# Optimizing Information Retrieval Using Evolutionary Algorithms and Fuzzy Inference System

Václav Snášel[1], Ajith Abraham[2], Suhail Owais[3], Jan Platoš[1], and Pavel Krömer[1]

[1] 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 {vaclav.snasel, jan.platos, pavel.kromer.fei}@vsb.cz

[2] Center of Excellence for Quantifiable Quality of Service, Norwegian University of Science and Technology, O.S. Bragstads plass 2E, N-7491 Trondheim, Norway ajith.abraham@ieee.org

[3] Information Technology, Al-Balqa' Applied University - Ajloun University College, P.O. Box 6, JO 26810 Ajloun, Jordan suhailowais@yahoo.com

## 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 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 mis-

takes 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 modelling 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.

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 introduce genetic and fuzzy oriented approach to these tasks 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 article is organized as follows: In Section 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 3 summarizes the usage of Evolutionary Computation, Genetic Algorithms, Genetic Programming and its application to information retrieval tasks. In Section 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 5 and finally some conclusions are also provided.

## 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 [6]. The rest of this article 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. 1 [6].

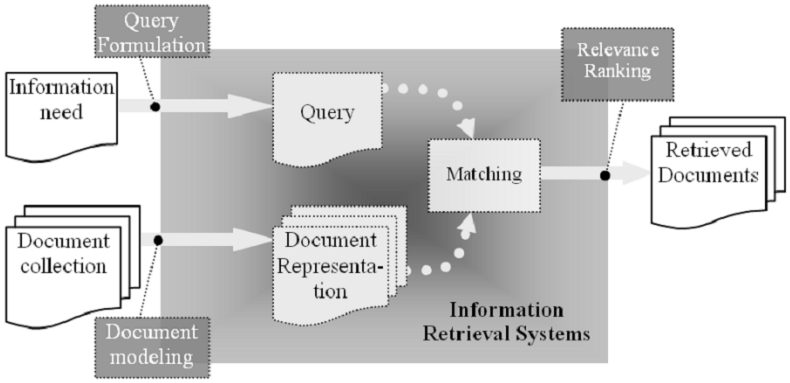



**Fig. 1.** An Information Retrieval System

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 [6]. 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 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. 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.

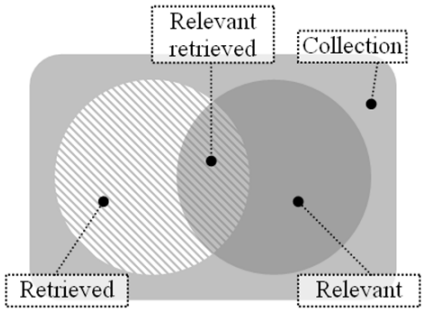



**Fig. 2.** Documents in collection classified in response to a query

In the previous paragraphs were documents classified against 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 [11]. Subjective relevance is user-centric and deals with fitness for use of the retrieved information [27]. Subjective relevance involves intellectual interpretation by human assessors or users [4] 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 modelling and user profiling.

### 2.1 Information Retrieval models

An *IR model* is a formal background defining internal document representation, query language and document-query matching mechanism. Conse-

quently, 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 [6, 14].

### Boolean IR model

Boolean IR model belongs to the oldest but till nowadays widely used information retrieval models [6, 1]. 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 [1]. 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 contained 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 Equations 1 and 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.

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

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

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 [1, 6, 10]. 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 [10].

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 [13]. 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.

### Vector space model

Vector space model (VSM) is based on interpretation of both, documents and queries, as points in a multidimensional document space [6, 10]. The dimension of the document space is given by the number of indexed terms in the documentary collection. Every term has 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 [6, 1, 14]. Formal description of VSM is almost identical to the description of Boolean model as provided in Equations 1 and 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 (Equation 3).

$$q = (t_{q1}, t_{q2}, \ldots, t_{qn}), \forall t_{qj} \in R \tag{3}$$

In Boolean IR, indexing procedure was due to the simplicity of internal document representation trivial task. In VSM is 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 [10]. 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 [24]. Consider normalised term frequency of term $t$ in document $d$ shown in Equation 4 as

the ratio of frequency of each term in the document to the maximum term frequency in that document. Therefore, the greater the frequency of particular term in the document, the greater the normalized frequency of such term in the document.

$$f_{dt} = \frac{freq(t,d)}{\max(freq(t_i,d))} \tag{4}$$

Normalized inverse document frequency, defined as shown in Equation 5, reflects the distribution of given term among all documents in the collection. The rarer is the term in the scope of whole collection, the greater is its inverse document frequency. $N$ stands in Equation 5 for 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. Finally, the weight of term $t$ in document $d$ according to $TFIDF_t$ is defined in Equation 6.

$$IDF_t = g(\log \frac{N}{N_t}) \tag{5}$$

$$F(d,t) = f_{dt} \cdot IDF_t \tag{6}$$

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 [10]. 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 Equation 7.

$$F(q,t) = (\frac{1}{2} + \frac{f_{qt}}{2}) \tag{7}$$

The document-query matching procedure is 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 [10]. Among the most popular are scalar product (8) and cosine measure (9) that can be interpreted as an angle between the query vector and document vector in $m$-dimensional document space.

$$Sim(q,d_i) = \sum_{j=1}^{m} t_{qj} \cdot t_{ij} \tag{8}$$

$$Sim(q,d_i) = \frac{\sum_{j=1}^{m} t_{qj} \cdot t_{ij}}{\sqrt{\sum_{j=1}^{m} t_{qj}^2 \cdot \sum_{j=1}^{m} t_{ij}^2}} \tag{9}$$

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 [10]. 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.

### 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 (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 $F$ [17] as shown in Equation 11.

$$P = \frac{|REL \cap RET|}{|RET|} \qquad R = \frac{|REL \cap RET|}{|REL|} \tag{10}$$

$$E = 1 - \frac{2}{\frac{1}{P} + \frac{1}{R}} \qquad F = 1 - E = \frac{2PR}{P + R} \tag{11}$$

### 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 is some knowledge about ones 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 [22].

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 ex-tension, query reformulation for other techniques improving the 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 [10]. User profile can be also exploited for document re-ranking according to individual preferences [22]. 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 [8].

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 [5]. 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 [9].

## 3 Evolutionary Computation

Evolutionary algorithms (EA) belongs to a family of iterative stochastic search and optimization methods based on mimicking successful optimization strategies observed in nature [7, 12, 18, 2]. The essence of EAs lies in the emulation of Darwinian evolution utilizing the concepts of Mendelian inheritance for the use in computer science and applications [2]. Together with fuzzy sets, neural net-works and fractals, evolutionary algorithms are among the fundamental members of the class of soft computing methods.

EA 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 [12]. In the following sections

will be introduced general principles common for all methods belonging to the class of evolutionary algorithms.

### 3.1 Evolutionary Search Process

For the purpose of EAs, a proper encoding representing solutions of given problem as en-coded chromosomes suitable for evolutionary search process, is necessary. Finding proper en-coding 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.

### 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.

### 3.3 Genetic Algorithms

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:

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
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 [12]. Different high-level approach is represented by steady-state Genetic Algorithms (SSGA). In GGA, in one iteration is replaced whole population [7] or fundamental part of population [26] 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 [26].

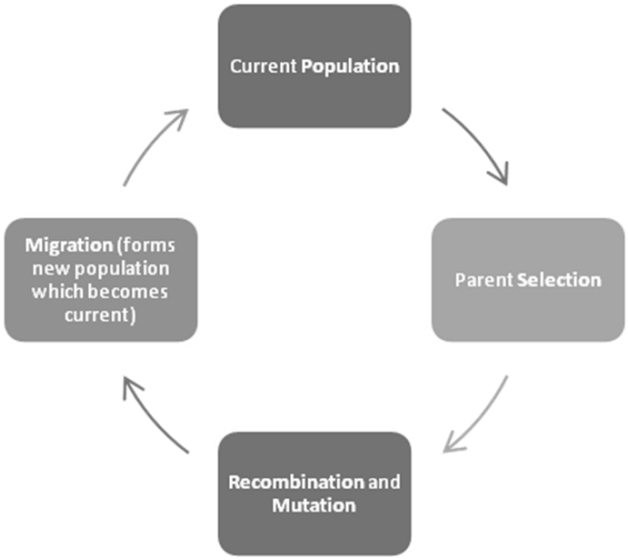



**Fig. 3.** Iterative phase of Genetic Algorithm

### 3.4 Genetic Programming

Genetic Programming by John Koza is referred as special case [26] or an ex-tension [16] to GA. Encoded individuals (chromosomes) have hierarchical structure, unlimited size and they are often modelled as tree structures. So can be modelled mathematical formulas, logical expressions or even whole computer programs (i.e. Lisp programs). Genetic programming is a native tool for modelling and artificial evolution of search queries.

## 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 article [13]. 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.

### 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. [8] introduced genetic ranking function discovery framework. Nyongesa and Maleki-dizaji [19] used evolutionary interactive evolutionary learning for user modelling.

Several contributions towards evolutionary optimization of search queries were introduced. Kraft et al. [13] 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. Cordn et al. [5] 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 [28] introduced query reformulation interface to transform Boolean search queries into more efficient search expressions.

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

### 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 [3].

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 [3].

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 [3]. 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 [3].

In the following, 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.

### 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 repre-senting the document (then a weight 1 is assigned to the term for the particular document repre-sentation) or not (a weight 0 is assigned). The term weight is either 0 or 1 and multiple occur-rences 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 modelled more accurately than in classic Boolean IR model. Formal collection description in extended Boolean IR model is shown in Equations 12 and 13.

$$d_i = (t_{i1}, t_{i2}, \ldots, t_{in}), \forall t_{ij} \in \{0, 1\} \tag{12}$$

$$D = \begin{pmatrix} t_{11} & t_{12} & \cdots & t_{1n} \\ t_{21} & t_{22} & \cdots & t_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ t_{m1} & t_{m2} & \cdots & t_{mn} \end{pmatrix} \tag{13}$$

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. [13]. 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 Equation 14, where D stands for the set of all documents and T for set of all indexed terms.

$$F : D \times T \rightarrow [0, 1] \tag{14}$$

Kraft in [13] 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 [6, 14]. 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 $a : Q \times T \rightarrow [0, 1]$. To evaluate atomic query of one term, stating therefore only one search criterion, will be used function $g : [0, 1] \times [0, 1] \rightarrow [0, 1]$. 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 [6, 14]. The theorems for RSV evaluation in the case of importance weight interpretation and threshold interpretation are shown in Equations 15 and 16 respectively [14, 6], 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 Equation 16 is illustrated in Fig. 4a. 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.

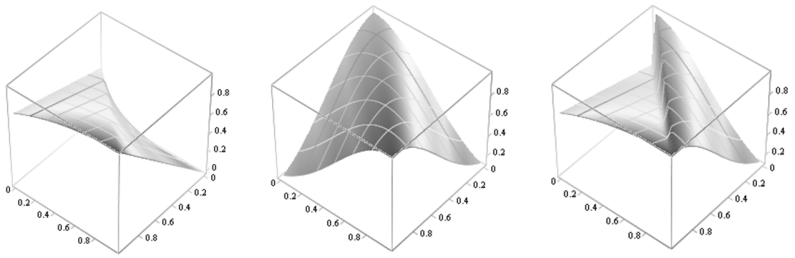
$$RSV = \begin{cases} \min(a, F(d,t)) & \text{if t is operand of OR} \\ \max(1-a, F(d,t)) & \text{if t is operand of AND} \end{cases} \tag{15}$$

$$RSV = \begin{cases} P(a)\frac{F(d,t)}{a} & \text{pro } F(d,t) < a \\ P(a) + Q(a)\frac{F(d,t)-a}{1-a} & \text{pro } F(d,t) \geq a \end{cases} \tag{16}$$

Query term weight a can be understood as ideal document term weight prescription. In that case, RSV will be evaluated according to Equation 17, enumerating the distance between $F(d,t)$ and a in a symmetric manner as shown in Fig. 4b. This 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 Equation 17 is shown in Equation 18 and illustrated in Fig. 4c.

$$RSV = e^{K\cdot(F(d,t)-a)^2} \tag{17}$$

$$RSV = \begin{cases} e^{K\cdot(F(d,t)-a)^2} & \text{pro } F(d,t) < a \\ P(a) + Q(a)\frac{F(d,t)-a}{1-a} & \text{pro } F(d,t) \geq a \end{cases} \tag{18}$$



(a) (b) (c)

**Fig. 4.** Graphic representation of the three RSV functions

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 [14]. 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 [6, 14].

### 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 Equation 10. New definitions were proposed on the basis of Zadehs cardinality as shown in Equations 19 and 20 [15].

$$\rho(X|Y) = \begin{cases} \frac{\|X \cap Y\|}{\|Y\|} & \|Y\| \neq 0 \\ 1 & \|Y\| = 0 \end{cases} \tag{19}$$

$$P = \rho(REL|RET) \text{ a } R = \rho(RET|REL) \tag{20}$$

## 5 Experimental evaluation

A series of computer experiments was conducted in order to evaluate proposed GA enabled IR framework in both, crisp Boolean IR model and fuzzified Extended Boolean IR model[21, 20, 23, 25]. Experiments were executed using data taken from the LISA[4] 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 Figure 5). 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,

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

three better 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 pertuberation 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 opti-mization 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.

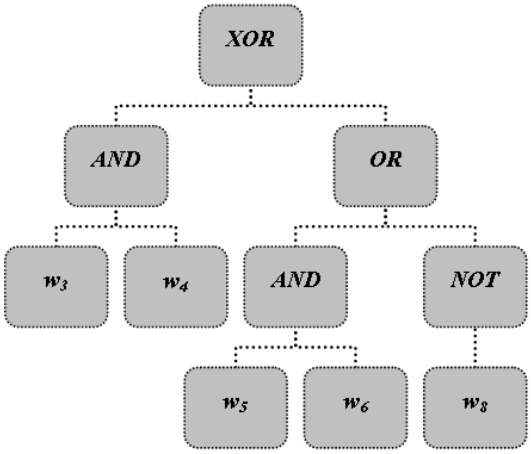



**Fig. 5.** Search query *($w_3$ and $w_4$) xor (($w_5$ and $w_6$) or not $w_8$)* encoded for GP

Table 1 lists the user query and better ranked queries injected into initial population in some experiments.

Table 2 and 3 summarizes the experimental results obtained for different scenarios. Experiments are labeled with the following flags: single point mutation **I**, 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 $F - score$.

**Table 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 ) |

**Table 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 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 |

From the experiments with Boolean queries we conclude the following results: Genetic Algorithms 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 6

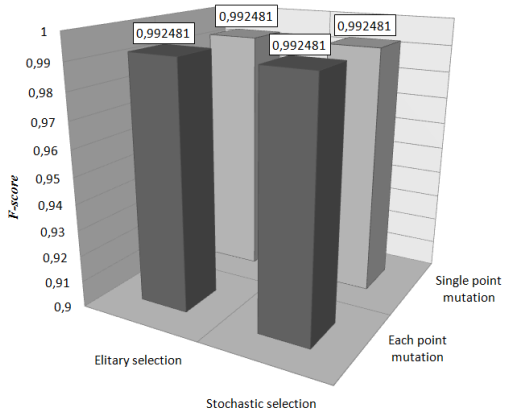



**Fig. 6.** The comparison of achieved F-score for different algorithm setups in BIRM with seeded initial population

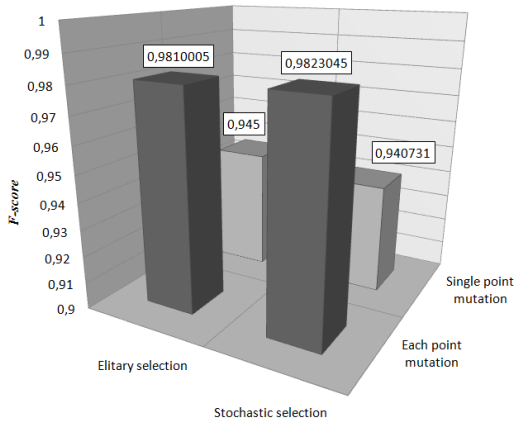



**Fig. 7.** The comparison of achieved F-score for different algorithm setups in EBIRM with seeded initial population

and 7 illustrate the improvements of $F - Score$ of the optimized queries in different experimental cases.

## 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 modelling 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 article. 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 modelling and genetic algorithms for query optimization brings a significant contribution to the ultimate goal of web search: bringing knowledge to man.

## References

1. 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.
2. Ulrich Bodenhofer. *Genetic Algorithms: Theory and Applications*. Lecture notes, Fuzzy Logic Laboratorium Linz-Hagenberg, Winter 2003/2004.
3. Gloria Bordogna and Gabriella Pasi. Modeling vagueness in information retrieval. pages 207–241, 2001.
4. 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.
5. Oscar Cordn, Flix 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.

6. 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.
7. 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.
8. 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.
9. 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.
10. E. Greengrass. *Information retrieval: A survey.* DOD Technical Report TR-R52-008-001, 2001.
11. Stephen P. Harter. Psychological relevance and information science. *JASIS*, 43(9):602–615, 1992.
12. Gareth Jones. Genetic and evolutionary algorithms. In Paul von Rague, editor, *Encyclopedia of Computational Chemistry.* John Wiley and Sons, 1998.
13. 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.
14. 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.
15. Henrik L. Larsen. Retrieval evaluation. In *Modern Information Retrieval course.* Aalborg University Esbjerg, 2004.
16. 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.
17. 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.
18. Melanie Mitchell. *An Introduction to Genetic Algorithms.* MIT Press, Cambridge, MA, 1996.
19. H. O. Nyongesa and S. Maleki-Dizaji. User modelling using evolutionary interactive reinforcement learning. *Inf. Retr.*, 9(3):343–355, 2006.
20. 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.
21. Suhail S. J. Owais, Pavel Krömer, and Václav Snášel. Evolutionary Learning of Boolean Queries by Genetic Programming. In *ADBIS Research Communications*, pages 54–65, 2005.

22. Suhail S. J. Owais, Pavel Krömer, and Václav Snášel. Query Optimization by Genetic Algorithms. In *DATESO*, pages 125–137, 2005.
23. 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.
24. Gerard Salton and Chris Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):pp. 513–523, 1988.
25. 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.
26. H. A. R. Townsend. *Genetic Algorithms - A Tutorial*, 2003.
27. P. Vakkari and N Hakala. Changes in relevance criteria and problem stages in task performance. *Journal of Documentation*, 5(56):540562, 2000.
28. Masaharu Yoshioka and Makoto Haraguchi. An Appropriate Boolean Query Reformulation Interface for Information Retrieval Based on Adaptive Generalization. In *WIRI*, pages 145–150, 2005.