visualization the user has access to the recommendations coming from this person, which have not been yet delivered. This visualization offered by PersonalSearcher agents provides a social browsing tool that enables users to become aware and keep abreast of other users. Thus, visualization of this network supports exploration and discovery of the social context of users.

7. Empirical Evaluation

Evaluation of personal agents and personalization systems in general is a difficult issue since it involves purely subjective relevance assessments. Most datasets are assembled to evaluate learning algorithms and, consequently, do not provide relevance judgments of users about documents. A specially suited dataset to evaluate recommendation is Syskill&Webert Web Page Ratings\(^b\) since it contains the ratings of a single user about the interestingness of Web pages.

\(^b\)http://kdd.ics.uci.edu/databases/syskillwebert/syskillwebert.html
Syskill&Webert Web pages belong to four different categories: Bands (61 pages), Goats (74 pages), Sheep (70 pages) and BioMedical (136 pages). In addition to the topical classification, a single user manually rated each page in a three point scale: hot or very interesting (93 pages), medium or quite interesting (11 pages) and cold or not interesting at all (223 pages). From the original collection we removed ‘Not Found’ pages and those pages with not assigned rating, yielding to a total of 327 Web pages.

Experimental evaluation described in this section aims to investigate how the generation of explicit user profiles can improve predictive accuracy in content-based recommendation as well as provide a means for identifying like-minded users for collaboration. From the perspective of the user who assigned rating to Web pages in the collection, regarded as the target user for experimentation, received content-based and collaborative recommendations were evaluated. By means of these experiments, we were able to assess several evaluation measures regarding both aspects to demonstrate the recommender system behavior empirically.

7.1. Evaluation of Content-Based Recommendation

In order to evaluate the content-based recommendation approach we first created a user profile using the Web pages and ratings of the Syskill&Webert collection. We randomly selected 229 pages ($\simeq 70\%$) that were used for training, i.e. profile learning, reserving for testing the remaining 98 pages ($\simeq 30\%$). This partition was made trying to keep the same proportion of examples in each class (hot, medium and cold), then 72 pages ($\simeq 31\%$) of the training set belong to hot and medium classes and the remaining pages to cold class.

The experimental procedure simulates a user interacting with an agent for obtaining recommendations. Each hot or medium page in the training set is added
to the profile of the target user with a relevance equal to 1 and 0.5, respectively. For each page to be recommended in the testing set, denoted \( r_i \), the most similar experiences in the profile are located to calculate \( \text{conf}(r_i) \) and determine whether it should be recommended. The real ratings given by the user to Web pages in the collection surrogates relevance feedback and are used to update the profile according to the success or failure of recommendations. For example, if a cold page in the testing set is actually recommended, a negative feedback equal to -1 is received and the profiles suffers an adaptation.

The assigned ratings were used as a means to measure the performance of personal agents in predicting the relevance of Web pages. Thus, accuracy was calculated by comparing the decisions taken by the agent about pages in the testing set and the real page ratings. Figure 8 shows the accuracy of recommendations as new experiences were incorporated in the profile. If no feedback is learned, new experiences in the profile lead the algorithm to recommend more pages. Most hot and medium pages are recommended, but accuracy decays since also more cold pages are included among the recommendations. Particularly, the magnitude of the reduction in accuracy accounts for the proportion of cold pages in the testing set in which negative feedback is more frequent than positive one.

In contrast, positive and negative feedback based on real ratings cause a significant increase in recommendation accuracy, which fluctuates depending on the pattern of feedbacks. It can be observed in the figure that accuracy is constantly above the trivial rejector (which classifies everything as cold since it is the majority class), so that the profile allows recognizing some positive examples besides rejecting negative ones. In spite of the subjective judgments about the relevance of pages, which implies more than just learning the topics of Web pages (i.e., classify pages into BioMedical, Bands, Goats and Sheep), the profiling approach enables agents to reach good levels of accuracy in content-based recommendation.
7.2. Evaluation of Collaborative-Based Recommendations

In order to evaluate the collaborative-based recommendation approach, we created a group of users, each interested in two of the categories of the Syskill&Weber Web page collection. More precisely, the profile of each user in the community was created to generate collaborative recommendations according to the following steps:

(i) Build a profile using a training set (~70%) of randomly selected experiences in two categories of the Syskill&Weber dataset, setting the initial relevance of experiences in this profile randomly. Even though there will be some intersection between the training set used for the target user and the remaining ones, the relevance assigned to the common experiences will be different, causing the profiles to focus on different aspects;

(ii) Obtain a testing set of Web pages matching the profile built in the previous step. For each category, select the 10 first pages resulting from a Web search performed using the most frequent keywords in the category and exceeding the relevance threshold regarding the profile. For example, using the keywords ‘medical university’ for BioMedical category or simply ‘goats’ for Goats category. This step simulates a user searching the Web and finding novel information to share with other users;

(iii) Inform the matchmaker agent of the user interests and the collected Web pages, so that it can perform profile comparison and generate recommendations for the target user exclusively.

Using this procedure, we created a total of six users interested in different pairs of topics of the Web page collection, leading to interest hierarchies of at least two concepts and some sub-concepts. Figure 9 depicts the hierarchies obtained when running the clustering algorithms for two different users in the experiments sharing some interest in BioMedical. For example, the hierarchy in Figure 9(b) has two subcategories inside BioMedical, one grouping Web pages related to universities and one grouping Web pages about international journals.

Fig. 9. Hierarchies obtained for different users using the Syskill&Weber collection

Collaborative Web Search based on User Interest Similarity 21

ROOT

music
excerpt
song
mono
band
guitar
vocal
pop

ROOT

got(s)

medic(al)

univers(ity)
health
resourc(es)

univers(ity)
center

journal

international

databas(es)
biologi

sequenc(e)

biologi

sequence

study

scienc(e)

resourc(es)

protein

(a)

(b)
7.2.1. Results of User Profile Comparison

In the described group of users, we first evaluated the recommender system capacity of identifying shared interests. Figure 10 shows the similarities between users regarding the different topics in the respective hierarchies from the target user perspective, i.e. how similar the agent assisting the target user considers other users regarding different interests. In the figure it is also shown the average profile similarity calculated as the average of the similarities among the categories in the profile of every user with the ones of the target user. Labels were added to categories in the profiles for illustrative purpose only, but concepts are given by sets of terms (e.g., ⟨goat, milk⟩ for Goats category).

Experimental evidence showed not only that the user profiling approach was able to isolate the involved concepts to generate comprehensible descriptions of user interests, but also that the method for comparing hierarchies effectively distinguished the concepts users had in common. In fact, a similarity threshold of $\simeq 0.5$ allowed detecting all valuable links relating the created profiles. Note that if the average profile similarity is used, only a low similarity threshold would allow collaboration among agents and, in that case, users would have received information about some interesting topics as well as some uninteresting, non-common ones.

7.2.2. Results of Collaborative Recommendation

In this section, we describe the evaluation of collaborative-based recommendation using two different testing sets: (1) the testing set form by Web pages of the Syskill/Webert collection used in Section 7.1 to evaluate content-based recommendations and (2) the sets of Web pages gathered using the profiles of the different users created in Section 7.2.

Figure 11(a) shows the result of recommendation using the first testing set of
Web pages for variations of the confidence threshold. In this experiment, users were asked for opinions about each Web page in the testing set to help the agent assisting the target user to decide whether to recommend a Web page. Thus, the opinions of users sharing interests are averaged and pages exceeding the confidence threshold are presented to the user. The real ratings of Syskill/Webert Web pages were used to obtain F-Measure scores in recommendation. The decrease in F-Measure as the threshold is higher accounts for a reduction in recall since less pages are recommended.

Naturally, the results of asking other users are less accurate than the results of comparing Web pages with the profile of the target user shown in Figure 8. However, pure content-based agents tend to over-specialization (i.e., users are limited to receive similar information to the pieces already seen), whereas collaborative-based recommendations usually entail higher novelty (not-obviousness). In order to evaluate the capacity of the system to recommend novel Web pages, we used the second testing set of pages obtained through Web search.

Figure 11(b) depicts the proportion of relevant recommendations of those provided by other users. Since pages in the testing set of each user were gathered from the Web there are no real relevance judgments to establish whether they can be considered relevant for the target user. Hence, pages in the testing sets of users were compared with the profile of the target user to determine if they can be considered interesting.

The average number of relevant Web pages received by the target user per user in the community and its standard deviation are reported for variations of the relevance threshold. This threshold indicates the minimum similarity with the profile of the target user a page should have in order to be considered relevant or interesting. The lower the relevance threshold the higher the number of relevant recommendation since a high threshold demands Web pages to be very similar to past experiences in order to be considered interesting. For a relevance threshold
of 0.4 approximately 70% of the pages that a user might find on the Web will be also interesting for the target user and then the diversity of recommendations is considerably improved.

8. Conclusions

In this paper we have described a recommender system based on personal agents assisting a group of users to find Web pages in a collaborative fashion. These agents learn conceptual hierarchies describing individual user interests through observation of user behavior on the Web. In turn, the knowledge acquired by multiple users sharing interests in different topics is exploited by means of interactions among their personal agents.

Identification of like-minded users is the basis for collaboration since it fosters social interaction. In this regard, the contributions of this paper are twofold. First, we propose a recommender system that combines content-based and collaborative filtering for information search and discovery on the Web. Second, we integrate learning of hierarchical representations of user interests with an algorithm for comparing hierarchies that allows agents to establish relationships based on the common interests of their corresponding users. On the one hand, agents acquire and maintain personalized ontologies instead of committing to a pre-defined shared ontology, which might be impractical in vast environments such as the Web, considering also the need of dynamic user profiling (e.g. changes in interests, topic drifts, etc.). On the other hand, recommendation is based on partial similarity of interesting topics on which a user can agree with others about only certain aspects of their preferences. Instead of user correlation, collaboration is focused exclusively on common interests, preventing dissimilar ones from affecting the outcome of recommendation.

Experiments carried out to evaluate the effectiveness of the system demonstrated the potential of the approach for reducing the burden of finding information on the Web. Our empirical findings indicated that the system is capable of both delivering accurate content-based recommendations by comparing Web pages with the acquired user profiles and diversifying the list of resulting recommendations through information exchange with users sharing similar interests. In this aspect, the method for comparing user profiles allows to assess the resemblance of interesting topics existing in two user profiles as the basis for contacting cooperative agents.

Future works include the analysis of issues related to trust, reputation and reliability, which are becoming increasingly important in recommender systems, in the context of the proposed system. This involves obtaining expertise information about users (i.e. if the person making the recommendation is an expert in a certain topic the recommendation can be considered more valuable), developing trust in other agents (i.e. agents can ask other reliable agents in given topics for their opinions and use their trust values to decide about the relevance of certain pieces of information), and combining these different aspects with information about the similarity of preferences to compute recommendations.
Acknowledgments

This research has been partially supported by ANPCyT PICT No. 34917.

References

26 Daniela Godoy and Analía Amandi


