Intelligent Search on the Internet

Alessandro Micarelli[1], Fabio Gasparetti[1], and Claudio Biancalana[1]

Department of Computer Science and Automation
Artificial Intelligence Laboratory
University of "Roma Tre"
Via della Vasca Navale, 79 00146 Rome, Italy
{micarel, gaspare, claudio.biancalana}@dia.uniroma3.it

The Web has grown from a simple hypertext system for research labs to an ubiquitous information system including virtually all human knowledge, e.g., movies, images, music, documents, etc. The traditional browsing activity seems to be often inadequate to locate information satisfying the user needs. Even search engines, based on the Information Retrieval approach, with their huge indexes show many drawbacks, which force users to sift through long lists of results or reformulate queries several times.

Recently, an important research activity effort has been focusing on this vast amount of machine-accessible knowledge and on how it can be exploited in order to match the user needs. The personalization and adaptation of the human-computer interaction in information seeking by means of machine learning techniques and in AI-based representations of the information help users to address the overload problem. This chapter illustrates the most important approaches proposed to personalize the access to information, in terms of gathering resources related to given topics of interest and ranking them as a function of the current user needs and activities, as well as examples of prototypes and Web systems.

1 Introduction

The decentralized and unregulated work of millions of authors spread around the world have allowed the Web to grow constantly since its creation. At the beginning of 2005, the part of the Web considered as potentially indexable by major engines was estimated to consist of at least 11.5 billion pages [23]. As the number of Internet users and the number of accessible Web pages grow, it is becoming increasingly difficult for users to find documents that are relevant to their current information needs. Therefore, it is understandable that the research field devoted to applying AI techniques to the problem of discovering and analyzing Web resources has been drawing much attention over the last few years [29].

The *browsing* interaction paradigm, where users analyze Web pages one at a time is a useful approach for reading and comprehending the content of a hypertext, but it is not suitable for locating a specific piece of information. A few years ago, a second interaction paradigm was introduced: *querying* a search engine. Directly retrieving documents from an index of millions of documents

in a fraction of a second is the paramount advantage of this approach, which is based on the classic Information Retrieval (IR) model. Nevertheless, despite having different goals and needs, two users who submit the same query obtain the same result list. Moreover, by analyzing search behavior, it is possible to see that many users are not able to accurately express their needs in exact query terms.

Search engines are the first search approach of users [49]. Personalized search aims at building systems which try to serve up individualized pages to the user; such systems are based on some form of model of the needs and context of the user's activities. For instance, some search engines employ the collaborative or community-based approach in order to suggest pages that other users, who submitted the same query, selected frequently [50].

Because of the vastness of the personalized search domain, in this chapter we focus on some of the most recent algorithms, techniques and approaches strongly related to the AI field. For instance, in order to model the user needs and to assign internal representations to Web documents, approaches based on knowledge representation and machine learning, such as semantic networks and frames [3, 33, 36], neural networks [2, 8, 22] and decision trees [27], have been successfully developed obtaining interesting results. Ontology-based search and autonomous crawling based on AI algorithms are two further interesting research activities addressing personalized search.

The rest of this chapter is organized as follows: Section 2 presents some AI-based approaches recently developed in order to model user needs and Web documents for the personalization task. Section 3 discusses some further emerging trends in personalized search. Finally, Section 4 closes this chapter.

2 User and Document Modeling for Personalized Search

The available information on the Web is usually represented through HTML documents, or more complex formats, such as images, audio, video, Flash animation, etc.. These layout-based representations organize and adapt the content to the reading user by means of the Web browser. For this reason, personalized Web information systems need to pre-process those formats in order to extract the real document content, ignoring the layout-related information, e.g., HTML table tags or fonts. Once the original document content is extracted, ad-hoc representations are usually employed to organize the information so that the personalization process is able to retrieve and filter it according to the user needs.

User knowledge and preference acquisition is another important problem to be addressed in order to provide effective personalized assistance. Some approaches exploit *user data*, that is, information about personal characteristics of the user (such as age, sex, education, country where he lives) in order to learn the knowledge needed to provide effective assistance. Other approaches analyze the *usage data* related to the user's behavior while interacting with the system.

User data are usually collected following explicit approaches, where the user constructs the model by describing the information in which he is interested. Nevertheless, 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 identify documents meeting their information needs. Instead of requiring user needs to be explicitly specified by queries and manually updated by 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, e.g., by means of server access logs or query and browsing histories [25].

Several user modeling approaches are based on simple sets of keywords extracted from interesting documents or suggested by users. A weight can be assigned to each keyword representing its importance in the user profile. Basically, these approaches are inspired by Information Retrieval approaches (IR), an area of research that enjoys a long tradition in the modeling and treatment of non-structured textual documents [4]. Because of the popularity in the Web domain, Sect. 2.1 briefly discusses this topic.

Further techniques are based on models and methods from AI Knowledge Representation, such as Semantic Networks, Frames, Latent Semantic Indexing and Bayesian classifiers. These latter kinds of user models will be discussed in the following sections.

2.1 IR-based Modeling

In the Information Retrieval (IR) field, a collection's documents are often represented by a set of *keywords*, which can be directly extracted from the document text or be explicitly indicated in an initial summary drawn up by a specialist. Independently of the extraction method, these very keywords provide a logic view of the document [4].

When collections are particularly bulky, though, even modern processors have to reduce the set of representative words. This can be done through *stemming*, reducing several words to their common grammatical root, or through the identification of *word groups* (which removes adjectives, adverbs and verbs).

One of the problems of IR systems is that of predicting which documents are relevant and which are not; this decision usually depends on the ranking algorithm used, which tries to put the retrieved documents in order, by measuring similarity. The documents on top of this list are the ones considered more relevant.

The classic models consider all documents described by a set of representative keywords, also known as index terms. The *Vector Model* assigns w weights to the index terms; these weights are used to calculate the similarity level between every document stored in the system and the submitted query.

The Vector Model appraises the level of similarity between document d_j and query q, as a correlation between vectors $\boldsymbol{d}$ and $\boldsymbol{q}$. This correlation can be

quantified, for example, as the cosine of the angle between the two vectors:

$$\mathrm{sim}(d_j, q) = \frac{\boldsymbol{d_j} \cdot \boldsymbol{q}}{|\boldsymbol{d_j}||\boldsymbol{q}|} = \frac{\sum_{i=1}^{t} w_{ij} w_{iq}}{\sqrt{\sum_{i=1}^{t} w_{ij}^2} \sqrt{\sum_{j=1}^{t} w_{jq}^2}}$$

Term weights can be calculated in many different ways [28, 47, 4]. The basic idea of the most efficient techniques is linked to the *cluster* concept, suggested by Salton in [9]. In literature it is possible to recognize many prototypes for Web search based on weighted or Boolean feature vector, e.g., [30, 5, 24, 40, 31].

2.2 Latent Semantic Indexing

The complexity of Information Retrieval is to be clearly seen in two main unpleasant problems:

- **synonymity** all the documents containing term B synonym of term A present in the query are lost, therefore relevant information is not retrieved.
- **polysemy** occurs when a term has several different meanings; it causes irrelevant documents to appear in the result lists.

In order to solve such problems, documents are represented through underlying concepts. The concealed structure is not simply a many-to-many mapping between terms and concepts, but it depends from the body. *Latent Semantic Indexing* (LSI) is a technique that tries to get hold of this hidden semantic structure through the spectral analysis of the terms-documents matrix [15]. It has been applied in different scopes, for example the simulation of human cognitive phenomena, such as modeling human word sorting and category judgments, estimating the quality and quantity of knowledge contained in an essay, or how easily it can be learned by individual students.

The vectors representing the documents are projected into a new subspace obtained by the *Singular Value Decomposition* (SVD) of the terms-document matrix A. This sub-dimension space is generated by the eigenvectors of matrix $A^T A$ corresponding to the highest eigenvalues, thus to the most obvious correlations between terms. A necessary step in implementing LSI in a collection of n documents is the formation of the terms-documents matrix $A_{m \times n}$, where m is the number of distinguished terms present in n documents. Each document is thus represented by an m-dimensional column vector. After having calculated the frequency of each word in each document, it is possible to implement *inter-cluster* and *intra-cluster* weighting functions.

Matrix A is therefore broken down through SVD, which somehow allows to obtain the semantic structure of the document collection by means of orthogonal matrices U_r and V_r (containing the single left and right vectors of A, respectively) and the diagonal matrix Σ of the singular values of A. Once matrices U_r,V_r and Σ are obtained, by extracting the first $k < r$ triple singulars, it is possible to approximate the original terms-document A matrix with matrix A_k of rank k:

$$A_k = \Sigma_k^{-1} \cdot U_k^T \cdot A$$

By using this formula it is possible to map any two documents with different origins in the m-dimensional space of all different terms in the same vector of the reduced space; the set of basic vectors of $k-$dimensional space accounts for the set of concepts or for the different meanings the several documents may have, therefore a generic document in $k-$dimensional space can be represented as a linear combination of concepts or equivalently of the basic vectors of the same space. We thus passed from a representation in a space with m dimensions (the dimension is equivalent to the number of terms found in the analysis of all documents) to a compressed form of the same vectors in $k < m$ dimensional space. It should be pointed out that through A_k one can deliberately reconstruct, even though not perfectly, matrix A, since it contains a certain noise level introduced by the randomness with which terms are chosen to tackle certain discourses or topics; this noise is "filtered" by annulling the singular, less significant values.

Besides recommender systems [48, 44], a generalization of the LSI approach has been also applied to the click-through data analysis in order to model the users' information needs by exploiting usage data, such as the submitted query and the visited results [52]. The evaluation shows that thanks to high order associations identified by the algorithm, it achieves better accuracy compared with other approaches although the whole computation is remarkably time-consuming.

2.3 Bayesian Approach

The idea of resorting to Bayesian probabilistic models to represent a document is undoubtedly very interesting. Indeed, the probability theory provides the required tools to better cope with uncertainty, while Bayesian formalism enables to represent the probabilistic relations within a very vast set of relevant variables, which very efficiently encodes the joint distribution of probability, exploiting the conditional independence relations existing between the variables [42]. The Bayesian approach to probability can be extended to the problem of comprehension: this union leads to an extremely powerful theory, which provides a general solution to noise problems, overtraining and good predictions.

In order to implement the Bayesian document-modeling technique, it is necessary to decide what features are mostly useful to represent a document, and thereafter decide how to assess the probability tables associated with the Bayesian network. The problem of document representation has a considerable impact on a learning system's generalization capacity [13, 34]. Usually, a document is a string of characters: it must be converted into a representation that is suitable for learning algorithms and classification. Words have been acknowledged to work well as representation units, whereas their positioning seems to be less relevant [19]. It is now a matter of deciding the features to describe a document.

The multinomial model is usually employed to assign a representation to a document based on a set of words, associated with their frequency. The word order is lost, but their frequency is recorded. A simplified model (Bernoulli model) represents a document as a set of binary features which indicate the presence or absence of a certain set of words forming the vocabulary.

In the multinomial model, in order to calculate a document probability, the probabilities of the words appearing in it are multiplied [18]. We illustrate the procedure that leads to the final representation of a document, highlighting the hypotheses required to simplify a domain's description.

As a first approach, we must define a feature for each position of a word in the document, and its value is the word in that very position. In order to reduce the number of parameters, the probability tables associated with the Bayesian network are considered the same for all features. The independence entailed in this context asserts that the probability of a word in a certain position does not depend on the words present in other positions.

In order to further reduce the number of parameters, we can suppose that the probability of finding a word is independent of its position: this means assuming that features are independent, and identically distributed. As far the document model is concerned, the reference is to the multinomial model that represents a document as a set of words with their frequency number associated to them. In this representation, the word order is lost, but their frequency is recorded.

Going back to Naïve Bayes, we can say that this twofold document representation is a *bag of words*: a document is seen as a set of words. That's how a feature-value representation of the text is done: each word differing from the others corresponds to a feature whose value is the number of times that word appears in the document. In order to avoid having sets of words with an excess cardinality, it is possible to follow a feature selection procedure that aims at reducing the number of words appearing in the vocabulary. The choice of a document representation is affected by the domain being analyzed.

Among the various search agents based on the Bayesian approach it is worth mentioning *Syskill & Webert* [41]. It can make suggestions about interesting pages during a browsing session and use the internal user model to query a search engine in order to retrieve additional resources. The default version of Syskill & Webert uses a simple Bayesian classifier to determine the document relevance. Sets of positive/negative examples are used to learn the user profile and keep it up-to-date. A feature selection phase based on the mutual information analysis is employed to determine the words to use as features.

Besides the explicit feedback technique that forces users to explicitly construct the model by describing the information in which he is interested, Syskill & Webert is able to suggest interesting Web pages. Nevertheless, this decision is made by analyzing the content of pages, and therefore it requires pages to be downloaded first.

2.4 Semantic Networks and Frames

Semantic networks consist of concepts linked to other concepts by means of various kinds of relations. It can be represented as a directed graph consisting of vertices which represent concepts and edges, themselves representing the semantic relations between the concepts. They were initially studied as a reasonable view of how semantic information is organized within a human memory [45].

The *SiteIF* project [33] uses semantic networks built from the frequencies of co-occurring keywords. In this approach, keywords are extracted from the pages in the user's browsing history. These keywords are mapped into *synsets* (grouping English words into sets of synonyms) contained in the *WordNet* database [37], a semantic lexicon for the English language, identifying meanings that are used as nodes in the semantic network-based profile.

The SiteIF model of the profile is kept up to date by "looking over the user's shoulder", trying to anticipate which documents in the Web repository the user might find interesting. It builds a representation of the user's interest by taking into account some properties of words in documents browsed by the user, such as their co-occurrence and frequency. This profile is used to provide personalized browsing, the complementary information seeking activity to search.

A further approach based on a semantic network structure is *ifWeb* [3], an intelligent agent capable of supporting the user in the Web surfing, 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 makes use of semantic networks in order to create profiles of users. More specifically, the profile is represented as a weighted semantic network whose nodes correspond to terms found in documents, and textual descriptions given by the user as positive or negative examples, that is, through relevance feedback. The arcs of the network link together terms that co-occurred in some documents. The use of semantic networks and co-occurrence relationships allows ifWeb to overcome the limitations of simple keyword matching, particularly polysemy, because each node represents a concept and not just ambiguous keywords.

The relevance feedback technique helps users to explicitly refine their profile, selecting which of the suggested documents satisfy their needs: ifWeb autonomously extracts the information necessary to update the user profile from the documents on which the user expressed some positive feedback. If some of the concepts in the user profile have not been reinforced by the relevance feedback mechanism over a long period of time, a temporal decay process called *rent* lowers the weights associated with these concepts. This mechanism allows the profile to be maintained so that it always represents the current interests of the user.

A different AI representation, which organizes knowledge into chunks is called *frames* [38]. Within each frame are many kinds of information, e.g., how to use the frame, expectations, etc.. The frame's *slots* are attributes for which fillers (scalar values, references to other frames or procedures) have to be specified and/or computed. Collections of frames are to be organized in frame systems, in which the frames are interconnected. Unlike slots, the frame's *properties* are inherited from superframes to subframes in the frame network according to some inheritance strategy. The processes working on such frame structures are supposed to match a frame to a specific situation, to use default values to fill unspecified aspects, and so on.

The *Wifs* system's goal is to filter Web documents retrieved by a search engine in response to a query input by the user [36]. This system re-rank urls returned by the search engine, taking into account the profile of the user who typed in the query. The user can provide feedback on the viewed documents, and the system accordingly uses that feedback to update the user model.

The user model consists of frame slots that contain terms (topics), each associated with other terms (co-keywords). This information forms a semantic network. Slot terms, that is, the topics, must be selected from those contained in a *Terms Data Base* (TDB), previously created by experts who select the terms deemed mostly relevant for the pertinent domain.

Document modeling is not based on traditional IR techniques, such as the Vector Space Model. The abstract representation of the document may be seen as a set of 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 directly proportional to the frequency with which term t appears in the body, and the frequency with which term t appears in the document title.

In order to evaluate each document, the system builds a vector $\overrightarrow{Rel}$, where the element Rel_i represents a relevant value of term t_i compared to user information needs.

Relevance is calculated taking into account the user model, the query, and the TDB. The relevance value $Rel_{new}(t)$ of term t, which simultaneously belongs to the document and user model, as a slot topic, is calculated by intensifying the old relevance value, $Rel_{old}(t)$. Basically, the new relevance value of term t is obtained from the old value, in this case initially equal to 0, as a proportional contribution to the sum of all semantic network weights of the user model containing term t as topic.

If the term that belongs to the user model and document, also belongs to the q query input by the user, then the term relevance value is further strengthened by means of w_{slot}, the weight associated with topic t. In this way, query q, which represents the user's immediate needs, is used to effectively filter the result set to locate documents of interest.

If term t belongs both to the query q, the document d, and the TDB, but is not included in the user model, then the only contribution to relevance is given by the old relevance plus a given constant. Finally, if term t is a topic for the $slot_j$, all contributions given by co-keywords are considered. 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.

The system calculates the final relevance score assigned to the document as follows:

$$Score(Doc) = \overrightarrow{Occ} \cdot \overrightarrow{Rel}$$

where $\overrightarrow{Occ}$ is the vector consisting of elements Occ_i, that is, the occurrence value of a term t_i, and $\overrightarrow{Rel}$ is the vector consisting of elements Rel_i.

Relevance feedback dynamically updates the user model upon receipt of the viewed documents provided by the user. The *renting* mechanism is used to decrease the weights of the terms appearing in the model that do not receive positive feedback after a certain period of time.

It should be pointed out how the semantic network knowledge representation, and its possibility to store text allowing semantic processing, is an intriguing feature in user modeling research, but it is not completely clear how much of the original model's formulation has been kept in the developed prototypes. For example, the ifWeb's semantic network differs from that of the knowledge representation domain, since it represents terms and not concepts, and the arcs are not explicit semantic relationships, rather just co-occurrences in some documents. Nevertheless, the evaluation performance obtained by the user modeling systems based on this technique in the Web domain is extremely interesting, in spite of the computational resources for processing the complex structures.

3 Other Approaches for Personalized Search

In the previous section, we discussed some AI-based techniques in order to model the user needs and represent the Web documents in information systems. Further personalization approaches can be devised in order to improve the current human-computer interaction with Web search tools. In this section, we discuss two of these approaches: ontology-based search and adaptive focused crawling.

3.1 Ontology-based Search

Ontologies and the Semantic Web[1] are two important research fields that are beginning to receive attention in the Web Intelligent Search context. Formal ontologies based on logic languages, such as OWL, are able to publish information in a machine readable format, supporting advanced Web search, software agents and knowledge management. Dolog *et al.* [16] 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 enables the formalization of personalization techniques in a common language, such as *FOL*, and reasoning over the Semantic Web.

Nevertheless, assigning a unique semantic (meaning) to each piece of information on the Web and stating all the possible relationships among the available

[1] ttp://www.w3.org/2001/sw/

data is a difficult task to automate and impossible to perform for a human being. For this reason, some prototypes are based on hierarchical taxonomies, such as YAHOO or OPEN DIRECTORY PROJECT (ODP)[2] directories. In these kinds of ontologies, the objects are simply categories that contain references to Web documents, and their interrelationships allow making inferences and retrieving more relevant information than the IR keyword-based search.

The goal is to create ontology-based user profiles and use these profiles to personalize the human-computer interaction with the Web, during a browsing session or when the user analyzes the search engine results.

The *misearch* system [51] improves search accuracy re-ranking the results obtained by an traditional search service by giving more importance to the documents related to topics contained in the query histories and/or examined search results. The user profiles are represented as weighted concept hierarchies. The OPEN DIRECTORY PROJECT is used as the reference concept hierarchy for the profiles.

The system collects for each user two different types of data collected by means of a wrapper of the search engine: the submitted queries, and the *snippets* of the results selected by the user. The approach is based on a classifier trained on the OPEN DIRECTORY PROJECT's hierarchy, which chooses the concepts most related to the collected information, assigning higher weights to them.

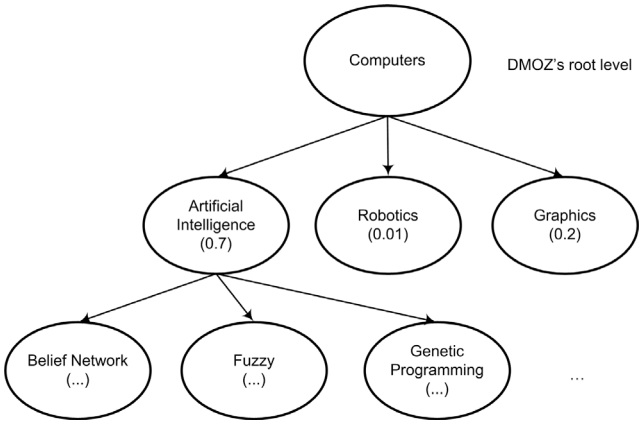

Fig. 1. The **misearch**'s user profile based on the ODP directory, where the categories are weighted according to what the user has seen in the past.

A matching function calculates the degree of similarity between each of the concepts associated with result snippet j and the user profile i:

[2] http://www.dmoz.org

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

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.

Finally, each document is assigned a weight, used for result re-ranking. The results that match the user's interests are placed higher in the list. These new ranks are drawn combining the previous degree of similarity with GOOGLE's original rank.

As for the performance measured in terms of the rank of the user-selected result, it improves by 33%. A user profile built from snippets of 30 user-selected results showed an improvement by 34%. Therefore, even though the text a user provides to the search engine is quite short, it is enough to provide more accurate, personalized results, and the implicit feedback reduces the time users need to spend to learn and keep their profiles up-to-date.

A similar approach is taken by Liu and Yu [32], 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 ODP category hierarchy.

Because queries are usually very short, they are often ambiguous. For each query and the current context, the system assigns the most appropriate ODP categories. Afterwards, the system performs query expansion based on the top-matching category reducing the ambiguity of the results.

The categories in the user profile are represented by 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.

3.2 Adaptive Focused Crawling

Adaptive Focused crawling concerns the development of particular crawlers, that is, a software system that traverses the Web collecting HTML pages or other kinds of resources [43, 53], able to find out and collect only Web pages that satisfy some specific topics [12]. This kind of crawler is able to build specialized/focused collections of documents reducing the computational resources needed to store them. Specialized search engines use those collections to retrieve valuable, most reliable and up-to-date documents related to the given topics. Vertical portals, personalized electronic newspapers [6], personal shopping agents [17] or conference monitoring services [26] are examples of implementations of those kinds of search engines in realistic scenarios.

New techniques to represent Web pages, such as the ones discussed in Sect. 2, and match these representations against the user's queries, such as algorithms based on Natural Language Processing (NLP), usually avoided due to the computational resources needed to run them, can be implemented owing to the reduced dimension of the document sets under consideration.

In this section, we discuss two recent approaches proposed in order to build focused crawlers based on sets of autonomous agents that wander the Web environment looking for interesting resources. The first approach is *InfoSpiders* [35] based on genetic algorithms where an evolving population of intelligent agents browses the Web driven by user queries. Each agent is able to draw the relevance of a resource with a given query and reason autonomously about future actions regarding the next pages to download and analyze. The goal is to mimic the intelligent browsing behavior of human users with little or no interaction among agents.

The agent's *genotype* consists of set of chromosomes, which determines its searching behavior. This very genotype is involved in the offspring generation process. The two principal components of the genotype are a set of keywords initialized with the query terms and a vector of real-valued weights, initialized randomly with uniform distribution, corresponding with the information stored in a feed-forward neural network. This network is used to judge what keywords in the first set best discriminate the documents relevant to the user. For each link to visit, the related text that is also included in the genotype set is extracted by the agent. The text is given as input to the neural network. The output of the net is used to draw a probability to choose and visit the given link. The outcome of the agent's behavior, as well as the user feedback, are used to train the neural network's weights. If any error occurs due to the agent's action selection, that is, visiting of irrelevant pages, the network's weights are updated through the backpropagation of error.

Mutations and crossovers among agents provide the second kind of adaptivity to the environment. Offspring are recombined by means of the crossover operation and, along with the mutation operator, they provide the needed variation to create agents that are able to behave better in the environment, retrieving an increased number of relevant resources.

A value corresponding to the agent's energy is assigned at the beginning of the search, and it is updated according to the relevance of the pages visited by that agent. The neural networks and the genetic operators aim at selecting the words that well describe the document that led to the energy increase, modifying the agents' behavior according to prior experience, learning to predict the best links to follow. The energy determines which agents are selected for reproduction and the ones to be killed amidst the population.

Ant foraging behavior research inspired a different approach for building focused crawlers. In [20, 21] an adaptive and scalable Web search system is described, based on a multi-agent reactive architecture, derived from a model of social insect collective behavior [7].

Biologists and ethologists created this model to understand how blind animals, such as ants, are able to find out the shortest ways from their nest to the feeding sources and back. This phenomenon can be easily explained, since ants can release pheromone, a hormonal substance to mark the ground. In this way, they can leave a trail along the followed path. This pheromone allows other ants

to follow the trails on the covered paths, reinforcing the released substance with their own.

The first ants returning to their nest from the feeding sources are those which chose the shortest paths. The back-and-forth trip (from the nest to the source and back) allows them to release pheromone on the path twice. The following ants leaving the nest are attracted by this chemical substance, therefore are more likely choose the one which has been frequently covered by the previous ants. For this reason, they will direct themselves towards the shortest paths.

In the proposed focused crawling algorithm each agent corresponds to a virtual ant that has the chance to move from the hypertextual resource where it is currently located to another, if there is a link between them. A sequence of links, i.e., pairs of urls, represents a possible agent's route where, at the end of each exploration, the pheromone trail could be released on. The available information for an agent when it is located in a certain resource is: the matching result of the resource with the user needs, e.g., query, and the amount of the pheromone on the paths corresponding to the outgoing links.

The ants employ the pheromone trail to communicate the exploration results one another: the more interesting are the resources an ant is able to find out, the more pheromone trail it leaves on the followed path. As long as a path carries relevant resources, the corresponding trail will be reinforced, thus increasing the number of attracted ants.

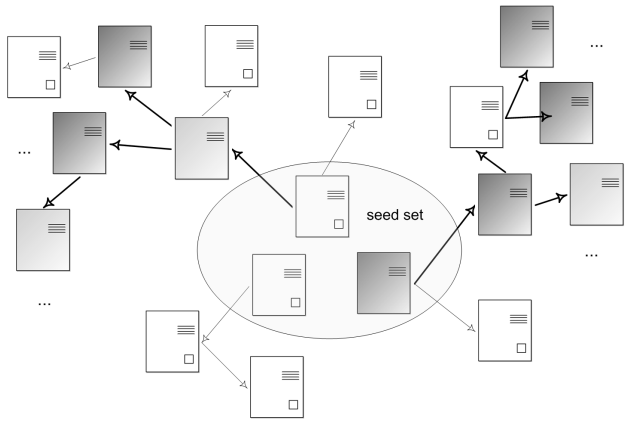

Fig. 2. Starting from a initial URL set, the agents look for interesting pages on the Web. If an ant finds a good page, a trail is left on the path. In this way, during the next cycles other ants will be attracted by the paths that head to good pages.

The crawling process is organized in cycles; in each one of them, the ants make a sequence of moves among the hypertextual resources. The maximum number of allowable moves varies proportionally according to the value of the

current cycle. At the end of a cycle, the ants update the pheromone intensity values of the followed path.

The resources from which the exploration starts can be collected from the first results of a search engine, or the user's bookmarks, and correspond to the seed URLs. To select a particular link to follow, a generic ant located on the resource url_i at the cycle t, draws the *transition probability* value $P_{ij}(t)$ for every link contained in url_i that connects url_i to url_j. The $P_{ij}(t)$ is considered by the formula:

$$P_{ij}(t) = \frac{\tau_{ij}(t)}{\sum_{l:(i,l)\in E} \tau_{il}(t)} \tag{2}$$

where $\tau_{ij}(t)$ corresponds to the pheromone trail between url_i and url_j, and $(i,l) \in E$ indicates the presence of a link from url_i to url_l.

At the end of a cycle, when the limit of moves per cycle is reached, the *trail updating process* is performed. The *updating rule* for the pheromone variation of the k-ant corresponds to the mean of the visited resource scores:

$$\Delta\tau^{(k)} = \frac{\sum_{j=1}^{|p^{(k)}|} score(p^{(k)}[j])}{|p^{(k)}|} \tag{3}$$

where $p^{(k)}$ is the ordered set of pages visited by the k-ant, $p^{(k)}[i]$ is the i-th element of $p^{(k)}$, and $score(p)$ is the function that, for each page p, returns the similarity measure with the current information needs: a $[0,1]$ value, where 1 is the highest similarity.

Afterworlds, the τ values are updated. The τ_{ij} trail of the generic path from url_i to url_j at the cycle $t+1$ is affected by the ant's pheromone updating process, through the computed $\Delta\tau^{(k)}$ values:

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \sum_{k=1}^{M} \Delta\tau^{(k)} \tag{4}$$

ρ is the trail evaporation coefficient that avoids unlimited accumulation of substance caused by the repeated positive feedback. The summation widens to a subset of the N ants living in the environment.

Two empirical observations are included in the developed architecture: Web pages on a given topic are more likely to link to those on the same topic [14], and Web pages' contents are often not self-contained [39].

The Ant crawler has two forms of adaptivity: the first concerns the query refinement during the execution when the results are not so satisfactory or, in general, the user does not know how to look for a query able to express what he wants. The second form regards the environment instability due to the updates of Web pages. These two types of adaptivity are possible because the value of the pheromone intensities $\tau_{ij}(t)$ is updated at every cycle, according to the visited resource scores.

Besides Genetic algorithms and the Ant paradigm, further focused crawlers are based on AI techniques and approaches, such as reinforcement learning [46, 11] or Bayesian statistical models [1].

4 Conclusions

Through the use of intelligent search, in particular techniques to represent user needs and Web documents, tailoring search engines' results, it is possible to enhance the human-computer interaction that provides useful information to the user.

AI-based knowledge representations prove their effectiveness in this context, outperforming the traditional and widespread IR approach. Moreover, ontologies can be employed to better understand and represent user needs, and intelligent search methods have been included in autonomous crawlers in order to retrieve up-to-date resources regarding particular topics.

Further techniques not mentioned in this chapter; nevertheless, Plan recognition (see for example [10]) and NLP are related to this context. The former attempts to recognize patterns in user behavior, analyzing aspects of their past behavior, in order to predict goals and their forthcoming actions. Language processing aims at understanding the meaning of Web content. How it relates to a user query can fuel the fire of further important research in the personalization domain.

References

1. Charu C. Aggarwal, Fatima Al-Garawi, and Philip S. Yu. Intelligent crawling on the world wide web with arbitrary predicates. In *Proceedings of the 10th World Wide Web Conference (WWW10)*, pages 96–105, Hong Kong, 2001.
2. Leonardo Ambrosini, Vincenzo Cirillo, and Alessandro Micarelli. A hybrid architecture for user-adapted information filtering on the world wide web. In A. Jameson, C. Paris, and C. Tasso, editors, *Proceedings of the 6th International Conference on User Modeling (UM97)*, pages 59–61. Springer, Berlin, 2–5 1997.
3. Fabio A. Asnicar and Carlo Tasso. 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)*, pages 3–12, Sardinia, Italy, 1997.
4. Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.
5. Marko Balabanović and Yoav Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.
6. Krishna Bharat, Tomonari Kamba, and Michael Albers. Personalized, interactive news on the web. *Multimedia Syst.*, 6(5):349–358, 1998.
7. Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. Inspiration for optimization from social insect behavior. *Nature*, 406:39–42, 2000.
8. Gary Boone. Concept features in re:agent, an intelligent email agent. In *Proceedings of the second international conference on Autonomous agents (AGENTS '98)*, pages 141–148, New York, NY, USA, 1998. ACM Press.

9. Chris Buckley, Gerard Salton, and James Allan. The effect of adding relevance information in a relevance feedback environment. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 292–300, New York, NY, USA, 1994. Springer-Verlag New York, Inc.
10. Sandra Carberry. Techniques for plan recognition. *User Modeling and User-Adapted Interaction*, 11(1-2):31–48, 2001.
11. Soumen Chakrabarti, Kunal Punera, and Mallela Subramanyam. Accelerated focused crawling through online relevance feedback. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, pages 148–159, New York, NY, USA, 2002. ACM Press.
12. Soumen Chakrabarti, Martin van den Berg, and Byron Dom. Focused crawling: A new approach to topic-specific web resource discovery. In *Proceedings of the 8th World Wide Web Conference (WWW8)*, pages 1623–1640, Toronto, Canada, 1999.
13. William S. Cooper. A definition of relevance for information retrieval. *Information Storage and Retrieval*, 7:19–37, 1971.
14. Brian D. Davison. Topical locality in the web. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 272–279, New York, NY, USA, 2000. ACM Press.
15. Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.
16. Peter Dolog, Nicola Henze, Wolfgang Nejdl, and Michael Sintek. Towards the adaptive semantic web. In François Bry, Nicola Henze, and Jan Maluszynski, editors, *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*, pages 51–68. Springer, 2003.
17. Robert B. Doorenbos, Oren Etzioni, and Daniel S. Weld. A scalable comparison-shopping agent for the world-wide web. In *AGENTS '97: Proceedings of the first international conference on Autonomous agents*, pages 39–48, New York, NY, USA, 1997. ACM Press.
18. Paolo Frasconi, Giovanni Soda, and Alessandro Vullo. Text categorization for multi-page documents : A hybrid naive bayes HMM approach. In *ACM/IEEE Joint Conference on Digital Libraries, JCDL 2001*, Roanoke, VA, USA, June 2001. ACM.
19. Robert Fung and Brendan Del Favero. Applying Bayesian networks to information retrieval. *Communications of the ACM*, 38(3):42–48, March 1995.
20. Fabio Gasparetti and Alessandro Micarelli. Adaptive web search based on a colony of cooperative distributed agents. In Matthias Klusch, Sascha Ossowski, Andrea Omicini, and Heimo Laamanen, editors, *Cooperative Information Agents*, volume 2782, pages 168–183. Springer-Verlag, 2003.
21. Fabio Gasparetti and Alessandro Micarelli. Swarm intelligence: Agents for adaptive web search. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004)*, 2004.
22. Gianluigi Gentili, Mauro Marinilli, Alessandro Micarelli, and Filippo Sciarrone. Text categorization in an intelligent agent for filtering information on the web. *IJPRAI*, 15(3):527–549, 2001.
23. Antonio Gulli and Alessio Signorini. The indexable web is more than 11.5 billion pages. In *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 902–903, New York, NY, USA, 2005. ACM Press.

24. Thorsten Joachims, Dayne Freitag, and Tom M. Mitchell. Webwatcher: A tour guide for the world wide web. In *Proceedings of the 15h International Conference on Artificial Intelligence (IJCAI1997)*, pages 770–777, 1997.
25. Diane Kelly and Jaime Teevan. Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, 37(2):18–28, 2003.
26. Andries Kruger, C. Lee Giles, Frans M. Coetzee, Eric Glover, Gary W. Flake, Steve Lawrence, and Christian Omlin. Deadliner: building a new niche search engine. In *CIKM '00: Proceedings of the ninth international conference on Information and knowledge management*, pages 272–281, New York, NY, USA, 2000. ACM Press.
27. Bruce Krulwich and Chad Burkey. The contactfinder agent: Answering bulletin board questions with referrals. In *Proceedings of the 13th National Conference on Artificial Intelligence and 8th Innovative Applications of Artificial Intelligence Conference, AAAI96, IAAI96*, pages 10–15, 1996.
28. Man Lan, Chew-Lim Tan, Hwee-Boon Low, and Sam-Yuan Sung. A comprehensive comparative study on term weighting schemes for text categorization with support vector machines. In *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 1032–1033, New York, NY, USA, 2005. ACM Press.
29. Alon Y. Levy and Daniel S. Weld. Intelligent internet systems. *Artificial Intelligence*, 118(1-2):1–14, 2000.
30. Henry Lieberman. Letizia: An agent that assists web browsing. In Chris S. Mellish, editor, *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI1995)*, pages 924–929, Montreal, Quebec, Canada, 1995. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.
31. Henry Lieberman, Neil W. Van Dyke, and Adrian S. Vivacqua. Let's browse: a collaborative web browsing agent. In *Proceedings of the 4th International Conference on Intelligent User Interfaces (IUI'99)*, pages 65–68, Los Angeles, CA, USA, 1998. ACM Press.
32. Fang Liu, Clement Yu, and Weiyi Meng. Personalized web search for improving retrieval effectiveness. *IEEE Transactions on Knowledge and Data Engineering*, 16(1):28–40, 2004.
33. Bernardo Magnini and Carlo Strapparava. User modelling for news web sites with word sense based techniques. *User Modeling and User-Adapted Interaction*, 14(2):239–257, 2004.
34. Andrew McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *Proceedings of AAAI-98, Workshop on Learning for Text Categorization*, 1998.
35. Filippo Menczer and Richard K. Belew. Adaptive retrieval agents: Internalizing local context and scaling up to the web. *Machine Learning*, 31(11–16):1653–1665, 2000.
36. Alessandro Micarelli and Filippo Sciarrone. Anatomy and empirical evaluation of an adaptive web-based information filtering system. *User Modeling and User-Adapted Interaction*, 14(2-3):159–200, 2004.
37. George A. Miller and Christiane Fellbaum. Lexical and conceptual semantics. In B. Levin and S. Pinker, editors, *Advances in Neural Information Processing Systems*, pages 197–229. Blackwell, Cambridge and Oxford, England, 1993.
38. Marvin Minsky. A framework for representing knowledge. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA, 1974.
39. Yoshiaki Mizuuchi and Keishi Tajima. Finding context paths for web pages. In *Proceedings of the 10th ACM Conference on Hypertext and Hypermedia: Returning to Our Diverse Roots (HYPERTEXT99)*, pages 13–22, Darmstadt, Germany, 1999.

40. Alexandros Moukas and Pattie Maes. Amalthaea: An evolving multi-agent information filtering and discovery system for the WWW. *Autonomous Agents and Multi-Agent Systems*, 1(1):59–88, 1998.
41. Michael J. Pazzani, Jack Muramatsu, and Daniel Billsus. Syskill webert: Identifying interesting web sites. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-96)*, pages 54–61, Portland, OR, USA, August 1996. AAAI Press.
42. Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, 1988.
43. Brian Pinkerton. Finding what people want: Experiences with the webcrawler. In *Proceedings of the 2nd World Wide Web Conference(WWW2)*, Chicago, USA, 1994.
44. Michael H. Pryor. The effects of singular value decomposition on collaborative filtering. Technical report, Hanover, NH, USA, 1998.
45. Ross M. Quillian. Semantic memory. In Marvin Minsky, editor, *Semantic information processing*, pages 216–270. The MIT Press, Cambridge, MA, USA, 2–5 1968.
46. Jason Rennie and Andrew McCallum. Using reinforcement learning to spider the web efficiently. In *ICML '99: Proceedings of the Sixteenth International Conference on Machine Learning*, pages 335–343, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.
47. Robertson, S. E., and Walker, S. On relevance weights with little relevance information. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Relevance Feedback, pages 16–24, 1997.
48. Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John T. Riedl. Application of dimensionality reduction in recommender systems - a case study. In *Web Mining for E-Commerce – Challenges and Opportunities*, Boston, MA, USA, 2000.
49. Jaques Savoy and Justin Picard. Retrieval effectiveness on the web. *Information Processing & Management*, 37(4):543–569, 2001.
50. Barry Smyth, Evelyn Balfe, Jill Freyne, Peter Briggs, Maurice Coyle, and Oisin Boydell. Exploiting query repetition and regularity in an adaptive community-based web search engine. *User Modeling and User-Adapted Interaction*, 14(5):383–423, 2005.
51. Mirco Speretta and Susan Gauch. Personalized search based on user search histories. In *Web Intelligence (WI2005)*, France, 2005. IEEE Computer Society.
52. Jian-Tao Sun, Hua-Jun Zeng, Huan Liu, Yuchang Lu, and Zheng Chen. Cubesvd: a novel approach to personalized web search. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 382–390, New York, NY, USA, 2005. ACM Press.
53. Budi Yuwono, Savio L. Y. Lam, Jerry H. Ying, and Dik L. Lee. A World Wide Web resource discovery system. In *Proceedings of the 4th World Wide Web Conference (WWW4)*, pages 145–158, Boston, Massachusetts, USA, 1995.