Chapter 7
User Profiles Modeling in Information Retrieval Systems

Václav Snášel, Ajith Abraham, Suhail Owais, Jan Platoš, and Pavel Krömer

Abstract The requirements imposed on information retrieval systems are increasing steadily. The vast number of documents in today's large databases and especially on the World Wide Web causes problems when searching for concrete information. It is difficult to find satisfactory information that accurately matches user information needs even if it is present in the database. One of the key elements when searching the web is proper formulation of user queries. Search effectiveness can be seen as the accuracy of matching user information needs against the retrieved information. As step towards better search systems represents personalized search based on user profiles. Personalized search applications can notably contribute to the improvement of web search effectiveness. This chapter presents design and experiments with an information retrieval system utilizing user profiles, fuzzy information retrieval and genetic algorithms for improvement of web search.

7.1 Introduction

Information retrieval activity is a derivation of real-world human communication. An information or knowledge, stored in a data repository by one person, is desired to be retrieved by another. Data repositories, emphasizing the Internet as the ultimate

V. Snášel (✉), J. Platoš, and P. Krömer
Department of Computer Science, Faculty of Electrical Engineering and Computer Science, VŠB, Technical University of Ostrava, 17. listopadu 15, 708 33 Ostrava, Poruba, Czech Republic
e-mail: vaclav.snasel@vsb.cz; jan.platos@vsb.cz; pavel.kromer.fei@vsb.cz

A. Abraham
Center of Excellence for Quantifiable Quality of Service, Norwegian University of Science and Technology, O.S. Bragstads plass 2E, 7491 Trondheim, Norway
e-mail: ajith.abraham@ieee.org

S. Owais
Department of Computer Science, Faculty of Information Technology, Applied Science University, Amman, 11931 Jordan
e-mail: dr_suhail@asu.edu.jo

R. Chbeir et al., *Emergent Web Intelligence: Advanced Information Retrieval*, Advanced Information and Knowledge Processing, DOI 10.1007/978-1-84996-074-8_7,

one, are used for persisting information in both, time and space. Data available on the Internet might be accessed by users distant in time and space. Unfortunately, the omnipresence of data is not equal to instant availability of information. In general, data can be seen as a state of information used for storage purposes, encapsulating the information content itself. A speech is not information, it contains information. An article is not information, it contains information. An electronic document can be seen similarly. To exploit stored data, it is desired to access the contained information in an efficient way. Such *information retrieval* activity is not an easy task and its complexity depends specially on the dimension of searched data basis. Moreover, when we are trying to automate information search process, the requirement to understand is becomes crucial. To retrieve the information in document, its content should be understood. To present required information to inquirer, the requests must be understood and correctly interpreted. Advanced techniques of information retrieval are under investigation to provide both – better content representation and better query apprehension.

There is fuzziness in human mind. It involves the means of communication. Estimations and intuition are present. Vagueness, imprecision and mistakes occur. These facts influence both – information content of documents and search request formulations. Contrariwise, any automated search tool has rather crisp and rough picture (i.e., model) of the information content of data, providing satisfactory search service for data collections up to certain size. Inevitably, the enormous growth of data repositories and especially of the Internet brings up more and more problems when performing information retrieval tasks. The amount of regular users of search services is growing as well. One approach to improve information retrieval in such conditions is approximating reality better than before. To improve the efficiency of information retrieval, soft computing techniques with special emphasis on fuzzy technology are being intensively investigated. When modeling information and requests containing vagueness or imprecision, fuzzy set theory providing formal background to deal with imprecision, vagueness, uncertainty and similar concepts might be used, introducing significant improvements to the search results.

The requirements imposed on search applications are increasing steadily. The amount of available data is growing and user demands as well. The search application should provide the users with accurate, sensible responses to their requests. Unified consensual approach to search requirements of all inquirers becomes with growing amount of data and documents on the WWW inefficient to satisfy needs of large number of individuals desiring to retrieve particular information from Internet. Personalized approach to the needs of each user is general trend in state-of-the-art web applications including search engines. Personalization, based on stored knowledge of users general needs, area(s) of interest, usual behavior, long and short term context and search practices can be evaluated when improving web search applications, no matter if they are standalone search engines or more advanced meta-search systems lying on the top of individual search applications.

User profiles, personalization of web search tasks and soft information retrieval are current challenges. Information retrieval optimization based on knowledge of previous user search activities and fuzzy softening of both, search criteria and information models, aims at enriching document sets retrieved in response to user requests and helping user when she or he has no clear picture of searched information. In this paper we describe implicit user modeling algorithm based on click-through data analysis and introduce genetic and fuzzy oriented approach to query optimization with the goal to determine useful search queries describing documents relevant to users area of interest as deducted from previous searches as a tool helping user to fetch the most relevant information in his or her current context.

The rest of this chapter is organized as follows: In Section 7.2, some background on information retrieval and fundamentals of information retrieval systems is provided. Fuzzy logic is briefly technology and its application in the area of information retrieval is introduced. Section 7.3 summarizes the usage of Evolutionary Computation, Genetic Algorithms, Genetic Programming and its application to information retrieval tasks. In Section 7.4, we present our contribution extending the usage of genetic algorithms for search optimization in both, crisp and fuzzy information retrieval systems. Experiment results are presented in Section 7.5 and finally some conclusions are also provided.

7.2 Information Retrieval

The area of *Information Retrieval* (IR) is a branch of Computer Science dealing with storage, maintenance and information search within large amounts of data. The data could be all – textual, visual, audio or multimedia documents [7]. The rest of this chapter is devoted to information retrieval dealing with extensive collections of unstructured textual documents.

An *Information Retrieval System* (IRS) is a software tool for data representation, storage and information search. The amount of documents contained in data collections managed by IRS is usually very large and the task of easy, efficient and accurate information search is specially highlighted. General architecture of an information retrieval system is shown in Fig. 7.1 [7].

Documentary collection is for the search purposes analyzed and transformed into suitable internal representation in a process called indexing. The real world information need of an IRS user must be for the use with particular IRS expressed by the means of query language understandable to that system. A search query is then evaluated against the internal document representation and the system decides whether and how much are particular documents relevant to the query. The way of document indexing, structure of internal document representation, query language and document-query matching mechanism depends on certain IRS model which is a theoretical background below particular information retrieval system [7]. For regular users provides an IRS two main functions: data storage and information retrieval in order to satisfy users' information need.

An *information need* is a state in which ones own knowledge is insufficient for satisfying her or his demands. If an IRS is to be used for information search, the demanded information need must be expressed in query language of the particular IRS

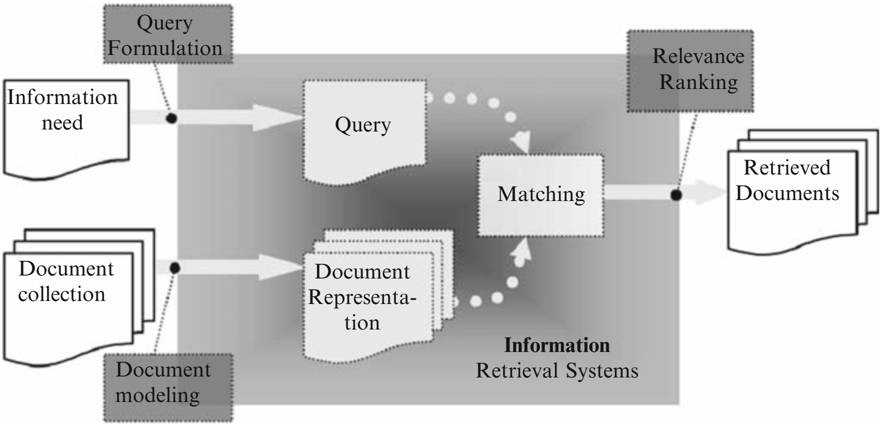
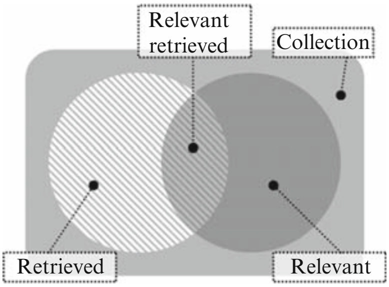


Fig. 7.1 An information retrieval system

Fig. 7.2 Documents in collection classified in response to a query



in a process called querying. The search system attempts to find in managed documentary collection entries relevant to the query. Ordered set of *retrieved* documents is then offered to the user. Retrieved documents are such subset of documentary collection that is considered by the information retrieval application to be *relevant* to the user query. Retrieved documents are presented in certain ordering as a source of information to satisfy information need stated in the query. The document ordering is based on particular ranking strategy which is realized by certain ranking function.

The typical allocation of documents within the collection in response to a query is illustrated in Fig. 7.2. We can see that not all relevant documents are usually retrieved and moreover, some non-relevant documents could be included in the set of retrieved documents. We may also legitimately consider different documents to be relevant to the query in a certain degree. One of the main goals in the research of IR systems is to improve the accuracy of retrieved document set. It means to maximize the subset of retrieved relevant documents and minimize the subset of retrieved non-relevant documents.

Documents classified against to a query as relevant and non-relevant, though the entire concept of relevance is a subject of discussion with no universal convergence yet. Objective relevance is an algorithmic measure of the degree of similarity between the query representation and the document representation. It is also referred to as a topicality measure, referring to the degree to which the topic of the retrieved information matches the topic of the request [12]. Subjective relevance is user-centric and deals with fitness for use of the retrieved information [29]. Subjective relevance involves intellectual interpretation by human assessors or users [5] and should be seen as a cognitive, dynamic process involving interaction between the information user and the information source. A general high-level relevance criterion is whether or not (and alternatively how much) the particular document contributes to the saturation of user's information need expressed by a query presented to the system at the beginning of search session. Different inquirers might be satisfied with different response to the same question. Among the most important factors having impact on the user request is long and short term context of the particular inquirer. When evaluating a search expression, the knowledge of user's area of interest, abilities, language capabilities, current needs, etc., can be important contribution to the search efficiency improvement. These are among the most fundamental reasons for personalized search research, user modeling and user profiling.

7.2.1 Information Retrieval Models

An *IR model* is a formal background defining internal document representation, query language and document-query matching mechanism. Consequently, the model determines document indexing procedure, result ordering and other aspects of particular information retrieval system. In the following, we will present two influential IR models – Boolean IR model and vector space IR model [7, 16].

7.2.1.1 Boolean IR Model

Boolean IR model belongs to the oldest but till nowadays widely used information retrieval models [2, 7]. It is based on set theory, Boolean logic and exact document-query match principle. The name Boolean comes from the fact that the query language uses as search expressions Boolean logic formulas composed of search terms and standard Boolean operators AND, OR and NOT [2]. The documents are represented as sets of indexed terms. The document indexing procedure distinguishes only whether a term is contained in the document or not and assigns to the term indexing weight 1 if the term is existed in the document or 0 if not. The inner representation of a documentary collection is a binary matrix composed of document representing vectors with term weights as coordinates. Therefore every column represents weight of certain term in all documents in the whole collection. Formally, an index of documentary collection containing n terms and m documents in Boolean IR model is described as shown in Eqs. 7.1 and 7.2, where d_i represents i^{th} document, t_{ij} the weight of j^{th} term in i^{th} document and D denotes the index matrix.

$$d_i = (t_{i1}, t_{i2}, \ldots, t_{in}), \quad \forall t_{ij} \in \{0, 1\} \tag{7.1}$$

$$D = \begin{pmatrix} t_{11} & t_{12} & \cdots & t_{1n} \\ t_{21} & t_{22} & \cdots & t_{2n} \\ \vdots & \vdots & & \vdots \\ t_{m1} & t_{m2} & \cdots & t_{mn} \end{pmatrix} \tag{7.2}$$

The document-query matching procedure is based on the exact match principle. Only documents utterly satisfying all conditions stated by particular search query are considered to be relevant and thus retrieved in response to the query. When a document fully conforms to the search request, the query is against it evaluated, according to the Boolean algebra rules, as true. In the contrary case, when the document is in conflict with at least one of the clauses in the search request, the query is evaluated as false. In that way, the set of all documents in the collection is divided into two disjunctive subsets – retrieved and non-retrieved documents. There is no consideration of different degrees of document-query relevancy. All retrieved documents are supposed to be equally (fully) relevant to the query and all non-relevant documents are expected to be equally non-relevant. The ordering, in which are the results presented to the user, does not depend on the relevancy but on other factors such as date of last modification, document length, number of citations and so on [2, 7, 11]. There are numerous variations of basic Boolean IR model. Frequent modification relies on addition of advanced query operators – XOR implementing the logical exclusive OR operation, operator OF simplifying the notion of search formulas or, among others, operator NEAR expressing the requirement to retrieve documents having several terms near each other [11].

Apparently, the greatest advantage of Boolean IR model lies in exuberance and flexibility of its query language, allowing expressing very sophisticated and complex search requirements. On the other hand, to formulate such powerful search queries appropriately, the user should have at least minimal knowledge of Boolean algebra. Remarkable disadvantage of Boolean IR model is the crisp differentiation of documentary collection in response to query and therefore impossibility to use some relevance ranking technique to present retrieved documents sorted in relevance order. Because of this, a too restrictive query could cause denial of useful documents and contrariwise a too general query might retrieve additional non-relevant documents [15]. The Boolean IR model provides the basis for extended Boolean IR model introducing the principles of fuzzy set techniques and fuzzy logic to the area of information retrieval.

7.2.1.2 Vector Space Model

Vector space model (VSM) is based on interpretation of both, documents and queries, as points in a multidimensional document space [7, 11]. The dimension of the document space is given by the number of indexed terms in the documentary collection. Every term in every document assigned a weight representing the coordinate in multidimensional space. The weight is based on the importance of corresponding term in the document and in the scope of whole collection respectively. Greater weight means greater importance of particular term [2,7,16]. Formal description of VSM is almost identical to the description of Boolean model as provided in Eqs. 7.1 and 7.2. The domain of t_{ij} in VSM is the set of real numbers R. Query q is formalized as a vector of searched terms as shown in Eq. 7.3.

$$q = (t_{q1}, t_{q2}, \ldots, t_{qn}), \quad \forall t_{qj} \in R \tag{7.3}$$

In Boolean IR, indexing procedure was due to the simplicity of internal document representation trivial task. In VSM, the matrix representing documentary collection composed of real values – the weights of terms in documents. The weight assessment can be done manually (this is too expensive and inefficient) or automatically [11]. Several automatic indexing approaches were proposed. They assign real weights to the terms in documents. The weighting algorithms are usually based on statistical distribution of the terms in particular document with respect to their distribution among all documents in the collection. Among the most popular and widely deployed indexing techniques takes significant place Gerard Salton's $TFIDF_t$ introduced in [26]. Consider normalised term frequency of term t in document d shown in Eq. 7.4 as the ratio of frequency of each term in the document to the maximum term frequency in that document. Therefore, the greatest frequency of particular term in the document, the greatest normalized frequency of such term in the document.

$$f_{dt} = \frac{freq(t,d)}{\max(freq(t_i,d))} \tag{7.4}$$

Normalized inverse document frequency, defined as shown in Eq. 7.5, reflects the distribution of given term among all documents in the collection where, N is the number of all documents in the collection, N_t is number of documents containing at least one occurrence of the term t and g is some normalizing function. The rarer is the term in the scope of whole collection, the greater is its inverse document frequency. Finally, the weight of term t in document d according to $TFIDF_t$ is defined in Eq. 7.6.

$$IDF_t = g\left(\log \frac{N}{N_t}\right) \tag{7.5}$$

$$F(d,t) = f_{dt} \cdot IDF_t \tag{7.6}$$

Summarizing previous definitions, high weight will be assessed to the terms frequent in given document and rare in the scope of whole collection. It is obvious that such terms are good significant marks distinguishing current document from other documents. More indexing functions for VSM can be found, i.e., in [11]. Also queries have in VSM the form of documents (term vectors) and a term weighting function should be deployed. Query term weighting function example is shown in Eq. 7.7.

$$F(q,t) = \left(\frac{1}{2} + \frac{f_{qt}}{2}\right) \tag{7.7}$$

The document-query matching procedure in VSM based on best match principle. Both, document and query are interpreted as points in multidimensional space and we can evaluate similarity between them. Several formulas expressing numerically the similarity between points in the document space have been introduced [11]. Among the most popular are scalar product as shown in Eq. 7.8, and cosine measure as shown in Eq. 7.9 that can be interpreted as an angle between the query vector and document vector in m-dimensional document space.

$$Sim(q,d_i) = \sum_{j=1}^{m} t_{qj} \cdot t_{ij} \tag{7.8}$$

$$Sim(q,d_i) = \frac{\sum_{j=1}^{m} t_{qj} \cdot t_{ij}}{\sqrt{\sum_{j=1}^{m} t_{qj}^2 \cdot \sum_{j=1}^{m} t_{ij}^2}} \tag{7.9}$$

The similarity measure does not directly predicate document's relevance to the query. It is supposed that among documents similar to the query should be many relevant documents whereas among dissimilar documents is only few relevant ones [11]. Querying is in VSM based on the best match principle. All documents are during the query evaluation process sorted according their distance to the query and presented to the user. Omitting the vague relationship between point distance and document relevance, we can consider this ordering as relevance ranking.

VSM is more recent and advanced than Boolean IR model. Its great advantage lies in relevance based ordering of retrieved documents allowing easy deployment of advanced IR techniques such as document clustering, relevance feedback, query reformulation and topic evolution among others. Disadvantages are vague relationship between relevance and similarity and unclear query term explication. From the interpretation of query as a searched document prescription originates another significant disadvantage of VSM – the query language allows specifying only what should be searched and there are no natural means how to point out what should not be contained in retrieved documents.

7.2.2 IR Effectiveness Evaluation

When evaluating an information retrieval system, we are interested in the speed of search processing, user comfort, the possibilities of querying, result presentation and especially in the ability of *retrieving relevant documents*. As it was already noted, the concept of relevance is vague and uncertain. Though, it is useful to measure

IR effectiveness by the means of query-document relevance. Precision P and recall R are among the most used IR effectiveness measures as shown in Eq. 7.10. In the precision and recall definition, REL stands for the set of all relevant documents and RET for the set of all retrieved documents. Precision can be then understood as the probability of retrieved document to be relevant and recall as the probability of retrieving relevant document. For easier effectiveness evaluation were developed measures combining precision and recall into one scalar value. Among most popular of these measures are effectiveness E and F-score F [19] as shown in Eq. 7.11.

$$P = \frac{|REL \cap RET|}{|RET|} \quad R = \frac{|REL \cap RET|}{|REL|} \tag{7.10}$$

$$E = 1 - \frac{2}{\frac{1}{P} + \frac{1}{R}} \quad F = 1 - E = \frac{2PR}{P + R} \tag{7.11}$$

7.2.3 User Profiles in IR Systems

In previous section was shown that the concept of document-query relevance is highly subjective matter. Information need of particular user can be satisfied better if there are some knowledge about user specific needs, abilities, long and short term context. That is the field of personalized IR systems exploiting user profiles. A user profile (or user model) is a stored knowledge about particular user. Simple profile consists usually of keywords describing user's area of long time interest. Extended profile is replenished with information about the user such as name, location, mother tongue and so on. Advanced user profiles contain rather than set of keywords a list of queries characterizing user's behavior and habits [24].

User profile can be exploited to make the search task more personalized. Information retrieval system equipped with user profiles could utilize user-specific information from the profile for retrieving documents satisfying stated query with special respect to individual user, her or his preferences, needs, abilities, history, knowledge and context. User profile information might be evaluated when improving search process. Keywords from the profile can be used for query extension, query reformulation and for other techniques such as for improving search results. Such IR improvement techniques aim at retrieving information that satisfy users needs rather than information that was explicitly ask by potentially imprecise query [11]. User profile can be also exploited for document re-ranking according to individual preferences [24]. Advanced user profiles can instead of a set of keywords contain whole search expressions allocating areas of users long term interests and needs. Those queries are called persistent queries [9].

Explicit profiles, created by users or system administrators, are imprecise, not flexible enough and they do not reflect dynamic changes of user preferences. Instead, various techniques for automated creation and maintenance of user profiles are being investigated [6]. Automatically created and updated user profiles are referred as implicit user profiles. From the perspective of user profiling, IR systems can be divided into two categories: personalized IR systems providing personalized search services and consensual search system not aware of individual users [10].

7.2.3.1 Click-Through Data

Among the most promising methods, personalization techniques based on click-through data analysis attract attention [13, 30]. Click-through data recorded during web search activities might be seen as triplet (q, R, C) consisting of query q, ordered set of retrieved documents R and set of clicks C denoting documents user picked from the set of retrieved documents R, introducing individual search preferences [13].

The appeal of click-through data analysis for user profiling is based on several facts. It is omnipresent during web browsing click-through data is present in the web browsing activities always. The clicks are needed by the very essential structure of html documents and the WWW.

Click-through data is implicit user clicks are almost necessary to browse the web. Click-through data gathering must not be seen as an additional disturbing or obstructing activity. The clicks (or alternative link-following actions) are necessary to work with web. Additionally, click-through data has relevance feedback potential. The users click on links that he or she feels as relevant to his or her needs. Mostly, these links relevant by belief are really relevant to previous request although the essential information contained in click-through data is still under investigation.

Click-through is up-to-date and with appropriate analysis, the data gathered for sufficient time period could provide information on both, users long time interests and needs and immediate contemporary context. Finally, click-through data stored in query logs which can be used for many methods of information retrieval improvement, including offline techniques. Summarizing, most users click on rather relevant results and we should benefit from a large quantity of query logs. Experiments show that about 82% of the queries are in fact related to the topics of the clicked Web pages [31].

On the other hand, there are known issues with click-through data [13, 30]: it is usually noisy and rather incomplete piece of evidence about users relevance assessments. It is sparse since user clicks can cover only very small portion of WWW document space.

Click-through data collecting can be done on the top of current search systems and services. There could be a server based solution, observing user click behaviour from some central point like web application used as proxy for access to search services or client based solution tracking user clicks from, i.e., web browser. The web application is limited by its scope and as soon as the user leaves the application, the clicks are almost unrecordable. The client application is limited by the abilities of user workstations; the accommodation of such application must not be disturbing, i.e., it must not consume too much processor time, memory or disc space.

7.2.3.2 Document Relevance Estimation Based on Click-Through Data

Document relevance estimation model based on click-through data consists of recorded clicks committed by particular user. Each click c is captured a triplet (u, d, t), where $u \in U$ is particular user from the set of all users $U, d \in D$ is the clicked document and t is timestamp, describing moment in which the click was committed. $D = \{(u, n, s)\}$ is set of all documents known to the application. Consider $c_t : D \times U \rightarrow R^n$ as a set of timestamps describing clicks issued by particular user on certain document. For each document and user, the relevance $r : D \times U \rightarrow R$ is estimated by Eq. 7.12.

$$r(u, d) = \sum_{t \in c_t(u,d)} (f(t)) \tag{7.12}$$

In the above, the function f enumerates the contribution of click issued at time t to relevance estimate of the document and t denotes age of the click. The contribution function used in this paper is reversed asymmetric sigmoid as defined in Eq. 7.13.

$$f(t) = 1 - \frac{1}{\left(1 + e^{\frac{t + c \cdot ln(2^{\frac{1}{g}} - 1) - b}{c}}\right)^{g}} \tag{7.13}$$

Asymmetric reverse sigmoid as defined in Eq. 7.13 is highly customizable function. The parameter b denotes centre of the transition, c and g are used for enumeration of transition width w as specified in Eq. 7.14.

$$w = \left| c \cdot ln\left(4^{\frac{1}{g}} - 1\right) - c \cdot ln\left(4^{\frac{1}{g}} 3^{\frac{-1}{g}} - 1\right) \right| \tag{7.14}$$

Figure 7.3 illustrates the reverse asymmetric sigmoid with $b = 5$, $c = -2$ and $g = 10$. The x axis unit is day and the transition width is 3.256. The scale of x axis in presented work is subject of further customization and it is variable parameter for different deployment cases of presented method.

Additionally, user model contains recorded recent user queries to be exploited later during query optimization process as an initial population for optimizing genetic algorithm.

7.3 Evolutionary Computation

Evolutionary algorithms (EAs) belongs to a family of iterative stochastic search and optimization methods based on mimicking successful optimization strategies observed in nature [3, 8, 14, 20]. The essence of EAs lies in the emulation of Darwinian evolution utilizing the concepts of Mendelian inheritance for the use in computer

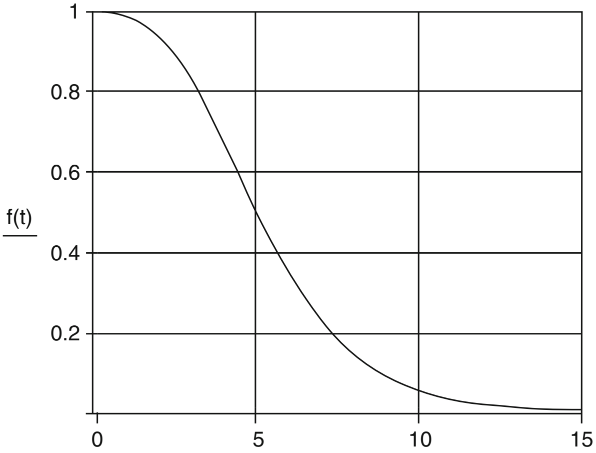


Fig. 7.3 Click contribution according to its age using reverse asymmetric sigmoid

science and applications [3]. Together with fuzzy sets, neural networks and fractals, evolutionary algorithms are among the fundamental members of the class of soft computing methods.

EAs operate with population (also known as pool) of artificial individuals (referred often as items or chromosomes) encoding possible problem solutions. Encoded individuals are evaluated using objective function which assigns a fitness value to each individual. Fitness value represents the quality (ranking) of each individual as solution of given problem. Competing individuals search the problem domain towards optimal solution [14]. In the following sections will be introduced general principles common for all methods belonging to the class of evolutionary algorithms.

7.3.1 Evolutionary Search Process

For the purpose of EAs, a proper encoding representing solutions of given problem as encoded chromosomes suitable for evolutionary search process, is necessary. Finding proper encoding is non-trivial problem dependent task affecting the performance and results of evolutionary search while solving given problem. The solutions might be encoded into binary strings, real vectors or more complex, often tree-like, hierarchical structures, depending on the needs of particular application.

The iterative phase of evolutionary search process starts with an initial population of individuals that can be generated randomly or seeded with potentially good solutions. Artificial evolution consists of iterative application of genetic operators, introducing to the algorithm evolutionary principles such as inheritance, survival of the fittest and random perturbations. Current population of problem solutions is modified with the aim to form new and hopefully better population to be used in next generation. Iterative evolution of problem solutions ends after satisfying specified termination criteria and especially the criterion of finding optimal solution. After terminating the search process, evolution winner is decoded and presented as the most optimal solution found.

7.3.2 Genetic Operators

Genetic operators and termination criteria are the most influential parameters of every evolutionary algorithm. All bellow presented operators have several implementations performing differently in various application areas. Selection operator is used for selecting chromosomes from population. Through this operator, selection pressure is applied on the population of solutions with the aim to pick more promising solutions to form following generation. Selected chromosomes are usually called parents. Crossover operator modifies the selected chromosomes from one population to the next by exchanging one or more of their subparts. Crossover is used for emulating sexual reproduction of diploid organisms with the aim to inherit and increase the good properties of parents for offspring chromosomes. Mutation operator introduces random perturbation in chromosome structure; it is used for changing chromosomes randomly and introducing new genetic material into the population.

Besides genetic operators, termination criteria are important factor affecting the search process. Widely used termination criteria are, i.e.:

- Reaching optimal solution (which is often hard, if not impossible, to recognize)
- Processing certain number of generations
- Processing certain number of generations without improvement in population

EAs are successful general adaptable concept with good results in many areas. The class of evolutionary techniques consists of more particular algorithms having numerous variants, forged and tuned for specific problem domains. The family of evolutionary algorithms consists of genetic algorithms, genetic programming, evolutionary strategies and evolutionary programming.

7.3.3 Genetic Algorithms

Genetic Algorithms (GA) introduced by John Holland and extended by David Goldberg are wide applied and highly successful EA variant. Basic workflow of original (standard) generational GA (GGA) is (see also Fig. 7.4):

```
1. Define objective function
2. Encode initial population of possible solutions as
   fixed length binary strings and evaluate
   chromosomes in initial population using objective
   function
```

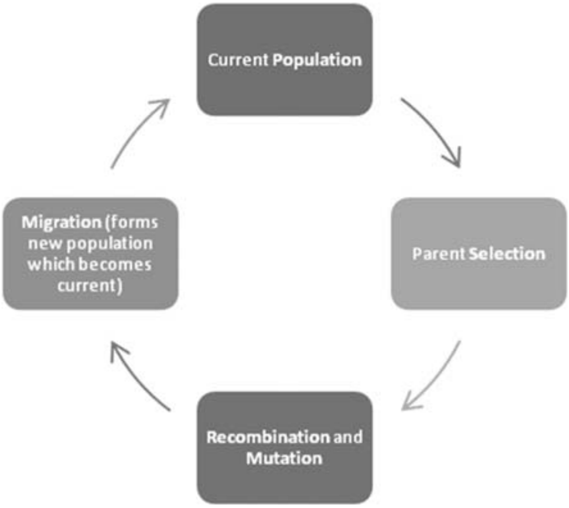

Current Population
Parent Selection
Recombination and Mutation
Migration (forms new population which becomes current)

Fig. 7.4 Iterative phase of genetic algorithm

```
3. Create new population (evolutionary search for
   better solutions)
   a. Select suitable chromosomes for reproduction
      (parents)
   b. Apply crossover operator on parents with
      respect to crossover probability to produce
      new chromosomes (known as offspring)
   c. Apply mutation operator on offspring
      chromosomes with respect to mutation
      probability. Add newly constituted chromosomes
      to new population
   d. Until the size of new population is smaller
      than size of current population go back to (a).
   e. Replace current population by new population
4. Evaluate current population using objective
   function
5. Check termination criteria; if not satisfied go
   back to (3).
```

Many variants of standard generational GA have been proposed. The differences are mostly in particular selection, crossover, mutation and replacement strategy [14]. Different high-level approach is represented by steady-state Genetic Algorithms (SSGA). In GGA, in one iteration is replaced whole population [8] or fundamental part of population [28] while SSGA replace only few individuals at time and never whole population. This method is more accurate model of what happens in the nature and allows exploiting promising individuals as soon as they are created. However, no evidence that SSGA are fundamentally better than GGA was found [28].

7.3.4 Genetic Programming

Genetic Programming by John Koza is referred as special case [28] or an extension [18] to GA. Encoded individuals (chromosomes) have hierarchical structure, unlimited size and they are often modeled as tree structures. So can be modeled mathematical formulas, logical expressions or even whole computer programs (i.e., Lisp programs). Genetic programming is a native tool for modeling and artificial evolution of search queries.

7.4 Evolutionary Techniques and Fuzzy Logic Principles in IRS

Fuzzy theory, as a framework describing formally the concepts of vagueness, imprecision, uncertainty and inconsistency provide interesting extensions to the area of information retrieval. Imprecision and vagueness are present in natural language and take part in real-world human communication. User friendly and flexible advanced IRS should be able to offer user interface for non experienced users allowing natural deployment of these concepts in user-system interaction for more effective information retrieval.

IR models exploiting fuzzy techniques can overcome some of the limitations pointed out in first part of this chapter [15]. They support different grades of document-query relevance, cut inaccuracies and oversimplifications happening during document indexing and introduce the concepts of vagueness and imprecision in query language.

7.4.1 Genetic Algorithms in Information Retrieval

Multiple works have been recently published in the area of IR and search query optimization as this topic becomes increasingly challenging. The use of various evolutionary algorithms was proposed at multiple stages of the information retrieval process. Fan et al. [9] introduced genetic ranking function discovery framework. Nyongesa and Maleki-dizaji [21] used evolutionary interactive evolutionary learning for user modeling.

Yeh et al. [32] described in 2007 a genetic programming for IR ranking function discovery and introduced a set of numerical experiments demonstrating the contribution of proposed method to IR efficiency. Yeh et al. innovatively combined different types of evidences including classical IR content features, structure features, and query independent features to the ranking function.

Several contributions towards evolutionary optimization of search queries were introduced. Kraft et al. [15] used genetic programming to optimize Boolean search queries over a documentary database with an emphasis on the comparison of several

IR effectiveness measures as objective functions. Cordón et al. [6] introduced MOGA-P, an algorithm to deal with search query optimization as a multi-objective optimization problem and compared their approach with several other methods including Kraft's. Yoshioka and Haraguchi [33] introduced query reformulation interface to transform Boolean search queries into more efficient search expressions.

In one of the recent contributions to evolutionary query optimization, A. Aly [1] introduced a genetic algorithm for vector query reformulation based on vector space model of an IR system. The method is based on evolutionary learning of significant terms from search results to modify user queries. Snášel, Nyongesa et al. [27] used similar approach to learn user profiles in IRS based on click-through data and search engine expertness profiling.

This work aims to evaluate evolutionary learning of Boolean search queries in both, traditional crisp Information Retrieval frameworks and advanced fuzzy Information Retrieval systems.

7.4.2 Fuzzy Principles in Information Retrieval

Fuzzy concepts affect most phases of IR process. They are deployed during document indexing, query formulation and search request evaluation. Information retrieval is seen as fuzzy multi-criteria decision making in the presence of vagueness. In general, document is interpreted as a fuzzy set of document descriptors and queries as a composite of soft search constraints to be applied on documents. Document-query evaluation process is based on fuzzy ranking of the documents in documentary collection according to the level of their conformity to the soft search criteria specified via user queries. The document-query matching has to deal with the uncertainty arising from the nature of fuzzy decision making and from the fact that user information needs can be recognized, interpreted and understood only partially. Moreover, the document content is described only in a rough, imperfect way [4].

In the fuzzy enabled IR frameworks, soft search criteria could be specified using linguistic variables. User search queries can contain elements declaring level of partial importance of the search statement elements. Linguistic variables such as "probably" or "it is possible that", can be used to declare the partial preference about the truth of the stated information. The interpretation of linguistic variables is then among the key phases of query evaluation process. Term relevance is considered as a gradual (vague) concept. The decision process performed by the query evaluation mechanism computes the degree of satisfaction of the query by the representation of each document. This degree, called Retrieval Status Value (RSV), is considered as an estimate of the relevance of the document with respect to the query. $RSV = 1$ corresponds to maximum relevance and $RSV = 0$ denotes no relevance. The values within the range $(0, 1)$ correspond to particular level of document relevance between the two extremes 0 and 1 [4].

Possibility theory together with the concept of linguistic variable defined within fuzzy set theory provides a unifying formal framework to formalize the processing of imperfect information [4]. Inaccurate information is inevitably present in information retrieval systems and textual databases applications. The automatically created document representation based on a selection of index terms is invariably incomplete and far worse than document representations created manually by human experts who utilize their subjective theme knowledge when performing the indexing task. Automated text indexing deals with imprecision since the terms are not all fully significant to characterise the document content and their statistical distribution does not reflect their relevance to the information included in the document necessarily. Their significance depends also on the context in which they appear and on the unique personality of the inquirer. During query formulation, users might have only a vague idea of the information they are looking for therefore face difficulties when formulating their information needs by the means of query language of particular IR system. A flexible IRS should be designed to provide detailed and rich representation of documents, sensibly interpret and evaluate soft queries and hence offer efficient information retrieval service in the conditions of vagueness and imprecision [4].

In the next section, Extended Boolean IR model as the representative of fuzzy IR models will be discussed in details. Some other recent fuzzy IR models will be briefly presented.

7.4.3 Extended Boolean IR Model

Fuzzy generalizations of the Boolean model have been defined to extend existing Boolean IRSs without the need to redesign them. Classic Boolean model of IR represents documents as sets of indexed terms. Therefore we can for every term say whether it belongs to the set representing the document (then a weight 1 is assigned to the term for the particular document representation) or not (a weight 0 is assigned). The term weight is either 0 or 1 and multiple occurrences of the term in the document do not affect its internal representation.

Extended Boolean model of IR is based on fuzzy set theory and fuzzy logic. Documents are interpreted as fuzzy sets of indexed terms, assigning to every term contained in the document particular weight from the range of [0, 1] expressing the degree of significance of the term for document representation. Hence documents are modeled more accurately than in classic Boolean IR model. Formal collection description in extended Boolean IR model is shown in Eqs. 7.15 and 7.16.

$$d_i = (t_{i1}, t_{i2}, \ldots, t_{in}), \quad \forall t_{ij} \in \{0, 1\} \tag{7.15}$$

$$D = \begin{pmatrix} t_{11} & t_{12} & \cdots & t_{1n} \\ t_{21} & t_{22} & \cdots & t_{2n} \\ \vdots & \vdots & & \vdots \\ t_{m1} & t_{m2} & \cdots & t_{mn} \end{pmatrix} \tag{7.16}$$

Next new feature of extended Boolean IR model is fuzzy extension of query language aiming at providing apparatus to express more flexible and accurate search requests. Two techniques are being used for query enhancement query term weighting using numeric weights or linguistic variables and Boolean conjunction parameterization for expressing relationships among the extremes of AND, OR, NOT, etc. [15]. Choosing appropriate indexing procedure is essential for exploitation of extended Boolean IR model benefits. Internal documentary collection model should be as accurate as possible snapshot of the collection of textual documents in natural language and at the same time a basis for efficient and practical search. Fuzzy indexing function is defined as shown in Eq. 7.17, where D stands for the set of all documents and T for set of all indexed terms.

$$F : D \times T \to [0,1] \tag{7.17}$$

Kraft in [15] used Salton's *TFIDF*$_t$ indexing formula introduced for VSM as textual document indexing mechanism in extended Boolean IR model. Query language is in extended Boolean model of IR upgraded by the possibility of weighting query terms in order to express different importance of those in search request and by weighting (parameterizing) aggregation operators to soften or blur their impact on query evaluation [7, 16]. Consider Q to be the set of user queries over a collection then the weight of term t in query q is denoted as $a(q,t)$ satisfying $a : Q \times T \to [0,1]$. To evaluate atomic query of one term, stating therefore only one search criterion, will be used function $g : [0,1] \times [0,1] \to [0,1]$. The value of $q(F(q,t),a)$ is called retrieval status value (RSV). For RSV enumeration is crucial the interpretation of query term weight a. The most used interpretations are to see query term weight as importance weight, threshold or ideal document description [7,16]. The theorems for RSV evaluation in the case of importance weight interpretation and threshold interpretation are shown in Eqs. 7.18 and 7.19 respectively [7, 16], where $P(a)$ and $Q(a)$ are coefficients used for tuning the threshold curve. An example of $P(a)$ and $Q(a)$ could be as follows: $P(a) = \frac{1+a}{2}$ and $Q(a) = \frac{1+a^2}{4}$. The RSV formula in Eq. 7.19 is illustrated in Fig. 7.5a. Adopting the threshold interpretation, an atomic query containing term t of the weight a is a request to retrieve documents having $F(d,t)$ equal or greater to a. For documents satisfying this condition will be rated with high RSV and contrariwise documents having $F(d,t)$ smaller than a will be rated with small RSV.

$$RSV = \begin{cases} \min(a, F(d,t)) & \text{if t is operand of OR} \\ \max(1-a, F(d,t)) & \text{if t is operand of AND} \end{cases} \tag{7.18}$$

$$RSV = \begin{cases} P(a)\frac{F(d,t)}{a} & \text{pro } F(d,t) < a \\ P(a) + Q(a)\frac{F(d,t)-a}{1-a} & \text{pro } F(d,t) \geq a \end{cases} \tag{7.19}$$

Query term weight a can be understood as ideal document term weight prescription. In that case, RSV will be evaluated according to Eq. 7.20, enumerating the distance between $F(d,t)$ and a in a symmetric manner as shown in Fig. 7.5b. This

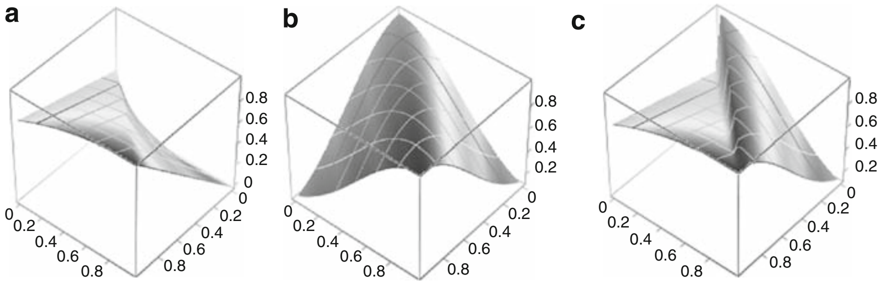


Fig. 7.5 Graphic representation of the three RSV functions. (**a**) PR-landscape for Eq. 7.19 (**b**) PR-landscape for Eq. 7.20 (**c**) PR-landscape for Eq. 7.21

means that a document with lower term weight will be rated with the same RSV as document with higher term weight, considering the same differences. Asymmetric version of Eq. 7.20 is shown in Eq. 7.21 and illustrated in Fig. 7.5c.

$$RSV = e^{K \cdot (F(d,t)-a)^2} \tag{7.20}$$

$$RSV = \begin{cases} e^{K \cdot (F(d,t)-a)^2} & \text{pro } F(d,t) < a \\ P(a) + Q(a)\frac{F(d,t)-a}{1-a} & \text{pro } F(d,t) \geq a \end{cases} \tag{7.21}$$

Single weighted term is basic element of fuzzified Boolean query. Aggregation operators concatenating query elements into more flexible and powerful search expressions might be weighted as well. The operator weight interpretation is another key part of fuzzy Boolean query evaluation. In general, various T-norm and T-conorm pairs might be used for fuzzy generalization of AND and OR operators while evaluating NOT as fuzzy complement. Operator weights are in these cases handled in the same manner as query term weight achieving higher flexibility and expressiveness of search expressions. Nevertheless, such approach does not reduce the complexity of Boolean logic needed to use the queries efficiently [16]. Alternatively, new definitions of aggregation operators for fuzzy queries have been introduced. Vague relationship among selection criteria is expressed using linguistic quantifiers such as all, most of, at least n, introducing blurred behaviour between AND and OR and allowing easier query formulation [7, 16].

7.4.4 *Fuzzy IR Effectiveness Evaluation*

When evaluating effectiveness of an IR system, precision and recall are among the most popular performance measures serving as a basis for numerous derived indicators such as effectiveness E or F-score F. For the enumeration of precision and recall in the framework of fuzzy IR systems cannot be used crisp precision and recall as specified in Eq. 7.10. New definitions were proposed on the basis of Zadehs cardinality (see Eq. 7.22; the function $\mu_A(x)$ assigns to every item x its fuzzy weight in fuzzy set A) as shown in Eqs. 7.23 and 7.24 [17].

$$card(A) = \|A\| = \sum_{x \in A} \mu_A(x) \tag{7.22}$$

$$\rho(X|Y) = \begin{cases} \frac{\|X \cap Y\|}{\|Y\|} & \|Y\| \neq 0 \\ 1 & \|Y\| = 0 \end{cases} \tag{7.23}$$

$$P = \rho(REL|RET) \text{ and } R = \rho(RET|REL) \tag{7.24}$$

7.5 Experimental Evaluation

This section describes computational experiments performed in order to validate presented search improvement and user modeling techniques. First, the experiments on Genetic Programming for query optimization will be presented Second, an online experiment comparing the efficiency of web search with and without query optimization and user profiling will be introduced.

7.5.1 Offline Query Optimization Experiments

A series of computer experiments was conducted in order to evaluate proposed GP enabled IR framework in both, crisp Boolean IR model and fuzzified Extended Boolean IR model [22, 23, 25, 27]. Experiments were executed using data taken from the LISA[1] collection. The collection was indexed for both Boolean IR and Extended Boolean IR systems, using Salton's indexing function based on normalized term frequency and normalized inverse document frequency in the latter case. Indexed collection contained 5999 documents and 18442 unique indexed terms.

Genetic Programming was used to evolve Boolean search queries. Boolean expressions were parsed and encoded into tree like chromosomes (see Fig. 7.6). Genetic operators were applied on nodes of the tree chromosomes. Several parameters were fixed for all experiments:

- Mutation probability = 0.2
- Crossover probability = 0.8
- Maximum number of generations = 1000
- Population of 70 individuals (queries)

We have used two scenarios for initial population. In the first case, all queries in initial population were generated randomly. In the second scenario, three better

[1] Available at: http://www.dcs.gla.ac.uk/idom/ir_resources/test_collections/

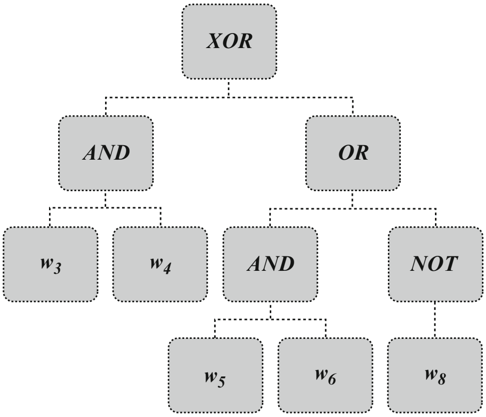


Fig. 7.6 Search query $(w_3 \textit{ and } w_4) \textit{ xor } ((w_5 \textit{ and } w_6) \textit{ or not } w_8)$ encoded for GP

Table 7.1 Summary for experiments results

IR model	User query	Initial population enhancement
BIRM	((“EXTREMELY” AND “POOR”) OR “FUNDS”)	“FUNDS” OR “BIBLIOGRAPHIC” “EXTREMELY” AND “INNOVATORS” NOT (“POOR” XOR “FUNDS”)
EBIRM	((“EXTREMELY”:0.94 AND “POOR”:0.50) OR:0.50 “FUNDS”:0.90)	“FUNDS”:0.9 OR “BIBLIOGRAPHIC”:0.8 “EXTREMELY”:0.3 AND “INNOVATORS” NOT (“POOR” XOR:0.03 “FUNDS”:0.5)

ranked queries, created by the experiment administrators, were added to the initial population. Two selection strategies were investigated: elitary selection choosing parents among the best ranked individuals and probabilistic selection implementing the roulette wheel selection algorithm. Two mutation strategies were under investigation. Single point mutation performs random perturbation of one gene (i.e., one node) of the query chromosome and each point mutation attempts to apply mutation operator on every gene in the chromosome. Mutation is implemented as replacement of the node by an equivalent. This means that OR might be replaced by XOR and AND. NOT operator might be inserted or removed.

A user query was used to mark documents in the collection with some relevance degree. The user query (or its equivalent) represents in laboratory conditions desired output of the optimization algorithm. The experiments were conducted in crisp and fuzzy laboratory Information Retrieval framework. The crisp IR framework was marked as Boolean Information Retrieval Model (BIRM) and the fuzzy IR framework was denoted as Extended Boolean Information Retrieval Model (EBIRM). Due to the stochastic character of GP process, all experiments were executed several times and mean experimental results evaluated.

Table 7.1 lists the user query and better ranked queries injected into initial population in some experiments.

Tables 7.2 and 7.3 summarize the experimental results obtained for different scenarios. Experiments are labeled with the following flags: single point mutation **I**,

Table 7.2 Summary of experimental results in BIRM

Scenario	Precision	Recall	F-score
REI	0.04699	0.089552	0.0486915
REC	0.040411	0.11194	0.0621065
RPI	0.064519	0.074627	0.069205
RPC	0.053471	0.119403	0.0689775
SEI	1	0.985075	0.992481
SEC	1	0.985075	0.992481
SPI	1	0.985075	0.992481
SPC	1	0.985075	0.992481

Table 7.3 Summary of experimental results in EBIRM

Scenario	Precision	Recall	F-score
REI	0.078706	0.027165	0.04039
REC	0.078706	0.027165	0.04039
RPI	0.0765365	0.0760845	0.0754315
RPC	0.163975	0.0389625	0.060813
SEI	0.9933375	0.9045225	0.9454495
SEC	0.993873	0.968469	0.9810005
SPI	0.9138465	0.9696315	0.940731
SPC	0.9965815	0.968436	0.9823045

each point mutation **C**, elitism **E**, probabilistic selection **P**, seeded queries **S** and random initial population **R**. The results were taken as an average for fitness values for precision, recall and F-score.

From the experiments with Boolean queries we conclude the following results: Genetic Programming succeeded in optimization of Boolean and extended Boolean search queries. Crucial for the optimization process was the quality of initial population. For successful optimization, initial population must contain at least some quality queries pointing to documents related to user needs. This fact was especially significant when optimizing extended queries with weighted terms and operators. Weight assessment rapidly increases search domain of the problem.

F-score fitness was preferred as a measure combining precision and recall into one value by the means of information retrieval and therefore simplifying query optimization from multi-objective to a single-objective task. Figures 7.7–7.12 illustrate the improvements of F-score, *Precision* and *Recall* of the optimized queries in different experimental cases. Figures 7.13 and 7.14 respectively show the significant differences of optimization results when using random initial population and seeded initial population.

7.5.2 Online Experiments

To evaluate proposed user modeling method and search optimization technique, a set of experiments comparing search experience in different cases with and

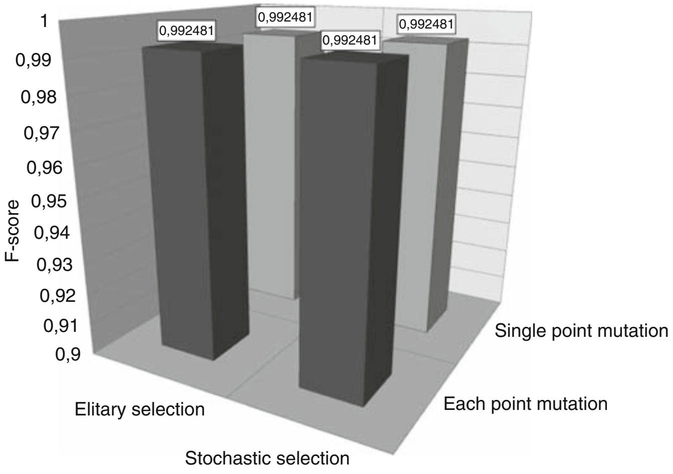


Fig. 7.7 The comparison of achieved F-score for different algorithm setups in BIRM with seeded initial population

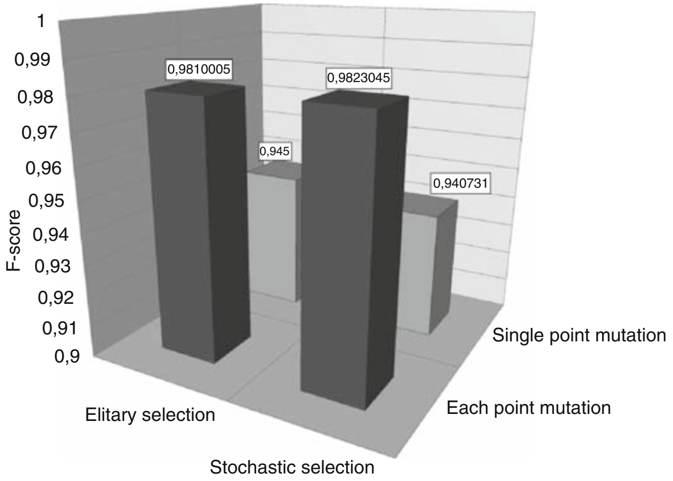


Fig. 7.8 The comparison of achieved F-score for different algorithm setups in EBIRM with seeded initial population

without query optimization support was designed and performed. Number of emitted queries, average click rate and length of mouse trajectory created before retrieving satisfactory information were traced as an objective measure of the efficiency of search task.

Intentionally, user queries were during experiments constructed from simple to more complex. In order to create initial user profile for optimized search, the participants performed common search activities focused on evolutionary algorithms

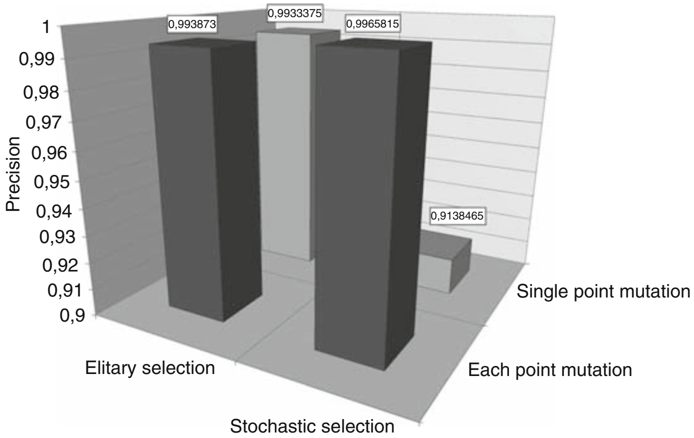


Fig. 7.9 The comparison of achieved precision for different algorithm setups in EBIRM with seeded initial population

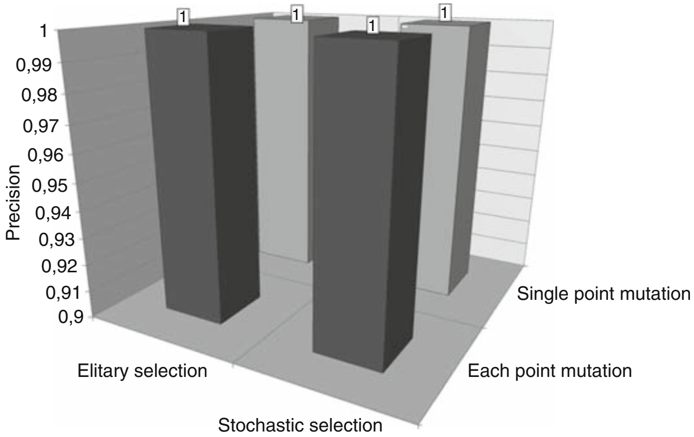


Fig. 7.10 The comparison of achieved precision for different algorithm setups in BIRM with seeded initial population

and optimization techniques. The resulting profile snapshot contained 1,044 terms in 120 documents and 25 queries such as:

- "genetic" AND "algorithm"
- "genetic" AND "operator"
- "dynamic" AND "optimization"
- "dynamic" AND "optimization" AND "task" AND NOT "dbm"

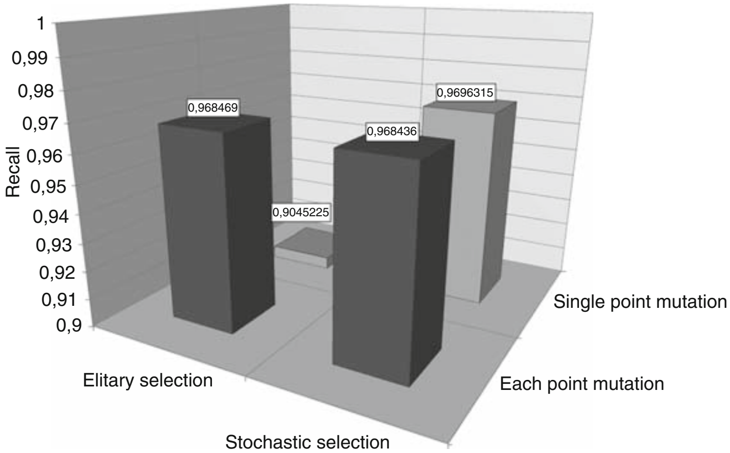


Fig. 7.11 The comparison of achieved recall for different algorithm setups in EBIRM with seeded initial population

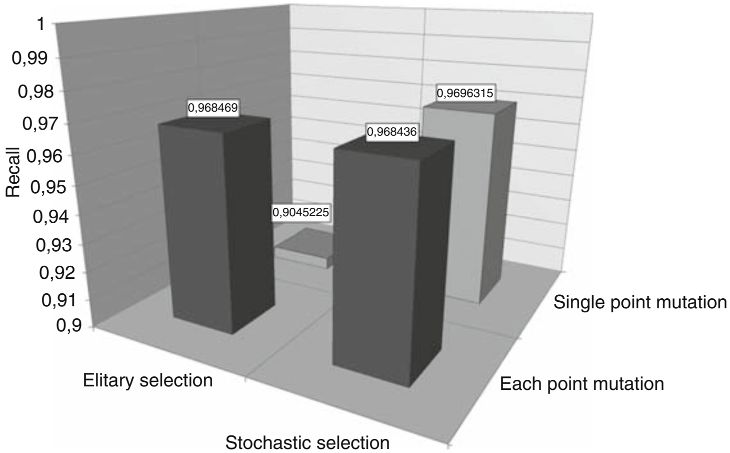


Fig. 7.12 The comparison of achieved recall for different algorithm setups in BIRM with seeded initial population

7.5.2.1 Online Experiments and Discussion

In the first experiment summarized in Table 7.4, the desired information should describe known evolutionary techniques for dynamic optimization problems which was searched during profiling phase. The submitted user query was

- “genetic” AND “algorithm”

In the first online experiment, the optimization process managed to modify query towards previously observed area of interest and the same document was retrieved

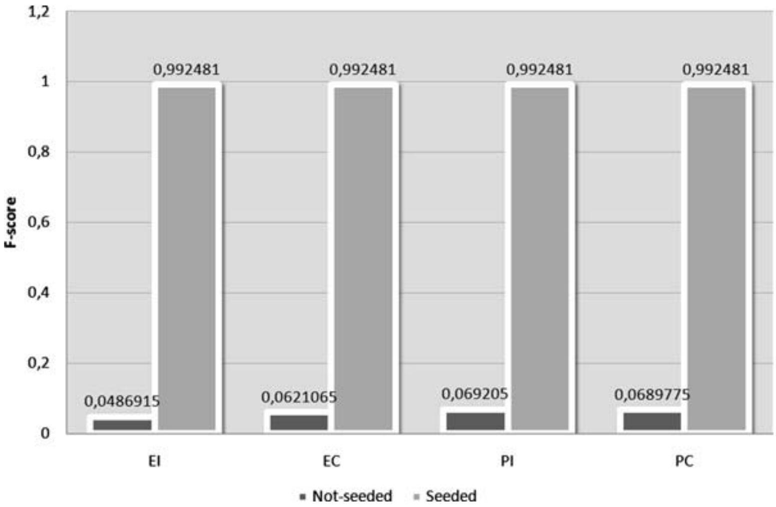


Fig. 7.13 The comparison of achieved F-score for used algorithm setups with seeded (S) and random (R) initial population in BIRM

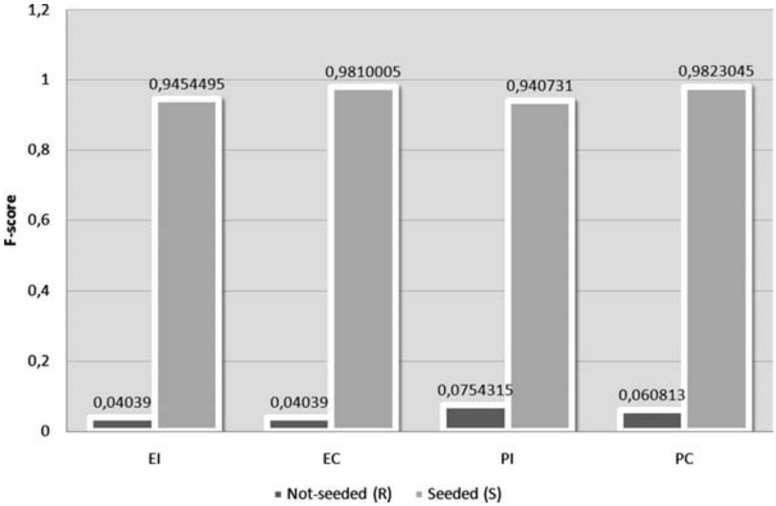


Fig. 7.14 The comparison of achieved F-score for used algorithm setups with seeded (S) and random (R) initial population in EBIRM

Table 7.4 Summary of first online experiment

	Search type	
Criterion	Non-optimized	Optimized
No. of queries	4	1
No. of clicks	55	7
Mouse trajectory [m]	12	1,8

Table 7.5 Summary of second online experiment

	Search type	
Criterion	Non-optimized	Optimized
No. of queries	5	2
No. of clicks	22	14
Mouse trajectory [m]	4,6	2,20

Table 7.6 Summary of third online experiment

	Search type	
Criterion	Non-optimized	Optimized
No. of queries	4	4
No. of clicks	92	104
Mouse trajectory [m]	14,44	17,21

in just one step. This was an expected result since the user profile was created when searching for exactly this information. Optimized search discovered the same document as un-optimized.

Second online experiment, as shown in Table 7.5, features search for new information from the same area, adaptive optimization via genetic programming. The submitted user queries were

- "adaptive" AND "optimization"
- "adaptive" AND "optimization" AND "genetic"

When searching information from an area covered by the user model, the optimization process helps to retrieve satisfactory information in fewer steps than non-optimized search. Different search types retrieved different documents but both documents were by the inquirer accepted as satisfactory.

The third online experiment covers situation when the searched information belongs to different area than the one captured by the user profile. The searched information is weather forecast for Ostrava, Czech Republic with aim to daylight period. The submitted queries were:

- "weather" AND "daylight"
- "weather" AND "daylight" AND "Ostrava"
- "weather" AND "daylight" AND "Ostrava" AND "year"
- "daylight" AND "Ostrava"

In the third online experiment, there was no improvement observed. The same number of queries was issued and the number of clicks and mouse trajectory were slightly higher for optimized search. This means, that when the search aims out of the known web region, the optimization process brings no benefits.

The performed experiments suggest that the method is able to improve (speed up) search in the areas covered by the user profile (Tables 7.4 and 7.5). It is not able to improve search process when aiming to totally new area of interest, however in real-life deployment; the application would be able to learn from every submitted query while in presented set of experiments the profile was constant (Table 7.6).

7.6 Conclusion

The area of information retrieval faces today enormous challenges. The information society in the age of Internet excels in producing huge amounts of data and it is often complicated to retrieve information retrieved in such data sources. Decades ago, sophisticated techniques and algorithms forming information retrieval systems were designed to handle document collections available at that time. Information retrieval systems have gone over an intensive evolution to satisfy increasing needs of growing data bases. In their mature form, they are still present in the heart of Internet search engines, one of the key communication hubs of our society.

Internet search allows exploitation of large amount of knowledge available in the ubiquitous multitude of data. Information search is one of the most important e-activities. The IR systems, despite their advanced features, need revision and improvement in order to achieve better performance and provide inquirer with more satisfactory answers. Aiming to achieve better performance, more flexible models and techniques are requested. Fuzzy set framework has been proved as suitable formalism for modeling and handling vagueness and imprecision, the hot topics of information retrieval. Numerous researches considering various applications of fuzzy set technology have been initiated and conducted, some recent summarized in this chapter. The deployment of fuzzy techniques in all phases of IR has brought improvement of IR results and therefore increases user satisfaction. Lotfi Zadeh once called fuzzy technology computing with words. Information retrieval performs real world computation with words for decades. The symbiosis of these two progressive areas promises exciting results for the coming years.

Evolutionary techniques are an excellent tool to extract non-explicit information from data. Their unique ability to estimate, evolve and improve can be used to model Internet search user. Implicit data, such as the click-stream, produced during the web browsing activities could be exploited to keep track of the preferences of every single user. Such model is accurate, flexible, and can be well exploited for query optimization. Simultaneous deployment of fuzzy set techniques for better document modeling and genetic algorithms for query optimization brings a significant contribution to the ultimate goal of web search: bringing knowledge to man.

References

1. Abdelmgeid A. Aly. Applying genetic algorithm in query improvement problem. *Int. Journal on Information Technologies and Knowledge*, 1(12):pp. 309–316, 2007.
2. Nicholas J. Belkin and W. Bruce Croft. Information filtering and information retrieval: two sides of the same coin? *Communications of the ACM*, 35(12):pp. 29–38, December 1992.
3. Ulrich Bodenhofer. *Genetic Algorithms: Theory and Applications*. Lecture notes, Fuzzy Logic Laboratorium Linz-Hagenberg, Winter 2003/2004.
4. Gloria Bordogna and Gabriella Pasi. Modeling vagueness in information retrieval. pages 207–241, 2001.
5. Pia Borlund and Peter Ingwersen. Measures of relative relevance and ranked half-life: performance indicators for interactive IR. In *SIGIR'98*, pages 324–331, Melbourne, Australia, August 1998.

6. Oscar Cordón, Félix de Moya, and Carmen Zarco. Fuzzy logic and multiobjective evolutionary algorithms as soft computing tools for persistent query learning in text retrieval environments. In *IEEE International Conference on Fuzzy Systems 2004*, pages 571–576, Budapest, Hungary, 2004.
7. Fabio Crestani and Gabriella Pasi. Soft information retrieval: Applications of fuzzy set theory and neural networks. In N. Kasabov and R. Kozma, editors, *Neuro-Fuzzy Techniques for Intelligent Information Systems*, pages 287–315. Springer Verlag, Heidelberg, DE, 1999.
8. Mehrdad Dianati, Insop Song, and Mark Treiber. An introduction to genetic algorithms and evolution strategies. Technical report, University of Waterloo, Ontario, N2L 3G1, Canada, July 2002.
9. Weiguo Fan, Michael D. Gordon, and Praveen Pathak. A generic ranking function discovery framework by genetic programming for information retrieval. *Inf. Process. Manage*, 40(4):pp. 587–602, 2004.
10. Weiguo Fan, Michael D. Gordon, Praveen Pathak, Wensi Xi, and Edward A. Fox. Ranking function optimization for effective web search by genetic programming: An empirical study. In *HICSS*, 2004.
11. E. Greengrass. *Information retrieval: A survey*. DOD Technical Report TR-R52-008-001, 2001.
12. Stephen P. Harter. Psychological relevance and information science. *JASIS*, 43(9):602–615, 1992.
13. Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2002.
14. Gareth Jones. Genetic and evolutionary algorithms. In Paul von Rague, editor, *Encyclopedia of Computational Chemistry*. John Wiley and Sons, 1998.
15. D. H. Kraft, F. E. Petry, B. P. Buckles, and T. Sadasivan. Genetic Algorithms for Query Optimization in Information Retrieval: Relevance Feedback. In E. Sanchez, T. Shibata, and L.A. Zadeh, editors, *Genetic Algorithms and Fuzzy Logic Systems*, Singapore, 1997. World Scientific.
16. Donald H. Kraft, Gloria Bordogna, and Gabriella Pasi. Fuzzy set techniques in information retrieval. In J. C. Bezdek, D. Didier, and H. Prade, editors, *Fuzzy Sets in Approximate Reasoning and Information Systems*, volume 3 of *The Handbook of Fuzzy Sets Series*, pages 469–500, MA, 1999. Kluwer Academic Publishers.
17. Henrik L. Larsen. Retrieval evaluation. In *Modern Information Retrieval course*. Aalborg University Esbjerg, 2004.
18. Gondy Leroy, Ann M. Lally, and Hsinchun Chen. The use of dynamic contexts to improve casual internet searching. *ACM Transactions on Information Systems*, 21(3):pp. 229–253, 2003.
19. Robert M. Losee. When information retrieval measures agree about the relative quality of document rankings. *Journal of the American Society of Information Science*, 51(9):pp. 834–840, 2000.
20. Melanie Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, 1996.
21. H. O. Nyongesa and S. Maleki-Dizaji. User modeling using evolutionary interactive reinforcement learning. *Inf. Retr.*, 9(3):343–355, 2006.
22. Suhail Owais, Pavel Kromer, Vaclav Snasel, Dusan Husek, and Roman Neruda. Implementing GP on optimizing both boolean and extended boolean queries in IR and fuzzy IR systems with respect to the users profiles. In Gary G. Yen, Lipo Wang, Piero Bonissone, and Simon M. Lucas, editors, *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, pages 5648–5654, Vancouver, BC, Canada, 6–21 July 2006. IEEE Computer Society.
23. Suhail S. J. Owais, Pavel Krömer, and Václav Snášel. Evolutionary Learning of Boolean Queries by Genetic Programming. In Johann Eder, Hele-Mai Haav, Ahto Kalja, and Jaan Penjam, editors, *ADBIS Research Communications*, volume 152 of *CEUR Workshop Proceedings*, pages 54–65. CEUR-WS.org, 2005.
24. Suhail S. J. Owais, Pavel Krömer, and Václav Snášel. Query Optimization by Genetic Algorithms. In Karel Richta, Václav Snásel, and Jaroslav Pokorný, editors, *DATESO*, volume 129 of *CEUR Workshop Proceedings*, pages 125–137. CEUR-WS.org, 2005.

25. Suhail S. J. Owais, Pavel Krömer, and Václav Snášel. Implementing gp on optimizing boolean and extended boolean queries in irs with respect to users profiles. In H. M. A. Fahmy, A. M. Salem, M. W. El-Kharashi, and A. M. B. El-Din, editors, *Proceedings of the 2006 International Conference on Computer Engineering & Systems (ICCES06)*, pages 412–417, Cairo, Egypt, November 2006. IEEE Computer Society. ISBN: 1-4244-0272-7.
26. Gerard Salton and Chris Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):pp. 513–523, 1988.
27. Václav Snášel, Pavel Krömer, Suhail S. J. Owais, Henry O. Nyongesa, and S. Maleki-Dizaji. Evolving web search expressions. In *Third International Conference on Natural Computation (ICNC'07)*, volume 4, pages 532 – 538, Haikou, Hainan, China, August 2007. IEEE Computer Society Press. ISBN: 0-7695-2875-9, IEEE CS Order Number: P2875, Library of Congress: 2007926988.
28. H. A. R. Townsend. *Genetic Algorithms - A Tutorial*, 2003.
29. P. Vakkari and N Hakala. Changes in relevance criteria and problem stages in task performance. *Journal of Documentation*, 5(56):540562, 2000.
30. Gui-Rong Xue, Hua-Jun Zeng, Zheng Chen, Yong Yu, Wei-Ying Ma, WenSi Xi, and WeiGuo Fan. Optimizing web search using web click-through data. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 118–126, New York, NY, USA, 2004. ACM Press.
31. R. R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision-making. In D. Dubois, H. Prade, and R. R. Yager, editors, *Readings in Fuzzy Sets for Intelligent Systems*, pages 80–87. Kaufmann, San Mateo, CA, 1993.
32. Jen-Yuan Yeh, Jung-Yi Lin, Hao-Ren Ke, and Wei-Pang Yang. Learning to rank for information retrieval using genetic programming. In *SIGIR*, 2007.
33. Masaharu Yoshioka and Makoto Haraguchi. An Appropriate Boolean Query Reformulation Interface for Information Retrieval Based on Adaptive Generalization. In *WIRI*, pages 145–150, 2005.