6

Personalized Search on the World Wide Web

Alessandro Micarelli[1], Fabio Gasparetti[1],
Filippo Sciarrone[1], and Susan Gauch[2]

[1] Department of Computer Science and Automation
Artificial Intelligence Laboratory
Roma Tre University
Via della Vasca Navale, 79 - 00146 Rome, Italy
{micarel, gaspare, sciarro}@dia.uniroma3.it
[2] Information & Telecommunication Technology Center
University of Kansas
2335 Irving Hill Road, Lawrence Kansas 66045-7612
sgauch@ittc.ku.edu

Abstract. With the exponential growth of the available information on the World Wide Web, a traditional search engine, even if based on sophisticated document indexing algorithms, has difficulty meeting efficiency and effectiveness performance demanded by users searching for relevant information. Users surfing the Web in search of resources to satisfy their information needs have less and less time and patience to formulate queries, wait for the results and sift through them. Consequently, it is vital in many applications - for example in an e-commerce Web site or in a scientific one - for the search system to find the right information very quickly. Personalized Web environments that build models of short-term and long-term user needs based on user actions, browsed documents or past queries are playing an increasingly crucial role: they form a winning combination, able to satisfy the user better than unpersonalized search engines based on traditional Information Retrieval (IR) techniques. Several important user personalization approaches and techniques developed for the Web search domain are illustrated in this chapter, along with examples of real systems currently being used on the Internet.

6.1 Introduction

Recently, several search tools for the Web have been developed to tackle the information overload problem, that is, the over-abundance of resources that prevent the user from retrieving information solely by navigating through the hypertextual space. Some make use of effective personalization, adapting the results according to each user's information needs. This contrasts with traditional search engines that return the same result list for the same query, regardless of who submitted the query, in spite of the fact that different users usually have different needs. In order to incorporate personalization into full-scale Web search tools, we must study the behavior of the users as they interact with information sources.

P. Brusilovsky, A. Kobsa, and W. Nejdl (Eds.): The Adaptive Web, LNCS 4321, pp. 195–230, 2007.

There are three information access paradigms that users undertake each time they need to meet particular information needs on the Web hypertextual environment: *searching by surfing* (or *browsing*), *searching by query* and *recommendation*. Recommendation-based systems suggest items, such as movies, music or products, analyzing what the users with similar tastes have chosen in the past [67, 58], see Chapter 12 of this book [12] for details.

In searching by surfing, users analyze Web pages one at a time, surfing through them sequentially, following hyperlinks. This is a useful approach to reading and exploring the contents of a hypertext, but it is not suitable for locating a specific piece of information. Even the most detailed and organized catalogs of Web sites, such as YAHOO! DIRECTORY[1] and the OPEN DIRECTORY PROJECT[2], do not always allow users to quickly locate the pages of interest. The larger the hypertextual environment is, the more difficulty a user will have finding what he is looking for.

The other dominant information access paradigm involves querying a search engine, an effective approach that directly retrieves documents from an index of millions of documents in a fraction of a second. This approach is based on an classic Information Retrieval (IR) model [71] wherein documents and information needs are processed and converted into ad-hoc representations. These representations are then used as the inputs to some similarity function that produces the document result list. Further details about this basic approach can be found in Chapter 5 [55] and 2 [29] [55] of this book.

Information Retrieval has always been characterized by relatively stable information sources and sequences of possibly unrelated user queries. It is usually considered distinct from the Information Filtering (IF) process [59], where the user needs are stable and there are large volumes of dynamically generated collections of documents. The user's interests in IF change relatively slowly with respect to the rate at which information sources become available. The Web is a highly dynamic environment, with information constantly being added, updated and removed, therefore IF prototypes seem to be the most appropriate choice on which to build Web search systems. Nevertheless, IF mostly employs complex representations of user needs and the time needed to perform the retrieval process, that is, matching the incoming stream of information with the model of user's interests, is quite long. This slow response is one of the reasons why IF prototypes have not become a widespread tool to retrieve information from the Web. For a closer examination of the most important user modeling techniques developed for IF, see Chapter 2 of this book [29].

In the last few years, attention has focused on the adaptation of traditional IR system to the Web environment, and related implementations of personalization techniques. The former task is accomplished by periodically collecting newly-created documents through re-crawling, keeping the search system's internal document index updated. This chapter discusses the second topic, personalization techniques and their implementation in real systems.

The two paradigms, searching by query and browsing, coexist: most of the times, browsing is useful when the user does not know beforehand the search domain keywords. Often, the user actually learns appropriate query vocabulary while surfing. Be-

[1] http://dir.yahoo.com
[2] http://dmoz.org

cause searching by query allows users to quickly identify pages containing specific information, it is the most popular way that users begin seeking information [35, 74], making the relevance of this paradigm paramount. For this reason, sophisticated search techniques are required, enabling search engines to operate more accurately for the specific user, abandoning the "one-size-fits-all" method. *Personalized search* aims to build systems that provide individualized collections of pages to the user, based on some form of model representing their needs and the context of their activities. Depending on the searcher, one topic will be more relevant than others. Given a particular need, e.g., a query, the results are tailored to the preferences, tastes, backgrounds and knowledge of the user who expressed it.

In spite of the fact that search engines are the principal tool by which users locate information on the Web, only a few search engines provide tools that adapt to user interaction. Moreover, users often judge these tools as not easy to personalize. In particular, the accessibility of these approaches is low since, as the personalization level increases, the users have more difficulty using these features [41]. There could be several reasons for this phenomenon. First, most personalization techniques are based on user profiles that incorporate information about the user, such as their information needs, interests, and preferences. Users may be uncomfortable with having their personal information stored on an external search system, see Chapter 21 [43]. Second, the personalization of Web search results is a computationally-intensive procedure. A typical search engine usually performs hundreds of queries per second and serves millions of users. Thus, the requirement to provide tailored results in a fraction of a second is not easy to accomplish. Finally, while users are familiar with the current search engines' interface, if the personalization is provided by some sort of new feature, users may find it difficult to understand and profitably use.

This chapter focuses on personalization approaches, techniques and systems developed for search activities, that is, when the user is actively looking for a particular piece of information on the Web. A strongly related topic is *Focused Crawling*, where the search is performed by specific information systems that autonomously crawl the Web collecting pages related to a given set of topics, reducing the network and computational resources. Chapter 7 of this book [53] provides a wide overview on this related topic, with a bias toward approaches which are able to dynamically adapt their behavior during the search according to the alterations of the environment or the given topics of interest.

The most common personalization approaches presented in literature are discussed in the next sections. Related techniques and prototypes are included for each discussed approach. The chapter is organized as follow: Sect. 6.2 provides a brief overview of the personalized search approaches, providing the reader with a broad description of the various methods and techniques proposed in the literature (some of which are fully treated in other chapters of the present volume, e.g., Chapters 2 [29], 9 [75] and 20 [36] of this book). Further details on the above-mentioned approaches are provided in the other sections.

The collection of implicit feedback from the current activity's context or search histories is reported in Sect. 6.3 and Sect. 6.4 respectively. Approaches in which complex and rich representations of user needs are built from user feedback are reported

in Sect. 6.5. Section 6.6 discusses collaborative search approaches while personalized clustering of the results are summarized in Sect. 6.7. Section 6.8 explores how hyperlink-based algorithms can be used to adapt the search engine's result lists to the user needs. Hybrid approaches to personalization are discussed in Section 6.9 and, finally, conclusions are presented in Sect. 6.10.

6.2 A Short Overview on Personalized Search

After a brief introduction of the motivations and goals of the personalized search, it is interesting to examine the personalization approaches and tools proposed to achieve this goal.

We begin with a preliminary taxonomy based on content and collaborative-based distinction. We then move on to how user profiles are implemented in the personalized systems and the typical sources employed to recognize user needs. An overview of the different personalized search approaches, which are discussed in depth later on in this chapter, closes this section.

6.2.1 Content and Collaborative-Based Personalization

Many techniques on which search engines are based on originated from the IR field, e.g., Vector Space Model (VSM) [72, 70], mostly *content-based* techniques, wherein each user is assumed to operate independently. The content of documents is used to build a particular representation that is exploited by the system to suggest results to the user in response to ad-hoc queries (Chapter 5 [55] provides details on document representations). The searching-by-query paradigm is definitely quicker when the user is aware of the problem domain and knows the appropriate discerning words to type in the query [60]. However, analyzing search behavior, it is possible to see that many users are not able to accurately express their needs in exact query terms. The average query contains only 2 to 3 terms [50, 78].

Due to *polysemy*, the existence of multiple meanings for a single word, and *synonymy*, for the existence of multiple words with the same meaning, the keyword search approach suffers from the so-called *vocabulary problem* [27]. This phenomenon causes mismatches between the query space and the document space, because a few keywords are unlikely to select the right pages to retrieve from sets of billions [26]. Synonymy causes relevant information to be missed if the query does not contain the exact keywords occurring in the documents, inducing a recall reduction. Polysemy causes irrelevant documents to appear in the result lists, affecting negatively the system precision. For these reasons, users face a difficult battle when searching for the exact documents and products that match their needs. Understanding the meaning of Web content and, more importantly, how it relates to the real meaning of the user's query, is a crucial step in the retrieval process. Figure 6.1 shows the principal content-based personalization approaches, discussed later in this section.

When the algorithm used to build the result list also takes into account models of different users, the approach is usually named *collaborative* [32, 66]. The basic idea

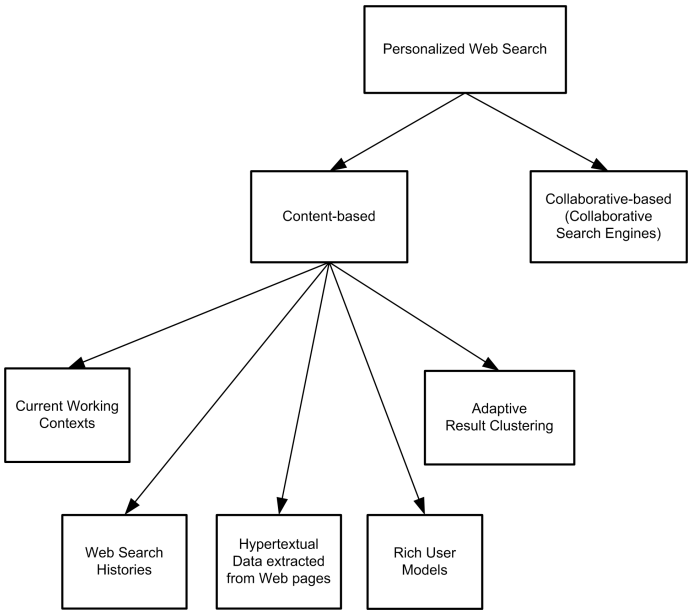

Fig. 6.1. Principal personalization approaches arranged by content/collaborative-based distinction.

behind collaborative-based approaches is that users with similar interests are likely to find the same resources interesting for similar information needs. *Social navigation* is the word coined by Dieberger *et al.* [21] to refer to software that allows people to leave useful traces on Web sites, such as reviews, comments, or votes, used by other people during browsing and searching-by-query.

Because most of the collaborative systems do not employ any search technology, this chapter does not cover them. These systems are discussed in Chapter 20 [36] and Chapter 9 [75] of this book. Two exceptions are EUREKSTER and I-SPY search engines, described in Sect. 6.6, which employ the collaborative or community-based approach to suggest pages that other users who submitted the same query selected frequently. Figure 6.1 shows a taxonomy of collaborative and content-based personalization approaches discussed later in this section.

6.2.2 User Modeling in Personalized Systems

Tracking what pages the user has chosen to visit and their submitted queries is a type of *user modeling* or *profiling* technique, from which important features of users are learned and then used to get more relevant information. Most of the personalized search systems discussed in this chapter employ a user modeling component that occurs during the information retrieval or filtering. Basically, this is the major component needed to provide tailored results that satisfy the particular needs of single users.

In the simplest cases, user models consist of a registration form or a questionnaire, with an explicit declaration of interest by the user. In more complex and extended cases,

a user model consists of dynamic information structures that take into account background information, such as educational level and the familiarity with the area of interest, or how the user behaves over time. For example, the *ifWeb* prototype [6] makes use of user models based on semantic networks [64, 18] in order to create a representation of the available topics of interests. It supports users during Web surfing, acting as hypermedia search assistant (see Sect. 6.5.1 for details).

As an example of a very simple personalized search tool, GOOGLE's *Alerts* is an agent that automatically sends emails to the user each time new results for given query terms become available, both from the Web and News sites. GOOGLE's Alerts builds user models using an *explicit approach* where users explicitly construct the model by describing the information in which they are interested in. In this particular case, the user suggests a set of keywords, sometimes called *routing query*, which must appear in the retrieved documents, thus filtering the information stream. As soon as new information is published on the Web, the system evaluates it according to the stored profile, *alerting* the user of such new and potentially interesting contents. The obtained profiles are relatively simple and act as standard queries. Since the routing queries are suggested by users and the results are never adapted by the system to particular needs or tasks, the system's personalization is really limited.

A further tool named GOOGLE's *Personalized Search* used to deliver customized search results based on user profiles overcomes some of the Alert's problems. The results were instantly rearranged by dragging a series of sliders that define the personalization level concerning pre-defined sets of topics. Basically, while indexing, the engine categorizes pages collected from the Web according to a topic taxonomy. When users submit a query, the system looks through pages associated with their interests, that is, the selected topics, to find matches affecting the search results. Due to the kind of feedback employed to build the profiles, the user is still required to point the system to the information that is considered most interesting or, in some cases, suggesting data to be ignored in the future. For this reason this tool has been replaced with a new technology discussed in Sect. 6.4.

In personalized search systems the user modeling component can affect the search in three distinct phases, showed in Fig. 6.2:

- *part of retrieval process*: the ranking is a unified process wherein user profiles are employed to score Web contents.
- *re-ranking*: user profiles take part in a second step, after evaluating the corpus ranked via non-personalized scores.
- *query modification*: user profiles affect the submitted representation of the information needs, e.g., query, modifying or augmenting it.

The first technique is more likely to provide quick query response, because the traditional ranking system can be directly adapted to include personalization, avoiding repeated or superfluous computation. However, since the personalization process usually takes a long time compared with traditional non-personalized IR techniques, most search engines do not employ any personalization at all. Time constraints that force the system to provide result lists in less than a second cannot be met for all users.

On the other hand, re-ranking documents as suggested by an external system, such as a search engine, allows the user to selectively employ personalization approaches

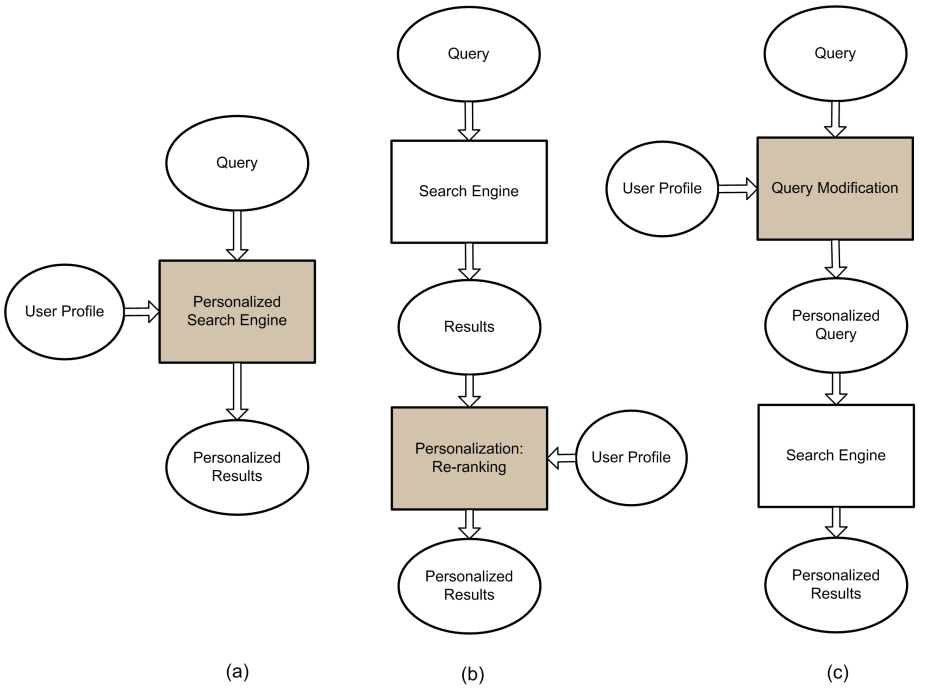

Fig. 6.2. Personalization process where the user profile occurs during the retrieval process *(a)*, in a distinct re-ranking activity *(b)* or in a pre-processing of the user query *(c)*.

able to increase precision. Many systems implement this approach on the client-side, e.g., [62, 54, 77], where the software connects to a search engine, retrieving query results that are then analyzed locally. In order to avoid spending time downloading each document that appears in the result list, the analysis is usually only applied to the top ranked resources in the list, or it considers only the snippets associated with each result returned by the search engine.

Because of the time needed to access a search engine and retrieve the pages to be evaluated, the re-ranking approach implemented via client-side software can be considerably slow. Nevertheless, complex representations of user needs can be employed, considerably improving the personalization performances (see Sect. 6.5).

Finally, profiles can modify the representations of the user needs before that retrieval takes place. For instance, if the user needs are represented by queries, the profile may transform them by adding or changing some keywords to better represent the needs in the current profile. Short queries can be augmented with additional words in order to reduce the vocabulary problem, namely, polysemy and synonymy, which often occur in this kind of keyword-based interaction. Alternatively, if the query retrieves a small number of resources, it is possible to expand it using words or phrases with a similar meaning or some other statistical relations to the set of relevant documents (see *query expansion* technique [7]). The major advantage of this approach is that the amount of work required to retrieve the results is the same as in the unpersonalized scenarios.

Nevertheless, user profiles affect the ranking only by altering the query representations. Unlike ranking that takes place in the retrieval process, the query modification approach is less likely to affect the result lists, because it does not have access to all the ranking process and its internal structures.

6.2.3 Sources of Personalization

The acquisition of user knowledge and preferences is one of the most important problems to be tackled in order to provide effective personalized assistance. Some approaches employ data mining techniques on browsing histories or search engine logs (see Chapter 3 [57]), while others use machine learning [87] to analyze *user data*, that is, information about personal characteristics of the user, in order to learn the knowledge needed to provide effective assistance. The user data usually differs from *usage data*. The latter are related to a user's behavior while interacting with the system. Examples of sources of user data are: personal data, e.g., name, address, phone number, age, sex, education; or geographic data, e.g., city and country.

Techniques such as *relevance feedback* and *query expansion* introduced in the IR field [72, 3] can be employed in the personalization domain in order to update the profile created by users. Basically, to improve ranking quality, the system automatically expands the user query with certain words that bring relevant documents not literally matching the original query. These words are usually extracted from pages in a previously retrieved list of ranked documents that have been explicitly judged interesting by the user through relevance feedback.

Besides considering important synonyms of the original queries' keywords that are able to retrieve additional documents, expansion helps users to disambiguate queries. For example, if the user submits the query '*Jaguar*', the result list will include information on the animal, the car manufacturer, the operating system, etc.. Following relevance feedback on a subset of documents relating to the meaning of interest to the user, the query is updated with words that help the system filtering out the irrelevant pages. Using a lexicon, it is also possible to expand queries such as '*IR*' to '*information retrieval*', increasing the chance of retrieving useful pages.

Even though these techniques have been shown to improve retrieval performance, some studies have found that explicit relevance feedback is not able to considerably improve the user model especially if a good interface is not provided to manage the model and clearly represent the contained information [86]. Users are usually unwilling to spend extra effort to explicitly specify their needs or refine them by means of feedback [5], and they are often not able to use those techniques effectively [79, 85], or they find them confusing and unpredictable [44].

Moreover, studies show that users often start browsing from pages identified by less precise but more easily constructed queries, instead of spending time to fully specify their search goals [84]. Aside from requiring additional time during the seeking processes, the burden on the users is high and the benefits are not always clear (see for example [88]), therefore the effectiveness of explicit techniques may be limited.

Because users typically do not understand how the matching process works, the information they provide is likely to miss the best query keywords, i.e., the words that

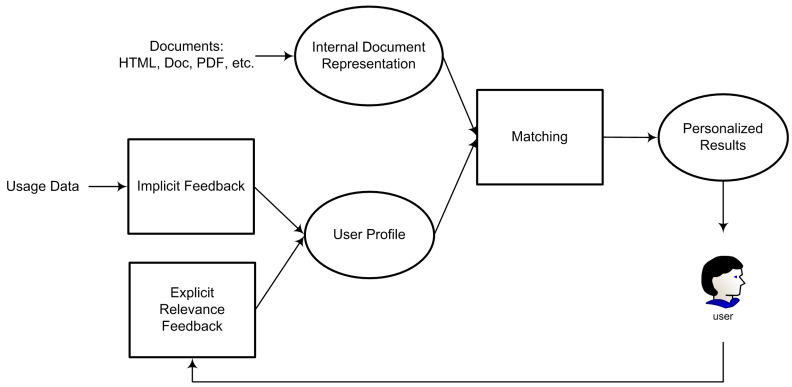

Fig. 6.3. Implicit and Explicit Feedback are used to learn and keep updated the profile of the user used during the personalization.

identify documents meeting their information needs. Moreover, part of the user's available time must be employed for subordinate tasks that do not coincide with their main goal. Instead of requiring user's needs to be explicitly specified by queries or manually updated by the user feedback, an alternative approach to personalize search results is to develop algorithms that infer those needs implicitly.

Basically, *implicit feedback* techniques unobtrusively draw usage data by tracking and monitoring user behavior without an explicit involvement (see Fig. 6.3). Personalized systems can collect usage data on the server-side, e.g., server access logs or query and browsing histories, and/or on the client-side, such as cookies and mouse/keyboard tracking. For a closer examination on implicit feedback techniques see for example [40, 17, 13] and Chapter 21 of this book [43] for the related privacy concerns.

For example, Bharat *et al.* [8] proposes monitoring some current user activity and implicitly building a user profile to provide *content personalization* in Web-based newspaper domain. They suggest that events such as scrolling or selecting a particular article reflect the user's current interest in the given topic. Each event adds a score to the current article, and if it exceeds a certain threshold, the global score increases and the change is reflected in the user profile. Basically, a subset of keywords are extracted from the article and included in the profile with a certain weight that will be updated if the same keyword appears in other articles browsed by the user. When a new article which includes some of the keywords contained in the profile appears, it gets a high score and is included in the personalized newspaper.

The system is somewhat unique in that it allows the user to control the system behavior by controlling the amount of personalization they wish be applied to their results. Sometimes, the user just wants to have an overview on general news, which may require a low-level of personalization since the personalization feature usually filters out information that is judged unrelated to the topics recognized in previous articles read by the user. For this reason, a control bar reduces the effect of personalization, allowing suggestions of popular articles unrelated with the past interests.

Table 6.1. Several important types of personalized search arranged by the type of feedback, implicit or explicit, used to learn the user profile and keep it updated, and the typical data related to the user given as input to the algorithms, such as, resources selected by the user from the results of a search engine, or subsets of pages browsed so far.

Sect.	Personalization based on:	Implicit/Explicit	Typical Input Data
6.3	Current Context	implicit	Word docs, emails, browser's Web pages...
6.4	Search History	implicit	past queries and browsed pages, selected results
6.5	Rich User Models	both	user feedback on results, past queries...
6.6	Collaborative approaches	both	past queries, selected results, user ratings
6.7	Result Clustering	explicit	selected clusters in taxonomies
6.8	Hypertextual Data	both	queries, selected pages...

As a further source of personalization, several desktop search systems, e.g., Copernic, Google Desktop Search, Mac Tiger, Windows Desktop Search, Yahoo! Desktop Search or X1, and several Search Toolbars provide simple access to indexes of information created, copied, or viewed by users. Microsoft's Stuff I've Seen (SIS) project [24] and the associated personalization technique [83, 85] provide personalization by exploiting this type of information. SIS does not involve the retrieval of new information, rather the re-use of what has been previously seen, providing a search interface based on an index of all personal information, such as emails, Web pages, documents, appointments, etc.. The ability to quickly retrieve such data has been proven to be very useful for the user. Essentially, the personalization technique re-ranks the search engine results as a function of a simplified user model based on the keywords occurring in the documents that the user has seen before. This kind of approach is able to use implicit feedback to build and update the user profile which can be used to disambiguate queries. Some of the advantages of this approach to personalization, which is also used by several Web-based personalized systems, is described further on in this chapter.

6.2.4 An Overview on Personalization Approaches

Personalized Search on the Web is a recent research area with a variety of approaches, sometimes tough to arrange in a framework where it is able to identify basic principles and techniques. A possible organization is shown in Table 6.1, where the personalization approaches are arranged by the type of feedback used to build user profiles, and the typical data related to the user given as input during the profiling. Obviously, it is possible to develop systems where more than one search approach is properly combined and implemented (see Sect. 6.9).

The first two approaches, discussed in Sect. 6.3 and Sect 6.4, are based on implicit feedback techniques, where users do not have to explicitly state their preferences or needs. Client-side software captures user interaction and information, such as browsed

pages, edited documents or emails stored on hard drives, which are impossible for the search engine itself to collect and analyze. These pieces of data are very useful to understand the user's current working context, that can in turn be employed for query refinement or as an implicit source of evidence on the user's interests. Personalization based on the *Current Context* exploits this information to recognize the current user needs, which are used to retrieve documents related to the user activities.

If the personalization is limited to the Web *Search History*, we distinguish the related personalized systems from the previous category. The reason is that search engines are able to access this information for each user, with no client-side software requirements. User query histories, resources returned in the search results, documents selected by the user, plus information such as anchor text, topics related to a given Web page, or data such as click through rate, browsing pattern and number of page visits, are easily collected and mined server-side. Moreover, the personalization process can be done during the traditional retrieval process, obtaining a faster response than a distinct post-ranking activity.

Nevertheless, usage data are sometimes not available or they contain too much noise to be successfully exploited by implicit feedback techniques. In that case, explicit user feedback may be the only viable way to learn the user profile and keep it updated.

Most of the time, explicit feedback corresponds to a preferential vote assigned to a subset of the retrieved results. This kind of technique, called relevance feedback is really helpful whenever the user is not able to correctly specify a query, because he can submit a vague query and then analyze the query's results and select the documents that are mostly related to what he is searching. In spite of the negative features of explicit relevance feedback previously discussed, the information collected usually allows the system to build *Rich Representations of User Needs*, composed of more than just Boolean sets or bag-of-words models. Examples of this approach are described in Sect. 6.4.

In environments where a large amount of low-quality items are present, such as the Web, the concept of social filtering is that users help each other to distinguish between high and low quality items by providing ratings for items they have analyzed. All the ratings are collected and can then be used by other users to find the best-rated items.

Delivering relevant resources based on previous ratings by users with similar tastes and preferences is a form of personalized recommendation that can also be applied in the Web search domain, following a *Collaborative approach*. Moreover, since the filtering does not depend on the content of the objects, social filtering is able to provide recommendation for objects such as movies and music, that are usually hard to represent and manage in information systems. Section. 6.6 introduces a few of these collaborative-based systems.

In many cases, search engines retrieve hundreds or thousands of links to Web sites in response to a single query. Although the user may find the material he is looking for in the result list, or at least find Web pages from which the browsing process may begin, the sheer vastness of the results list can make sifting through the retrieved information an impossible task.

One idea to help the users during their search is to group the query results into several *clusters*, each one containing all the pages related to the same topic. In this way,

an overview of the retrieved document set is available to the user and interesting documents can be found more easily. Typically, the pages might be clustered either into an exhaustive partition or into a hierarchical tree structure. The clusters are matched against the given query, and the best ones are returned as a result, possibly sorted by score. A retrieval system that organizes the results into clusters can be considered personalized because the user is able to customize the set of shown results navigating through the clusters driven by their search needs. This kind of *Adaptive Result Clustering* is shortly investigated in Sect. 6.7, while Chapter 13 of this book [11] provides an extensive dissertation on the Adaptive content presentation.

In the same scenario of queries that retrieve a large number of documents, following a given content-based matching function, search engines might assign the same ranks to several resources that share similar content. This is why some search engines include additional factors in their ranking algorithm on top of the query/document content similarity used by traditional information retrieval techniques. These factors may take into account the popularity and/or the authority of a page, in order to assign more accurate scores to the resources during the ranking process. Usually those measures are based on the Web hypertextual data, which is possible to collect by analyzing the set of links between pages. For a closer examination of these measures, such as the PageRank or HITS's authority and hubness, see Chapter 5 of this book [55].

Both the ranking techniques, the traditional IR's and the hyperlink-based algorithms, compute rank values based on page content as a single and global value for each Web page, ignoring any form of personalization based on the user's preferences regarding the quality for an individual page. Recent work aims to extend hyperlink-based algorithms by considering different notions of importance for different users, queries and domains. In other words, the idea is to create personalized views of the Web by redefining the importance assigned by the hypertextual algorithms according to the implicitly expressed user preferences, for example, through previously submitted queries, or explicitly, via a subset of bookmarks or categories in a given taxonomy. Obviously, the query results that match the user-selected topics will be ranked higher by the search engine, providing tailored output for each user. Section 6.8 introduces this personalization based on *Hypertextual data*.

6.3 Contextual Search

Rhodes [68] proposes a new approach for the search named Just-in-Time IR (JITIR) where the information system proactively suggests information based on a person's working context. Basically, the system continuously monitors the user's interaction with the software, such as typing in a word processor or surfing with Internet browsers, in a non-intrusive manner, automatically identifying their information needs and retrieving useful documents without requiring any action by the user. The retrieval process can exploit a variety of data sources, i.e., any number of pre-indexed databases of documents, such as e-mails or commercial databases of articles.

The JITIR approach combines the alerting approach of Google Alert, briefly described in Sect. 6.2.2, with personalization based on the events inside the user's local

working context. Alerting pushes information related to predefined sets of topics toward the user regardless of his current activity, usually requiring a sudden change of user attention. By means of a dynamic user profile kept updated according to changes of the local working contexts, JITIR provides the information tailored to the current user activity.

Describing the JITIR approach, the author suggests three different implementations based on agents. The *Remembrance Agent* presents a list of documents that are related to what the user is typing or reading. *Margin Notes* follows an adaptive hypermedia approach, automatically rewriting Web pages as they are loaded, adding hyperlinks to related local files. The third agent, *Jimminy*, provides information related to the user's physical environment, e.g., spatial location, time of day, subject of conversation, etc., by means of a wearable computer that includes different ways to sense the outside world.

Each of the agents in the JITIR approach share the same back-end system, called *Savant*. It consists of a client-resident search engine that is queried by the agents as the user interacts with the system. The search engine index usually stores public corpora as newspapers or journal articles, and/or personal sources such as e-mail and notes. In order to extract the data needed to build the index, Savant is able to recognize, parse, and index a variety of document formats. During retrieval, the fields extracted from the current document and the ones from the stored documents are compared sequentially, and an overall similarity score is calculated using a linear combination of those similarities. Kulyukin's *MetaCenter* [47] shares many features of the previous prototype, performing automatic query generation according to the current resources the user is working on, e.g., browsed pages or Word documents. The queries are submitted to search engines that operate on online collections of documents to which the user subscribes.

A further instance of the JITIR approach is *Watson* [9, 10]. It monitors the user's actions and the files that he is currently working on to predict the user's needs and offer them related resources. The Watson agent works in a separate window and can track the user across different applications, such as Internet Explorer and Mozilla browsers, and the Microsoft Office suite. As the user's work goes on, Watson looks for related information, following a different context for every open window it is tracking. Relying on the contextual information learned by the agent from the current active window, it generates its own queries to several sources of information and presents them after a result aggregation process. It is also possible to submit explicit queries, which are added to the contextual query while the result post-processing takes place to aggregate results from the different sources.

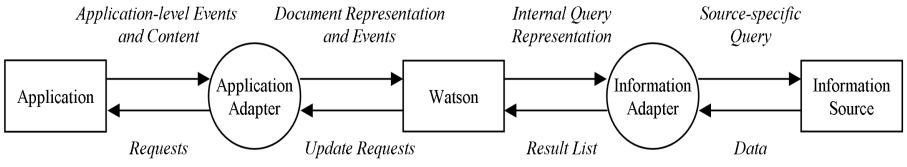

Fig. 6.4. Watson monitors the user activity and sends ad-hoc queries to specific information sources.

According to the authors, Watson uses several sources of information, such as the search engines ALTAVISTA, YAHOO! and DOGPILE, news sources such as Reuters and the New York Times, Blog sites and e-commerce sites. *Application adapters* are used to gain access to internal representations and events generated by user interactions with a specific application, as shown in Fig. 6.4. For example, if the user edits a Word document, the keyboard events trigger Watson to request an updated representation of it. This representation is translated into a query to be submitted to an appropriate source by means of an *Information adapter*. The current task affects the choice of the information source to query, e.g., if the user is editing a medical document, Watson might choose a specialized search engine on this topic.

The user can access the top ranked results from all the relevant sources or filter them by resource type, i.e., Web, news, etc., a capability provided by recent local search systems, such as Apple Tiger's Spotlight and Google Desktop Search. The *TFxIDF* technique [73] is used to create the contextual query based on the currently active window (see Chapter 5 [55] for further details). The bag of words representation in which any document is treated as a set of words regardless of the relations that may exist among them, enhanced with additional formatting information, is used to create a list of term-weight pairs. The top 20 weighted pairs are sorted in their original order of appearance in the document and are used to create the query that is submitted to the information sources. Many heuristics have been considered in order to increase the performance, such as removing *stop words* (common words, such as `I`, `the`, `when`, etc.)and giving more weight to the words that appear at the beginning of documents and those that are emphasized via specialized formatting.

6.4 Personalization Based on Search Histories

User queries are undoubtedly an important source in recognizing the information needs and personalizing the human-computer interaction. A search engine is able to access and process all this information in a non-invasive way, i.e., without installing external proxy servers or client desktop bots, therefore it can tailor the query results based on the previous requests and interests [49]. Simple log-in forms and cookies can be employed in order to identify the user and the related click streams data instead of complex heuristics based on IPs, last access times or user agents data, which cannot be considered entirely accurate [61].

As already noticed, if the user submits a short query, such as *Visa*, it is not clear if he is looking for the credit card company, the policy and procedures to travel to foreign countries, the procedures to change the citizenship, a last name, etc.. The browsing/query history could be a way to weight the different alternatives for example. If the user has recently searched for a flight to a foreign country, a Visa query is more likely to be related to bureaucratic procedures.

Approaches based on search history can be organized in two groups. Offline approaches exploit history information in a distinct pre-processing step, usually analyzing relationships between queries and documents visited by users. Online approaches capture these data as soon as they are available, affecting user models and providing personalized results taking into consideration the last interactions of the user. Even though

the latter approaches provide updated suggestions, an offline approach can implement more complex algorithms because there are usually less urgent time constraints than in an online one.

6.4.1 Online Approaches

Following the first personalization attempt briefly described in Sect. 6.2, Google Labs released an enhanced version of Personalized Search that builds the user profile by means of implicit feedback techniques. In particular, the system records a trail of all queries and the Web sites the user has selected from the results, as shown in Fig. 6.5, building an internal representation of his needs. During the search process, the search engine adapts the results according to needs of each user, assigning a higher score to the resources related to what the user has seen in the past. Unfortunately, no details or evaluations are available on the algorithms exploited for that re-ranking process at present except that contained in the patent application filed in 2004 [92]. Nevertheless, the developers claim they can produce more relevant search results based on what the system learns from the search history, especially when the history contains enough data to be analyzed.

Raghavan and Sever [65] use a database of past queries that is matched with the current user query. If a significant similarity with a past query is found, the past results associated with the query are proposed to the user. The research focuses on the similarity measure used to calculate the query-to-query similarity. This cannot be based on traditional word-to-word IR matching functions, such as the cosine measure, because the short nature of queries makes them particularly susceptible to the vocabulary problems of polysemy and synonymy.

Speretta and Gauch developed the *misearch* system [77], which improves search accuracy by creating user profiles from their query histories and/or examined search results. These profiles are used to re-rank the results returned by an external search service by giving more importance to the documents related to topics contained in their user profile.

In their approach, user profiles are represented as weighted concept hierarchies. The OPEN DIRECTORY PROJECT (ODP) is used as the reference concept hierarchy for the profiles. GOOGLE has been chosen as the search engine to personalize through a software wrapper that anonymously monitors all search activities. For each individual user, two different types of information are collected: the submitted queries for which at least one result was visited, and the *snippets*, i.e., titles and textual summaries, of the results selected by the user. Afterward, a classifier trained on the ODP's hierarchy, chooses the concepts most related to the collected information, assigning higher weights to them. In the current implementation, for comparison purposes, the query and the snippet data are kept distinct and therefore two different profiles are built.

After a query is submitted to the wrapper, the search result snippets are classified into the same reference concept hierarchy. A matching function calculates the degree of similarity between each of the concepts associated with result snippet j and the user profile i:

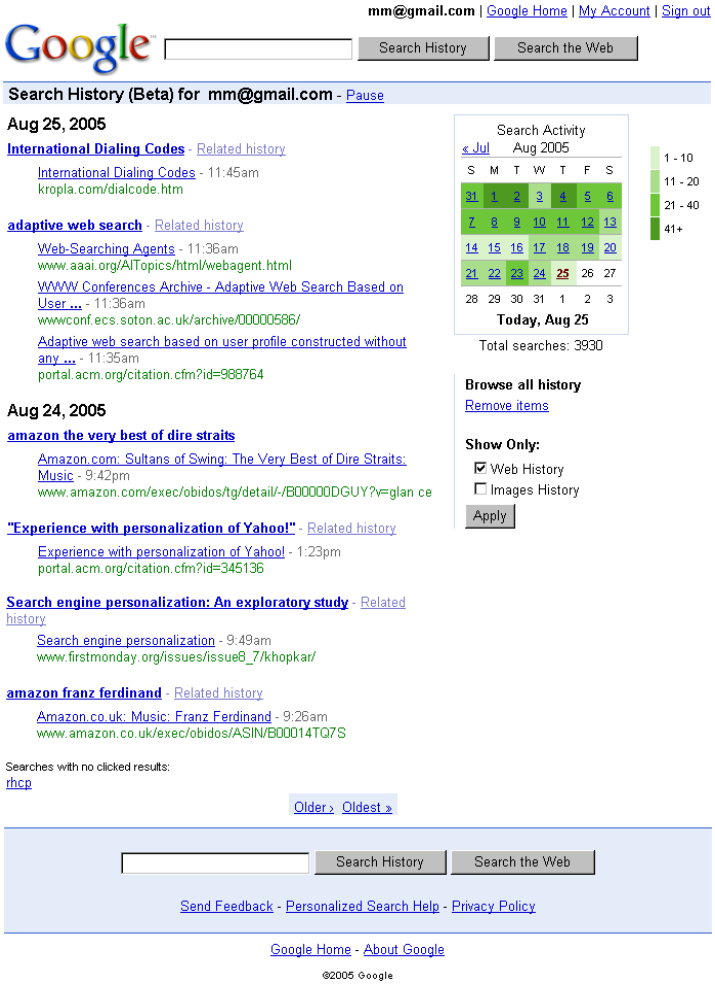

Fig. 6.5. The Search History feature of the Google Labs' Personalized Search records the history of searches and the search results on which the user has clicked. This information is exploited to personalize search results by ranking resources related to what the user has seen in the past higher. *(Reproduced with permission of Google)*

$$sim(user_i, doc_j) = \sum_{k=1}^{N} wp_{i,k} \cdot wd_{j,k} \tag{6.1}$$

where $wp_{i,k}$ is the weight of the concept k in the user profile i, $wd_{j,k}$ is the weight of the concept k in the document j, and N is the number of concepts.

The final weight of the document used for reordering - so that the results that best match the user's interests are ranked higher in the list - is calculated by combining the previous degree of similarity with GOOGLE's original rank, using the following weighting scheme:

$$match(user_i, doc_j) = \alpha \cdot sim(user_i, doc_j) + (1 - \alpha) \cdot googlerank(doc_j) \quad (6.2)$$

where α gets values between 0 and 1. When α is 0, conceptual rank is not given any weight, and the $match$ is equivalent to the original rank assigned by GOOGLE. If α has a value of 1, the search engine ranking is ignored and pure conceptual match is considered. Obviously, the conceptual and search engine-based rankings can be blended in different proportions by varying the value of α.

A thorough evaluation has been done in order to investigate the effectiveness of user profiles built out of queries and snippets. The accuracy of such profiles is analyzed comparing, for user-selected results, GOOGLE's original rank with the conceptual rank based on the profile. The evaluation employed 6 users. Using a profile built from 30 queries, the performance measured in terms of the rank of the user-selected result improves of 33%. A user profile built from snippets of 30 user-selected results showed an improvement of 34% (see [77] for details). Therefore, it is possible to assert that, even though the text a user submits to the search engine is quite short, it is enough to provide more accurate, personalized results.

The ability to recognize user interests in a completely non-invasive way, without installing software or using proxy servers, and the accuracy obtained from the personalized results, are some of the major advantages of this approach. Moreover, result-ordering does not exclusively depend on a global relevance measure, where the computed rank for the whole population is deemed relevant for each individual, but it is tailored to a personal relevance where the rank is computed according with each user within the context of their interactions.

Liu and Yu [51] take a similar approach to personalization, where user profiles are built by analyzing the search history, both queries and selected result documents, comparing them to the first 3 levels of the OPEN DIRECTORY PROJECT category hierarchy. Basically, for each query, the most appropriate categories are deduced and used along with the query as current query context. Because queries are short, they are often ambiguous, so they are likely to match multiple categories in the ODP. The system can automatically use the top-matching category for query expansion, or the user can reduce the ambiguity, by explicitly choosing one of the three top-ranked categories provided by the categorization algorithm.

Each category in a user profile is represented with a weighted term vector, where a highly-weighted keyword indicates a high degree of association between that keyword and the category. The system updates the user profile after a query, when the user clicks on a document, and there is a reasonable duration before the next click, or the user decides to save or print it.

Koutrika and Ioannidis [45] proposed an online approach where user needs are represented by a combination of terms connected through logical operators, e.g., conjunction, disjunction, negation, substitution. These operators are used to transform the queries in personalized versions to be submitted to the search engines. The content of the documents for which the user has performed explicit feedback is used to build the user profile. An evaluation shows that when this personalization approach is applied, the users satisfy their needs faster compared with a traditional search engine, improving the number of relevant documents found among the top results.

Quickstep system [56] follows a quasi-online approach and shares some features with the previous systems. A proxy server monitors browsed research papers and a nearest neighbor classifier assigns OPD categories to them overnight. Sets of recommendations based on the correlations between the user profile and research paper topics are drawn on a daily basis. The user can provide feedback in the form of new training examples or adjustments in the classification outcomes. The user profile consists of a set of topics and the related items, computed following the number of browsed research papers about the given topic, while the Vector Space Model is employed to represent the documents.

6.4.2 Offline Approaches

An innovative personalized search algorithm is the *CubeSVD* algorithm, introduced by Sun *et al.* [80] based on the *click-through data* analysis [38]. This technique is suitable for the typical scenario of Web searching, where the user submits a query to the search engine, the search engine returns a ranked list of the retrieved Web pages, and finally the user clicks on pages of interest. After a period of usage, the system will have recorded useful click-through data represented as triples:

$$\langle user,\ query,\ visited\,page \rangle$$

that could be assumed to reflect users' interests. The proposed algorithm aims to model the users' information needs by exploiting such data. It addresses two typical challenges of Web search. The first concerns the study of the complex relationship between user, the query, and the visited Web pages: given a user and her/his query, how to recommend the right Web page to visit? The authors propose a framework for capturing the latent associations among the aforesaid objects.

The second challenge faces the problem of data sparseness: a user generally submits a small number of queries compared with the size of the query set submitted by all the users, and visits few pages. In this case, recognizing relationships among the data becomes a hard task to carry out.

The authors develop a unified framework to model a click-through element as a 3-order tensor, that is, a higher order generalization of a vector (first order tensor) and a matrix (second order tensor), on which 3-mode analysis is performed using the *Singular Value Decomposition* (SVD) technique [33], generalized to HOSVD, Higher-Order SVD [48]. The tensor element measures the preference of a $\langle user, query \rangle$ pair on a given Web page.

Indeed, the CubeSVD algorithm takes the click-though data set as input and outputs a *reconstructed* tensor $\hat{A}$. The tensor measures the degree of relationship among users, queries, and Web pages. The output is represented by a quadruple of the type:

$$\langle user,\ query,\ visited\,page,\ w \rangle$$

representing w the probability that the $user$, after having submitted a given $query$, would be interested in visiting a particular $page$. In this way, relevant Web pages can be recommended to the user by the system. Users are not consulted on the relevance of

the visited Web pages during the search process, and the system records and analyzes their clicks as in other implicit feedback based approaches.

An evaluation on a 44.7 million record click-through data set showed that CubeSVD, thanks to high order associations identified by the algorithm, achieves better accuracy compared with collaborative filtering and LSI-based approaches [20]. Although the whole computation is remarkably time-consuming, it is part of an offline process that does not affect the runtime activity. Nevertheless, the algorithm has to be periodically run in order to take into consideration new click-through data.

Further offline approaches exploiting data mining techniques are discussed in Chapter 3 of this book [57].

6.5 Personalization Based on Rich Representations of User Needs

This section presents three prototypes of personalized search systems based on complex representations of user needs constructed using explicit feedback: *ifWeb*, *Wifs* and *InfoWeb*. They are mostly based on frames and semantic networks, two AI structures developed in order to represent concepts in a given domain, and the related relationships between them. Even though these prototypes share some features, the mechanisms employed to build the profiles and the way the needs are represented are fairly different. Therefore, we prefer to discuss them in distinct sections. Complex user modeling techniques applied to the Web personalization are exhaustively discussed in Chapter 2 of this book [29].

6.5.1 ifWeb

ifWeb [6] is a user model-based intelligent agent capable of supporting the user in Web navigation, retrieval, and filtering of documents taking into account specific information needs expressed by the user with keywords, free-text descriptions, and Web document examples. The *ifWeb* system exploits semantic networks in order to create the user profile.

More specifically, the user profile is represented as a weighted semantic network whose nodes correspond to terms (concepts) found in documents and textual descriptions given by the user as positive or negative examples, i.e., relevance feedback. Network's arcs link pairs of terms that co-occurred in some document. The use of the semantic network and of the co-occurrence relationships allows *ifWeb* to overcome the limitations of simple keyword matching, particularly polysemy.

The *ifWeb* prototype also performs autonomous focused crawling (see Chapter 7 [53] for details), collecting and classifying interesting documents. From specific documents pointed out by the user or identified through search engines, the system autonomously performs an extended opportunistic navigation of the Web, then retrieves and classifies documents relevant to the user profile. As a result, the system shows the user the documents that have been classified as the most relevant ones, in decreasing order of probable interest.

The user profile is updated and refined by explicit relevance feedback provided by the user: *ifWeb* presents a collection of documents to the user (usually no more than

ten for each feedback session), who then explicitly selects the ones that meet his needs. Then, *ifWeb* autonomously extracts the information necessary to update the user profile from the documents on which the user expressed some positive feedback. Moreover, the prototype includes a mechanism for temporal decay called *rent*, which lowers the weights associated with concepts in the profile that have not been reinforced by the relevance feedback mechanism for a long period of time. This technique allows the profile to be kept updated so that it always represents the current interests of the user.

6.5.2 Wifs

The *Wifs* system described in [54] is capable of filtering HTML or text documents retrieved by the search engine ALTAVISTA[3] in response to a query input by the user. This system evaluates and reorders page links returned by the search engine, taking into account the user model of the user who typed in the query. The user can provide feedback on the viewed documents, and the system uses that feedback to update the user model accordingly.

In short, the user model consists of a frame whose *slots* contain terms (topics), each one associated with other terms (co-keywords) which form a simple semantic network. Slot terms, that is, the topics, must be selected from those contained in a *Terms Data Base* (TDB), created *a priori* by experts who select the terms deemed most relevant for the pertinent domain. Figure 6.6a illustrates a simplified description of a hypothetical user model.

The filtering system is based on a content-based approach, where the documents retrieved by ALTAVISTA are assessed solely according to their contents. The document modeling is not based on traditional IR techniques, such as the Vector Space Model, due to the high variability of Web information sources.

The abstract representation of the document may be seen as described in Fig. 6.6b, where active terms, or *planets*, $T_1, T_2, ..., T_n$ are the ones contained both in the document and TDB, whereas the *satellite* terms $t_1, t_2, ..., t_m$ are the terms included in the document, but not in the TDB, but which co-occur with T_i's. It is evident that the structure is similar to the user model one, but there are no affinity values between the planets and the satellites. For each of these terms, however, document occurrence is calculated. The occurrence value of a term t appearing in a retrieved document is given by the following formula:

$$Occ(t) = c_1 * freq_{body}(t) + c_2 * freq_{title}(t) \tag{6.3}$$

where $freq_{body}(t)$ is the frequency with which term t appears in the body, while $freq_{title}(t)$ is the frequency with which term t appears in the document title, and c_1 and c_2 are two constants.

For the document evaluation, the $\overrightarrow{Rel}$ vector is built, where the element Rel_i represents a relevant value of term t_i compared to user information needs. The user model, the query, and the TDB are taken into account to draw the relevance.

This calculation is done as follows:

[3] http://www.altavista.com

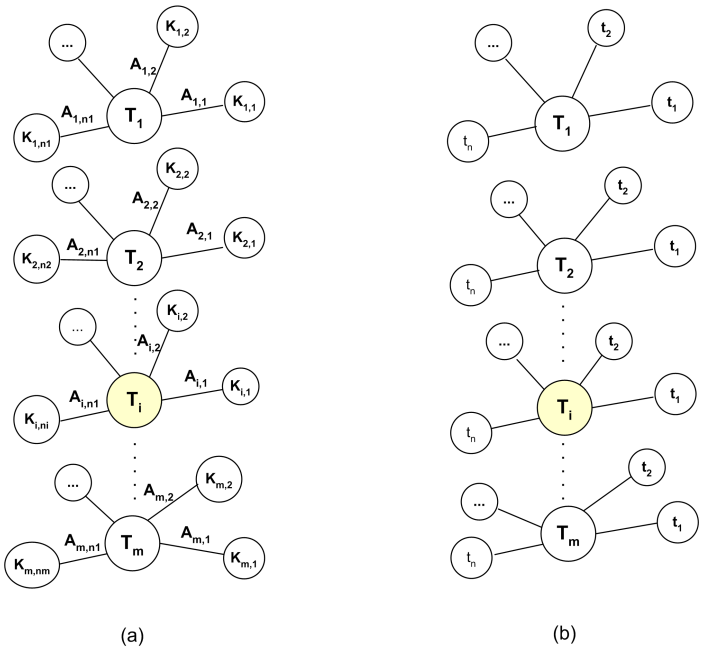

Fig. 6.6. Representations of the User model *(a)* and Document model *(b)*

– *Step 1*. The term t's relevance $Rel_{new}(t)$ (where the term t belongs to the document and user model, as a slot topic) is calculated by intensifying the old relevance value, $Rel_{old}(t)$, through the following formula:

$$Rel_{new}(t) = Rel_{old}(t) + c_3 * \sum_j w_j, \forall w_j : t \in slot_j \tag{6.4}$$

where c_3 is a constant whose value is 2, calibrated experimentally, and w_j the weight associated to slot j containing term t as a topic. In a few words, the new relevance value of term t is obtained from the old value plus the sum of all semantic network weights of the user model containing term t as topic.
– *Step 2*. If the term, as well as belonging to the user model and document, also belongs to the q query input by the user, then the term relevance value is further strengthened, through the following formula:

$$Rel_{new}(t) = Rel_{old}(t) * w_{slot} \tag{6.5}$$

where w_{slot} is the weight associated with topic t. This way, query q, which represents the user's immediate needs, is used to effectively filter the result set to locate documents of interest.
– *Step 3*. If term t belongs to query q, to document d, and to the TDB, but is not included in the user model, then the only contribution to relevance is given by the following formula:

$$Rel_{new}(t) = Rel_{old}(t) + c_3 \tag{6.6}$$

– *Step 4*. If term t is a topic for the $slot_j$, then this step is considers the contributions given by co-keywords. This is where the true semantic network contributes: all the co-keywords K connected to topic t give a contribution, even if previously unknown to the system, i.e., not currently belonging to the user model, nor to the TDB, but only to the document.

$$Rel_{new}(t) = Rel_{old}(t) + w_j * \sum_i A_{j,i} \tag{6.7}$$

$\forall\, co-keyword\ L_i \in slot_j : K_i \in doc\ \forall slot_j : topic_j \in d$

In this stage, the system calculates the final relevance score assigned to the document as follows:

$$Score(Doc) = \overrightarrow{Occ} \cdot \overrightarrow{Rel} = \sum_{\forall t \in Doc} Occ(t) * Rel(t) \tag{6.8}$$

where $\overrightarrow{Occ}$ is the vector consisting of elements Occ_i, and $\overrightarrow{Rel}$ is the vector consisting of elements Rel_i, evaluated in the previous steps.

This system is capable of dynamically updating the user model upon receipt of relevance feedback on the viewed documents provided by the user. In addition, the system uses a *renting* mechanism to decrease the weights of the terms appearing in the model that do not receive positive feedback after a period of time. Further details on the user model updates in *Wifs* are described in Chapter 2 of this book [29].

The *Wifs* system has been evaluated to determine the effectiveness of the user profile in providing personalized reordering of the documents retrieved by ALTAVISTA. Considering the whole set of documents retrieved by the search engine following the query, three relevance sorting structures are taken into account based on results provided by ALTAVISTA, *Wifs* and the user. The metrics defined in [39, 90] have been employed, in the perfect ranking hypothesis, to measure the gaps between *user-AltaVista* and *user-system* sorting. By means of a non parametric test, it was shown that the two distributions are different, with the *user-system* variable giving lower values, which shows that the alternative hypothesis is real. Hence, the system sorts sets of documents in a more relevant way for user needs. The evaluation considered 15 working sessions (where for each session a query was submitted) and 24 users. The ordering of the first 30 results was considered. It shows that the system provides roughly a 34% improvement when compared to the search engine's non-personalized results (see [54] for details).

Another interesting experiment showed that the system is capable of responding quickly to the user's sudden interest changes, through the aforementioned dynamic update mechanism, activated by relevance feedback supplied directly by the user.

6.5.3 InfoWeb

A further approach to personalization is taken by *InfoWeb* [30], an interactive system developed for adaptive content-based retrieval of documents belonging to Web digital

libraries. The distinctive characteristic of *InfoWeb* is its mechanism for the creation and management of a stereotype knowledge base, and its use for user modeling. A *stereotype* [69] contains the vector representation of the most significant document belonging to a specific category of users, initially defined by a domain expert. The system helps the domain expert build the stereotypes through a *k-means* clustering technique [52], which is applied to the whole document collection in an off-line phase. The clustering starts with specific documents as initial seeds, each one acting as a representative centroid for a class of users. *InfoWeb* uses the stereotypes exclusively for the construction of the initial user model. The user's profile evolves over time in accordance to the user's information needs, formulated through queries, using an explicit relevance feedback algorithm that allows the user to provide an assessment of the documents retrieved by the system.

The filtering system extends the traditional one based on the Vector Space model because it also takes into account the co-occurrences of terms in the computation of document relevance and involves user profiles to perform query expansion. The final document evaluation process involves the representation of the documents, of the user model, and of the expanded query. The results of the experiments are promising, both in terms of performance and in the ability to adapt to the user' shifting interests.

The *InfoWeb* prototype is specifically designed for digital libraries with an established document collection and the presence of a domain expert. Nevertheless, some of the proposed techniques, e.g., stereotypes and automatic query expansion, can be also adapted to vast and dynamic environments, such as the Web.

6.6 Collaborative Search Engines

The EUREKSTER[4] search engine includes a proprietary module named *SearchParty* based on collaborative filtering to help users find the best pages related to a given query. EUREKSTER implements social filtering by storing all the results selected by the users for each query submitted to the search engine. Those results will be shared among the community of users interested in the same topics.

In addition to the social filtering module, the EUREKSTER search engine stores all the queries submitted by a user and the resources on which he clicks. If a certain amount of time is spent on a particular resource, when the user re-submits the same query later on, the previously clicked pages are ranked higher in the result list. Thus, the user does not have to wade through a long list of search results again in order to find a previously selected page.

A social adaptive navigation system called *Knowledge Sea* [89] exploits both the traditional IR approach, where documents and queries are represented through the Vector Space Model, and social navigation based on past usage history and user annotations.

Users can search socially, referring to other users' behavior and opinions, by examining the color lightness and exploring icons next to each result, which respectively provide users with information about the popularity of the page and allow the user to view any available annotations. For example, a dark background means that a document

[4] http://www.eurekster.com

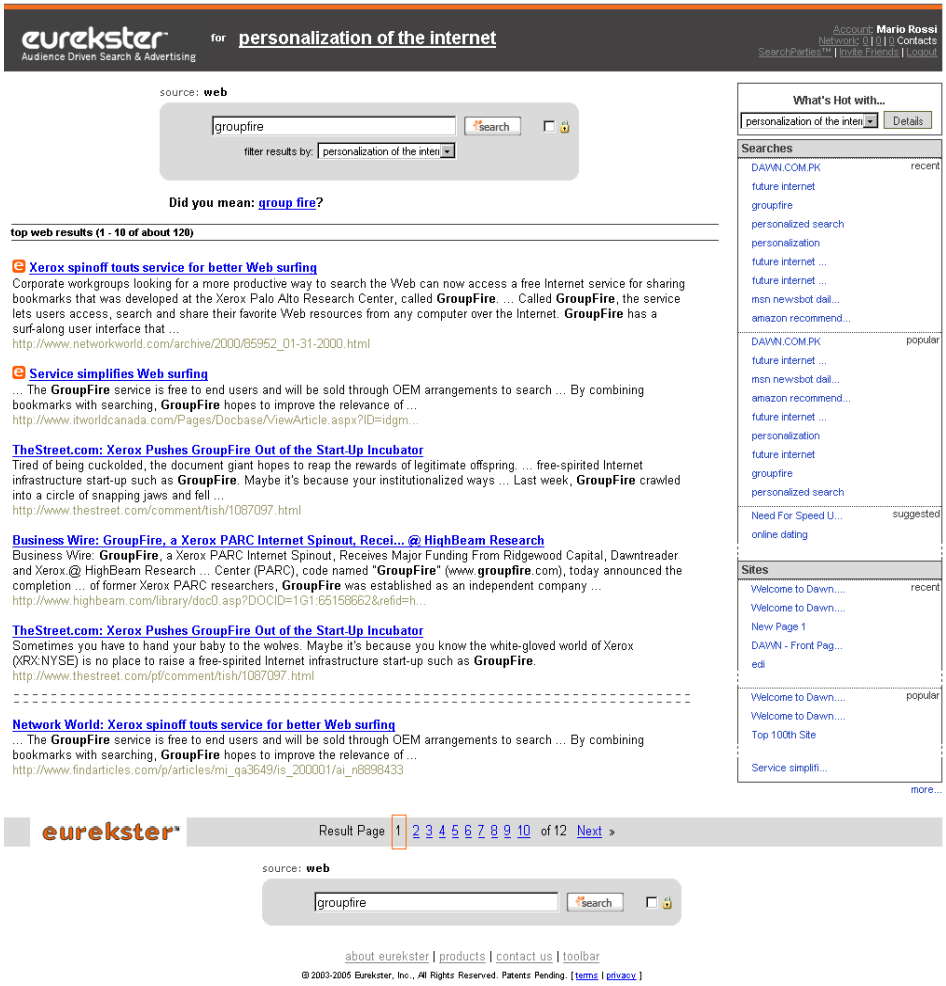

Fig. 6.7. After having selected a particular topic, such as "Personalization of the Internet", EUREKSTER is able to suggest results that other people have previously found useful. In this example, the first two results are proposed by social filtering. *(Reproduced with permission of Eurekster.)*

is popular or it has many annotations, while a light foreground color suggests that the users have chosen to view the document less frequently than most. Even though the results of the search are not socially re-ranked, every result is annotated with social visual cue according with the other users' past searches.

In order for a search engine to employ a collaborative approach, it is important to calculate similarity measures among user needs, which could be identified through queries, and selected documents in result lists. Glance [31] states that the measure of relatedness among two queries should not depend on the actual terms in them, but on the documents returned by the queries. Two queries could be considered synonymous, even

though they contain no terms in common, such as 'handheld devices' and 'mobile computers', by looking at the relationship between the documents returned by each. If the search engine produces many common results for two syntactically different queries, they should be considered semantically correlated. Zhao *et al.* [94] present a framework where the similarity among queries is extended by analyzing the temporal characteristics of the historical click-through, that is, the timestamps of the log data.

In the I-SPY collaborative search engine [76], the queries are considered sets of unique terms on which the Jaccard measure is used to compute the similarity measure:

$$Sim(q_1, q_2) = \frac{|q_1 \cap q_2|}{|q_1 \cup q_2|}$$

Two queries are considered similar if the value computed by the aforesaid formula exceeds a given similarity threshold. For example 'modem adsl' and 'modem usb' are considered to be duplicates above a 0.25 similarity threshold but not above a 0.5 threshold.

Based on the idea that specialized search engines, that is, engines focused on a particular topic, attract communities of users with similar information needs, it is possible to build a statistical model of query-page relevance based on the probability that a page p is selected by a user when returned as a result for a given query q. In practice, I-Spy improves result lists from a traditional general purpose search engine analyzing the interests of communities of users. A community may be identified by a query log of a search box located on specialized Web sites.

This model allows the search engine to personalize search results without relying on content-analysis techniques, but on the relative frequency with which a page has been selected in the past in response to a given query. Results frequently selected by users are promoted ahead of other results returned by a traditional search engines by means of the following relevance:

$$Relevance(p_j, q_i) = \frac{H_{ij}}{\sum_{\forall j} H_{ij}}$$

where H_{ij} indicates the number of users that have selected a page p_j given the query q_i so far. The H matrix represents the statistical model of query-page relevance built with data extracted from a specialized search engine, therefore different matrices are used for different communities of users. For a closer examination of further group recommendation approaches see Chapter 20 of this book [36].

Compass Filter [46] follows a similar collaborative approach, but it is based on Web communities, that is, sets of Web documents that are highly inter-connected. A pre-processing step identifies these communities analyzing the Web hyperlink structure, similarly to the HITS algorithm [42]. If the user has frequently visited documents in a particular community *X*, when he submits a query about *X*, all the results that fall into the same community are boosted by the collaborative service. Instead of performing a re-ranking process, a different approach uses Web communities in order to find contextualization cues to be combined with the queries [4]. Claypool *at al.* [16] explores a possible combination of collaborative and content-based approaches by basing the interest prediction of a document on a weighted average adapted to the individual user. An evaluation has shown good results in the on-line newspaper domain.

6.7 Adaptive Result Clustering

Traditional search engines show the query results in long lists ranked by the similarity between query and page content. Users usually sift through the list sequentially, examining the titles and the textual snippets extracted from the pages, in order to find the information matching their needs. Obviously, this activity might take a long time, especially if the user is not able to clearly formulate and submit to the search engine a textual representation of what he is looking for.

Several Web search engines organize results into folders by grouping pages about the same topics together, for example CLUSTY[5] and KARTOO[6]. The former is based on the VIVÍSIMO[7] clustering engine that arranges results in the style of folders and subfolders. In addition to the traditional HTML layout, the meta search engine KARTOO organizes the returned resources on a graphic interactive map. When the user moves the pointer over those resources, a brief description of the site appears. The size of the icons corresponds to the relevance of the site to the given query. As previously noticed, search systems that arrange results into clusters can be considered personalized because the users are able to customize the results by navigating and choosing selected clusters based on their needs.

In the Web domain, clustering is usually performed after the retrieval of the query results, therefore the whole process must be fast enough to be computed interactively, while the user waits for results. For this reason, the clustering algorithms usually take document snippets instead of whole documents as a representation of page contents. Since, unlike classification, clustering does not require pre-defined categories, the number and the organization of the clusters should be chosen so that the user can navigate easily through them. Finally, clustering should provide concise and accurate cluster descriptions that allow the user to find the most useful ones, even in case of polysemous or misleading queries. For a brief overview of clustering techniques, see Chapter 5 [55].

Further clustering systems are described in the literature, e.g., [91, 93]. The SnakeT meta-search engine [25] includes an innovative hierarchical clustering algorithm with reduced time complexity. It allows the users to select a subset of the clusters that are more likely to satisfy their needs. Then, the system performs a query refinement, building and submitting a new query that incorporates keywords extracted by the system from the selected clusters.

Scatter/Gather [19] uses a similar approach, where the user is able to select one or more clusters for further analysis. The system gathers together all the selected groups and applies the clustering again, scattering the Web sites into a small number of clusters, which are again presented to the user. After a sequence of iterations, the clusters become small enough and the resources are shown to the user.

[5] http://www.clusty.com/
[6] http://www.kartoo.com/
[7] http://www.vivisimo.com/

6.8 Hyperlink-Based Personalization

Based on one of the enhanced versions of the PageRank algorithm [37], Chirita *et al.* [15] proposed a personal ranking platform called *PROS* that provides personalized ranking of Web pages according to user profiles built automatically, using user bookmarks or frequently-visited page sets.

In short, the PageRank (PR) is a vote assigned to a page A collected from all the pages $T_1..T_n$ on the Web that point to it. It represents the importance of the page pointed to, where a link to a page counts as a vote of support. The PageRank of a page A is given as follows:

$$PR(A) = (1-d) + d\left[\frac{PR(T_1)}{C(T_1)} + ... + \frac{PR(T_n)}{C(T_n)}\right]$$

where the parameter d is the damping factor that can be set between 0 and 1 and $C(T_n)$ is defined as the number of links in the page T_n. The PR scores provide *a priori* importance estimates for all of the pages on the Web, independent of the search query. At query time, these importance scores are combined with traditional IR scores to rank the query results.

Briefly, in PROS, the pages judged more interesting for the user are given to the *HubFinder* module that collects hub pages related to the user topics, that is, pages that contain many links to high-quality resources. That module analyzes just the link structure of the Web, running a customized version of Kleinberg's HITS algorithm [42]. A further algorithm, called *HubRank*, combines the PR value with the hub value of Web pages in order to further extend the result set of HubFinder. The final page set is given to the personalized version of PageRank [37] that re-ranks the result pages each time the user submits a query.

The two algorithms, HubFinder and HubRank use the Web link structure to find topic-related pages and to rank the Web pages needed to build the user profile for the Personalized PageRank algorithm. The pages judged more interesting are collected and the expanded sets are built automatically, using bookmarks and the most visited pages. The process does not require explicit activity by the user.

In order to enable "topic sensitive" Web searches, in [34], the importance for each page is calculated by tailoring the PageRank scores for each topic. Thus, pages considered important in some subject domains may not be considered important in others. For this reason, the algorithm computes 16 topic-sensitive PageRank sets of values, each based on URLs from the top-level categories of the OPEN DIRECTORY PROJECT. Each time a query is submitted, it is matched to each of these topics and, instead of using a single global PageRank value, a linear combination of the topic-sensitive ranks are drawn, weighted using the similarities of the query to the topics. Since all the link-based computation are performed off-line, the time spent for the process is comparable to the original PageRank algorithm. Experiments led on this system concluded that the use of topic-specific PageRank scores can improve Web search accuracy.

Qui and Cho [63] extends the Topic-Sensitive PageRank computing multiple ranks, one for each OPD topic. When a query is submitted, the most suitable rank is selected (that is, the rank of the topic that most closely matches the given query) and used for ranking. A personalized version of PageRank based on DNS domains is proposed in

[2], while a personalized system named *Persona* based on the ODP taxonomy, and on an improved version of the HITS algorithm [14] that incorporates user feedback is discussed in [81].

6.9 Combined Approaches to Personalization

Some prototypes provide personalized search combining more than one adaptive approach. For example Outride uses both the browsing history and the current context in order to perform personalization, in the form of query modification and result reranking. A second system, named *infoFACTORY*, uses an alerting approach trained according to the categories explicitly selected by the user.

6.9.1 Outride

Outride Inc., an information retrieval technology company acquired by GOOGLE in 2001, introduced a contextual computing system for the personalization of search engine results [62]. *Contextualization* and *individualization* are the two different computational techniques used to perform the personalization. The former is related to the "interrelated conditions that occur within an activity", e.g., the kind of information available, the applications in use, and the documents currently examined, while individualization refers to the "characteristics that distinguishes an individual", such as his goals, knowledge and behaviors assumed during the search.

Adomavicius and Tuzhilin [1] stress this division from the user profiling point of view, identifying two components: *behavioral* and *factual*. The latter corresponds to the output of the above-mentioned individualization process. In contrast, the behavioral component contains information about the on-line activities of the user. For instance, a common representation is based on association rules, where interesting associations or correlation relationships among large set of usage data are extracted, e.g., when shopping on Friday, user X usually spends more than $20 on DVDs. Further details can be found in Chapter 3 of this book [57].

Outride's user model includes both the contextualization and individualization technique, aiming at determining a measure of importance that differs from the traditional relevance measures based, for example, on citation and hyperlink approaches. Those measures are characterized by values that affect the results for the entire user population, without taking into account any contextual or individual information on the user, the change in the user's interests and knowledge over time, or the documents he deems relevant.

In practice, the Outride client is integrated into the sidebar of the Internet Explorer browser. Its user model is based on the hierarchical taxonomy of the OPEN DIRECTORY PROJECT, where a subset of categories are weighted according to the current user needs. These weights are initially set by looking through links suggested by the user, and they are kept updated each time the user clicks on a document, while a surfing history stores the last 1,000 selected links. Therefore, both the explicit and implicit feedback have been utilized.

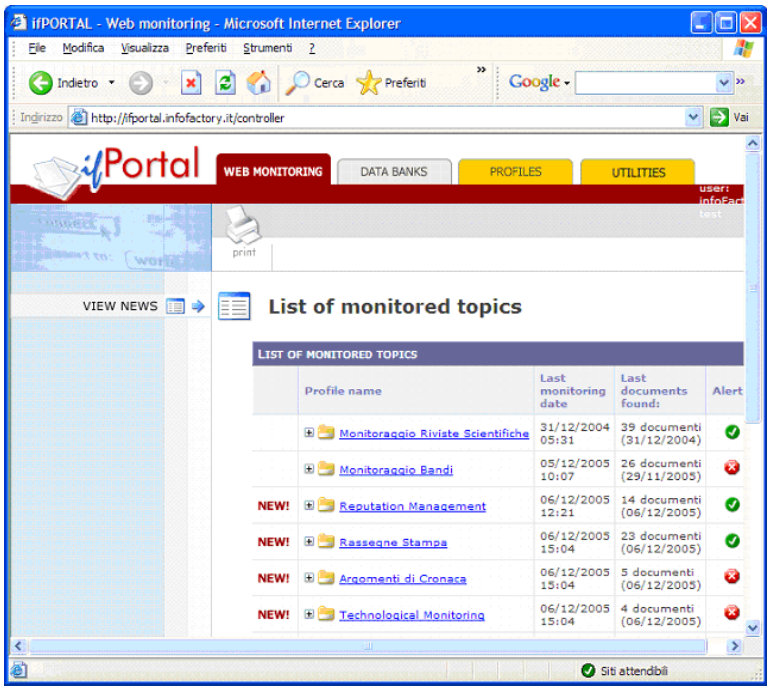

Fig. 6.8. The main page of the *infoFACTORY* monitoring service. From this page, the user can access documents classified according to custom, user-defined categories (see the folder icons). New recently discovered updates, are labeled with a *NEW!* icon. Users may have several profiles, one per topic. A round green icon with a check mark inside indicates that the notification service is enabled for that profile.

The user model is used both for query augmentation and result re-ranking. In the first case, information from the selected categories from the ODP and the Web document currently viewed are compared to the query. If they are similar, the submitted query is related to topics the user has previously seen, an the system can improve it with similar terms in order to disambiguate the query and suggest synonyms. The results from a search engine are re-ranked according to the content of the user model and the current user context, extracting textual information from the pages, e.g., titles and other metadata, and comparing them with a VSM-based representation of the profile. An evaluation of the time spent completing a given set of tasks shows that both novice and expert users are able to find information more quickly using the Outride client than using traditional tools.

6.9.2 infoFACTORY

Finally, it is worth mentioning that *infoFACTORY* [82] contains a large set of integrated Web tools and services that are able to evaluate and classify documents retrieved following a user profile. This system suggests new, potentially interesting contents as soon

as it is published on the Web. Thus, it is an application of personalized information provided by means of *push* technology, instead of the traditional *pull* technology employed by search engines.

The *infoMONITOR* component of the system automatically and periodically monitors a selected set of Web resources in order to discover and notify the user about new and interesting documents. Examples of monitored resources include Web sites, portals devoted to a particular topic, daily news sites, journals and magazines, UseNet news, and search engines. Users are able to define their own custom categories, each one represented by a topic-specific profile. Documents are collected and classified into these user-defined categories, which are then used to display the new information, as shown in Fig. 6.8. The user can customize the notification service, requesting e-mail and/or SMS alerting.

6.10 Conclusions

Personalized search on the Web is a research field that has been recently gaining interest, since it is a possible solution to the information overload problem. The reason is quite simple: information plays a crucial role for each user, and users are constantly challenged to take charge of the information they need to achieve both their personal and professional goals. The ability to filter and create a personalized collection of resources simplifies the whole searching process, increasing search engine accuracy and reducing the time the user has to spend to sift through the results for a given query. The same personalization techniques could also be employed to provide advertisements tailored to the current user activity or to proactively collect information on behalf of a user. This chapter provides an introduction to that field, focusing on some of the most interesting and promising approaches and techniques. Some of these researches have been employed in real information systems, while others remain under exploration in research labs. The novelty and liveliness of the personalization field suggests that, over the next few years, new and interesting algorithms and approaches will be proposed and probably transferred to the information systems with which users interact in every day use, such as, search engines or desktop search tools. Ontologies and the Semantic Web[8] are two important research fields that are beginning to receive attention in this context. Gauch *et al.* [28] are investigating techniques that build ontology-based user profiles without user interaction, automatically monitoring the user's browsing habits. Dolog *et al.* [22] are studying mechanisms based on logical mapping rules and description logics, which allow metadata and ontology concepts to be mapped to concepts stored in user profiles. This logical characterization formally enables the personalization techniques in a common language, such as *FOL*, and the reasoning over the Semantic Web (for a closer examination see Chapter 23 [23]).

If the user is working to achieve specific goals, successful systems should recognize those goals and predict aspects of their future behavior. Since the system has expectations about the next user actions, if it is flexible it can adapt itself to the users, thus it should be possible to considerably speed up human-computer interaction.

[8] http://www.w3.org/2001/sw/

The *plan-recognition* techniques applied during personalization usually attempt to recognize patterns in user behavior, finding in the set of past actions the ones that are likely to be taken next. For example, some statistical models based on random variables make assumptions about unknown parameters, extrapolating them from observed sample results. These parameters could represent aspects of a user's future behavior, such as their goals, allowing the system to predict their forthcoming actions [95].

Language semantic analysis to understand the meaning of Web content and - more importantly - how it relates to a user's query is a further important field of research in the personalization domain. Language Modeling and Question Answering are two important Natural Language Processing (NLP) research areas that could lead to breakthroughs in the development of personalized search systems. New search engines based on these technologies may be able to understand the users' intention through the analysis of user-supplied natural language questions. They may be able to better understand keywords in the queries by recognizing various sentence types, analyze syntax, and disambiguate word senses in context. As a result, search results will be more accurate, satisfactory, and reliable.

References

1. Adomavicius, G., Tuzhilin, A.: User profiling in personalization applications through rule discovery and validation. In: KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM Press (1999) 377–381
2. Aktas, M.S., Nacar, M.A., Menczer, F.: Personalizing pagerank based on domain profiles. In Mobasher, B., Liu, B., b.Masand, Nasraoui, O., eds.: Proceedings of the sixt WEBKDD workshop Web Mining and Web Usage Analysis (WEBKDD'04), Seattle, Washington, USA (2004) 83–90
3. Allan, J.: Incremental relevance feedback for information filtering. In: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Zurich, Switzerland (1996) 270–278
4. Almeida, R.B., Almeida, V.A.F.: A community-aware search engine. In: WWW '04: Proceedings of the 13th international conference on World Wide Web, New York, NY, USA, ACM Press (2004) 413–421
5. Anick, P.: Using terminological feedback for web search refinement: a log-based study. In: SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, New York, NY, USA, ACM Press (2003) 88–95
6. Asnicar, F.A., Tasso, C.: ifWeb: a prototype of user model-based intelligent agent for document filtering and navigation in the world wide web. In: Proceedings of Workshop Adaptive Systems and User Modeling on the World Wide Web (UM97), Sardinia, Italy (1997) 3–12
7. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley (1999)
8. Bharat, K., Kamba, T., Albers, M.: Personalized, interactive news on the web. Multimedia Syst. **6**(5) (1998) 349–358
9. Budzik, J., Hammond, K.J.: User interactions with everyday applications as context for just-in-time information access. In: IUI '00: Proceedings of the 5th international conference on Intelligent user interfaces, New York, NY, USA, ACM Press (2000) 44–51
10. Budzik, J., Hammond, K.J., Birnbaum, L.: Information access in context. Knowledge-Based Systems **14**(1-2) (2001) 37–53

11. Bunt, A., Carenini, G., Conati, C.: Adaptive content presentation for the web. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web: Methods and Strategies of Web Personalization. Volume 4321 of Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
12. Burke, R.: Hybrid web recommender systems. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web: Methods and Strategies of Web Personalization. Volume 4321 of Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
13. Chan, P.K.: Constructing web user profiles: A non-invasive learning approach. In: WEBKDD '99: Revised Papers from the International Workshop on Web Usage Analysis and User Profiling, London, UK, Springer-Verlag (2000) 39–55
14. Chang, H., Cohn, D., McCallum, A.: Learning to create customized authority lists. In: ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (2000) 127–134
15. Chirita, P.A., Olmedilla, D., Nejdl, W.: Pros: A personalized ranking platform for web search. In: 3rd International Conference Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2004). Volume 3137 of Lecture Notes in Computer Science., Eindhoven, The Netherlands, Springer (aug 2004) 34–43
16. Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., Sartin, M.: Combining content-based and collaborative filters in an online newspaper. In: ACM SIGIR Workshop on Recommender Systems - Implementation and Evaluation, ACM Press (1999) `http://www.csee.umbc.edu/~ian/sigir99-rec/`.
17. Claypool, M., Le, P., Wased, M., Brown, D.: Implicit interest indicators. In: IUI '01: Proceedings of the 6th international conference on Intelligent user interfaces, New York, NY, USA, ACM Press (2001) 33–40
18. Collins, A.M., Quillian, R.M.: Retrieval time from semantic memory. Journal of Learning and Verbal Behavior **8** (1969) 240–247
19. Cutting, D.R., Karger, D.R., Pedersen, J.O., Tukey, J.W.: Scatter/gather: a cluster-based approach to browsing large document collections. In: SIGIR '92: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press (1992) 318–329
20. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. Journal of the American Society of Information Science **41**(6) (1990) 391–407
21. Dieberger, A., Dourish, P., Höök, K., Resnick, P., Wexelblat, A.: Social navigation: techniques for building more usable systems. Interactions **7**(6) (2000) 36–45
22. Dolog, P., Henze, N., Nejdl, W., Sintek, M.: Towards the adaptive semantic web. In Bry, F., Henze, N., Maluszynski, J., eds.: Principles and Practice of Semantic Web Reasoning, International Workshop, PPSWR 2003, Mumbai, India, December 8, 2003, Proceedings. Volume 2901 of Lecture Notes in Computer Science., Springer (2003) 51–68
23. Dolog, P., Nejdl, W.: Semantic web technologies for personalized information access on the web. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web: Methods and Strategies of Web Personalization. Volume 4321 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, Heidelberg, and New York (2007) this volume
24. Dumais, S., Cutrell, E., Cadiz, J., Jancke, G., Sarin, R., Robbins, D.C.: Stuff i've seen: a system for personal information retrieval and re-use. In: SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, New York, NY, USA, ACM Press (2003) 72–79
25. Ferragina, P., Gulli, A.: A personalized search engine based on web-snippet hierarchical clustering. In: WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web, New York, NY, USA, ACM Press (2005) 801–810

26. Freyne, J., Smyth, B.: An experiment in social search. In Bra, P.D., Nejdl, W., eds.: Adaptive Hypermedia and Adaptive Web-Based Systems, Third International Conference, AH 2004, Eindhoven, The Netherlands, August 23-26, 2004, Proceedings. Volume 3137 of Lecture Notes in Computer Science., Springer (2004) 95–103
27. Furnas, G.W., Landauer, T.K., Gomez, L.M., Dumais, S.T.: The vocabulary problem in human-system communication. Commun. ACM **30**(11) (1987) 964–971
28. Gauch, S., Chaffee, J., Pretschner, A.: Ontology-based personalized search and browsing. Web Intelligence and Agent System **1**(3-4) (2003) 219–234
29. Gauch, S., Speretta, M., Chandramouli, A., Micarelli, A.: User profiles for personalized information access. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web: Methods and Strategies of Web Personalization. Volume 4321 of Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
30. Gentili, G., Micarelli, A., Sciarrone, F.: Infoweb: An adaptive information filtering system for the cultural heritage domain. Applied Artificial Intelligence **17**(8-9) (2003) 715–744
31. Glance, N.S.: Community search assistant. In: IUI '01: Proceedings of the 6th international conference on Intelligent user interfaces, New York, NY, USA, ACM Press (2001) 91–96
32. Goldberg, D., Nichols, D., Oki, B.M., Terry, D.: Using collaborative filtering to weave an information tapestry. Commun. ACM **35**(12) (1992) 61–70
33. Golub, G.H., Loan, C.F.V.: Matrix computations (3rd ed.). Johns Hopkins University Press, Baltimore, MD, USA (1996)
34. Haveliwala, T.H.: Topic-sensitive pagerank. In: WWW '02: Proceedings of the 11th international conference on World Wide Web, New York, NY, USA, ACM Press (2002) 517–526
35. Höscher, C., Strube, G.: Web search behavior of internet experts and newbies. In: Proceedings of the 9th World Wide Web Conference (WWW9), Amsterdam, Netherlands (2000) 337–346
36. Jameson, A., Smyth, B.: Recommending to groups. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web: Methods and Strategies of Web Personalization. Volume 4321 of Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
37. Jeh, G., Widom, J.: Scaling personalized web search. In: WWW '03: Proceedings of the 12th international conference on World Wide Web, New York, NY, USA, ACM Press (2003) 271–279
38. Joachims, T.: Optimizing search engines using clickthrough data. In: Proceedings of 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press (2002) 133–142
39. John Kemeny, J.L.S.: Mathematical Models in the Social Sciences. MIT Press, New York (1962)
40. Kelly, D., Teevan, J.: Implicit feedback for inferring user preference: a bibliography. SIGIR Forum **37**(2) (2003) 18–28
41. Khopkar, Y., Spink, A., Giles, C.L., Shah, P., Debnath, S.: Search engine personalization: An exploratory study. First Monday **8**(7) (2003) `http://www.firstmonday.org/issues/issue8_7/khopkar/index.html`.
42. Kleinberg, J.: Authoritative sources in a hyperlinked environment. In: Proceedings of the 9th annual ACM-SIAM symposium on Discrete algorithms, San Francisco, CA, USA (1998) 668–677
43. Kobsa, A.: Privacy-enhanced web personalization. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web: Methods and Strategies of Web Personalization. Volume 4321 of Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg New York (2007) this volume

44. Koenemann, J., Belkin, N.J.: A case for interaction: a study of interactive information retrieval behavior and effectiveness. In: CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM Press (1996) 205–212
45. Koutrika, G., Ioannidis, Y.: A unified user profile framework for query disambiguation and personalization. In: Proceedings of the Workshop on New Technologies for Personalized Information Access (PIA2005), Edinburgh, Scotland, UK (2005) 44–53 http://irgroup.cs.uni-magdeburg.de/pia2005/docs/KouIoa05.pdf.
46. Kritikopoulos, A., Sideri, M.: The compass filter: Search engine result personalization using web communities. In Mobasher, B., Anand, S.S., eds.: Intelligent Techniques for Web Personalization, IJCAI 2003 Workshop, ITWP 2003, Acapulco, Mexico, August 11, 2003, Revised Selected Papers. Volume 3169 of Lecture Notes in Computer Science., Springer (2003) 229–240
47. Kulyukin, V.A.: Application-embedded retrieval from distributed free-text collections. In: AAAI/IAAI. (1999) 447–452
48. Lathauwer, L.D., Moor, B.D., Vandewalle, J.: A multilinear singular value decomposition. SIAM J. Matrix Anal. Appl. **21**(4) (2000) 1253–1278
49. Lawrence, S.: Context in web search. IEEE Data Eng. Bull. **23**(3) (2000) 25–32
50. Lawrence, S., Giles, C.L.: Context and page analysis for improved web search. IEEE Internet Computing **2**(4) (1998) 38–46
51. Liu, F., Yu, C., Meng, W.: Personalized web search for improving retrieval effectiveness. IEEE Transactions on Knowledge and Data Engineering **16**(1) (2004) 28–40
52. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability. Volume 1., University of California Press (1967) 281–297
53. Micarelli, A., Gasparetti, F.: Adaptive focused crawling. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web: Methods and Strategies of Web Personalization. Volume 4321 of Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
54. Micarelli, A., Sciarrone, F.: Anatomy and empirical evaluation of an adaptive web-based information filtering system. User Modeling and User-Adapted Interaction **14**(2-3) (2004) 159–200
55. Micarelli, A., Sciarrone, F., Marinilli, M.: Web document modeling. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web: Methods and Strategies of Web Personalization. Volume 4321 of Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
56. Middleton, S.E., Roure, D.C.D., Shadbolt, N.R.: Capturing knowledge of user preferences: ontologies in recommender systems. In: K-CAP '01: Proceedings of the 1st international conference on Knowledge capture, New York, NY, USA, ACM Press (2001) 100–107
57. Mobasher, B.: Data mining for web personalization. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web: Methods and Strategies of Web Personalization. Volume 4321 of Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
58. Montaner, M., Lopez, B., Rosa, J.L.D.L.: A taxonomy of recommender agents on the internet. Artificial Intelligence Review **19** (2003) 285–330
59. Oard, D.W.: The state of the art in text filtering. User Modeling and User-Adapted Interaction **7**(3) (1997) 141–178
60. Olston, C., Chi, E.H.: ScentTrails: Integrating browsing and searching on the web. ACM Transactions on Computer-Human Interaction **10**(3) (2003) 177–197
61. Pirolli, P.L.T., Pitkow, J.E.: Distributions of surfers' paths through the world wide web: Empirical characterizations. World Wide Web **2**(1-2) (1999) 29–45

62. Pitkow, J., Schütze, H., Cass, T., Cooley, R., Turnbull, D., Edmonds, A., Adar, E., Breuel, T.: Personalized search. Commun. ACM **45**(9) (2002) 50–55
63. Qiu, F., Cho, J.: Automatic identification of user interest for personalized search. In: WWW '06: Proceedings of the 15th international conference on World Wide Web, New York, NY, USA, ACM Press (2006) 727–736
64. Quillian, R.M.: Semantic memory. In Minsky, M., ed.: Semantic information processing. The MIT Press, Cambridge, MA, USA (1968) 216–270 `http://citeseer.ist.psu.edu/ambrosini97hybrid.html`.
65. Raghavan, V.V., Sever, H.: On the reuse of past optimal queries. In: Research and Development in Information Retrieval. (1995) 344–350 `http://citeseer.ist.psu.edu/raghavan95reuse.html`.
66. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: an open architecture for collaborative filtering of netnews. In: CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work, New York, NY, USA, ACM Press (1994) 175–186
67. Resnick, P., Varian, H.R.: Recommender systems. Commun. ACM **40**(3) (1997) 56–58
68. Rhodes, B.J.: Just-In-Time Information Retrieval. PhD thesis, MIT Media Laboratory, Cambridge, MA (May 2000) `http://citeseer.ist.psu.edu/rhodes00justtime.html`.
69. Rich, E.: User modeling via stereotypes. In: Readings in intelligent user interfaces. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1998) 329–342
70. Rijsbergen, C.J.V.: Information Retrieval. Butterworth-Heinemann, Newton, MA, USA (1979)
71. Robertson, S.E.: Theories and models in information retrieval. Journal of Documentation **33**(2) (1977) 126–148
72. Salton, G., McGill, M.: An Introduction to modern information retrieval. Mc-Graw-Hill, New York, NY (1983)
73. Salton, G., Wong, A., Yang, C.: A vector space model for automatic indexing. Commun. ACM **18**(11) (1975) 613–620
74. Savoy, J., Picard, J.: Retrieval effectiveness on the web. Information Processing & Management **37**(4) (2001) 543–569
75. Schafer, J.B., Frankowski, D., Herlocker, J.L., Sen, S.: Collaborative filtering recommender systems. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web: Methods and Strategies of Web Personalization. Volume 4321 of Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
76. Smyth, B., Balfe, E., Freyne, J., Briggs, P., Coyle, M., Boydell, O.: Exploiting query repetition and regularity in an adaptive community-based web search engine. User Modeling and User-Adapted Interaction **14**(5) (2005) 383–423
77. Speretta, M., Gauch, S.: Personalized search based on user search histories. In: Web Intelligence (WI2005), France, IEEE Computer Society (2005) 622–628 `http://dx.doi.org/10.1109/WI.2005.114`.
78. Spink, A., Jansen, B.J.: A study of web search trends. Webology **1**(2) (2004) 4 `http://www.webology.ir/2004/v1n2/a4.html`.
79. Spink, A., Jansen, B.J., Ozmultu, H.C.: Use of query reformulation and relevance feedback by excite users. Internet Research: Electronic Networking Applications and Policy **10**(4) (2000) 317–328 `http://citeseer.ist.psu.edu/spink00use.html`.
80. Sun, J.T., Zeng, H.J., Liu, H., Lu, Y., Chen, Z.: Cubesvd: a novel approach to personalized web search. In: WWW '05: Proceedings of the 14th international conference on World Wide Web, New York, NY, USA, ACM Press (2005) 382–390

81. Tanudjaja, F., Mui, L.: Persona: A contextualized and personalized web search. In: HICSS '02: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 3, Washington, DC, USA, IEEE Computer Society (2002) 67
82. Tasso, C., Omero, P.: La Personalizzazione dei contenuti Web: e-commerce, i-access, e-government. Franco Angeli (2002)
83. Teevan, J.: Seesaw: Personalized web search. Student Workshop for Information Retrieval and Language (SWIRL '04)
(November 2004) http://ciir.cs.umass.edu/~hema/swirl/swirl.htm.
84. Teevan, J., Alvarado, C., Ackerman, M.S., Karger, D.R.: The perfect search engine is not enough: a study of orienteering behavior in directed search. In: CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM Press (2004) 415–422
85. Teevan, J., Dumais, S.T., Horvitz, E.: Personalizing search via automated analysis of interests and activities. In: SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press (2005) 449–456
86. Wærn, A.: User involvement in automatic filtering: An experimental study. User Modeling and User-Adapted Interaction **14**(2-3) (2004) 201–237
87. Webb, G.I., Pazzani, M., Billsus, D.: Machine learning for user modeling. User Modeling and User-Adapted Interaction **11**(1-2) (2001) 19–29
88. White, R., Jose, J.M., Ruthven, I.: Comparing explicit and implicit feedback techniques for web retrieval: Trec-10 interactive track report. In: TREC. (2001) http://trec.nist.gov/pubs/trec10/papers/glasgow.pdf.
89. wook Ahn, J., Brusilovsky, P., Farzan, R.: Investigating users' needs and behavior for social search. In: Proc. of Workshop on New Technologies for Personalized Information Access at 10th International User Modeling Conference, UM 2005. (2005) 1–12 http://irgroup.cs.uni-magdeburg.de/pia2005/docs/AhnBruFar05.pdf.
90. Yao, Y.: Measuring retrieval effectiveness based on user preference of documents. Journal of the American Society for Information Science **46**(2) (1995) 133–145
91. Zamir, O., Etzioni, O.: Web document clustering: a feasibility demonstration. In: SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press (1998) 46–54
92. Zamir, O.E., Korn, J.L., Fikes, A.B., Lawrence, S.R.: Us patent application #0050240580: Personalization of placed content ordering in search results (July 2004)
93. Zeng, H.J., He, Q.C., Chen, Z., Ma, W.Y., Ma, J.: Learning to cluster web search results. In: SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press (2004) 210–217
94. Zhao, Q., Hoi, S.C.H., Liu, T.Y., Bhowmick, S.S., Lyu, M.R., Ma, W.Y.: Time-dependent semantic similarity measure of queries using historical click-through data. In: WWW '06: Proceedings of the 15th international conference on World Wide Web, New York, NY, USA, ACM Press (2006) 543–552
95. Zukerman, I., Albrecht, D.W.: Predictive statistical models for user modeling. User Modeling and User-Adapted Interaction **11**(1-2) (2001) 5–18