ISSN 1512-1232

# Ranking of Relevant Context Information Based on Content and User Preferences Via DROPT Technique

*Agbele Kehinde[1] and Ekong Daniel[2]

[1]Machine Learning and Intelligent System Research Group,
Dept. of Computer Science, Private Bag X17, Bellville 7535, University of the Western Cape,
Cape-Town, South Africa
[2]Dept. of Mathematical Sciences (Computer Science Option),
Ekiti State University, P.M.B. 5363, Ado-Ekiti, Nigeria.
*Correspondence author

***Abstract***

*As the volume of information available on the Web is growing continuously, ranking algorithms play a significant role in Web search. Presently, there are some ranking algorithms based on content such as TF-IDF and BM25. Regrettably, these algorithms have low precision and are not satisfying users' information needs. This paper proposes an adaptive technique based on the content and user preferences, called DROPT. The idea of relevance has been used to improve retrieval effectiveness based on user's preferences. We have used online interactive reinforcement learning to integrate users' actions to satisfy user information needs and context awareness to reformulate the queries so as to improve the relevance of the retrieved documents. Furthermore, fitness function measure is used to compute the relevance weight of each document. Our technique adapts itself with the environment to present an appropriate ranking for the user's satisfaction. The DROPT algorithm is designed to overcome some of the limitations of these algorithms by comparison and producing an overall better ranking criterion. Experimental results show that our technique retrieves more relevant documents to a specific expert compared to TF-IDF and BM25 algorithms in P@n measure. We have used 30 queries created on WampServer site localhost databases search engine back end to evaluate our technique.*

***Keywords:*** *Web ranking, TF-IDF, BM25, Fitness function, Information retrieval, Relevance feedback, Context.*

## 1. Introduction

Information retrieval (IR) is a well-established discipline in Computer Science since the 1950s - has experienced a revival during the last decade. The reason behind it is to be found mainly in the information explosion caused by the World Wide Web (WWW) and its related technologies. While IR used to be a restricted field with specialized users like librarians and information professionals, today millions of people use IR every day to search the web or search their email, resulting in the need for new user interfaces and query languages [1]. The main focus in the area of IR is satisfying the user need by returning the most relevant information. One of the most important aspects in IR is providing a highly efficient and effective retrieval technique, which retrieves the most relevant documents and rank them at the top of the list. One of the earliest and effective techniques to retrieve and rank documents in IR was based on term frequency. Using term frequency to determine the relevance of the document was the focus of many traditional IR models and it goes back as early as work reported in [2] proposed that the frequency of word occurrences in an article furnishes a useful measurement of word significant. The significance factor of document

ISSN 1512-1232
retrieval will therefore be based on this measurement using ad hoc retrieval task to develop novel technique of document ranking.

A document ranking technique is an algorithm that tries to match documents in the corpus to the user, and then ranks the retrieved documents by listing the most relevant documents to the user at the top of the ranking. Web search engines, designed for retrieving online documents, are very popularand generally perceived to do an excellent job in finding relevant information onthe web. However, the work reported in [3 and 4] highlighted that users interact only with a limited number of search results usually among the first page. The authors demonstrated that information seekers usually choose some relevant informationwithin the first page of results having viewed very few documents. Uncertainabout the availability of other relevant documents most users end their searchsessions after one or two iterations. Only way to satisfy user information needs is to searchon a continuous basis, that is keep looking for information often. This is a time-consuming task.

Given the increasing amount of this information that is available today, there is a clear need for IR systems that can process this information in an efficient and effective way. Efficient and effective text retrieval techniques arecrucial in managing the increasing amount oftextual information.Many problems must be resolvedbefore natural-language processing techniques can be effectivelyapplied to a large collection of texts.Most existing text retrieval techniques rely on indexing keywords.Unfortunately, keywords or index terms alone cannotadequately capture the document contents, resulting in poorretrieval performance. Regrettably, despite the exposure of users to domains of Web retrieval and online documentation systems with document ranking features, it rarely addresses the coreissue: the relevance of the ranked output.

Relevance information is a vital factor for determining the relevance weight, but getting this information is crucial. We achieve this by using the user feedback on retrieved documents, who indicates documents that are relevant and the ones that are not. Ranking hasalways been an important component of any IR system. In the case ofWebsearch its importance becomes critical. To this end the Web offers a rich contextof information which is expressed through the relevancy of document contents. User information needs modelling is utilized as an effort to define a relevance model from the users' perspective to improve the retrieval effectiveness.

Finding relevant documents is one of the most challenging issues for any web search engine. Ideally, the relevance of documents is defined based on user preferences. So the problem of ranking is to sort documents based on user preferences. Certainly, to make the web more interesting, we need a good and efficient ranking algorithm to present more suitable results for users.Usually, there are thousands of relevant documents for each query. Nevertheless, users typically consider only the top 10 or 20 results. To this end our focus is to provide limited number of ranked documents to the users. To achieve this, a better ranking criterion is required and a more efficient mechanism has to be used. This will enable the search engine to present the best relevant documents to the user in response to her queries. However, current ranking algorithms have low precision in average and are not adaptive to user needs. Obviously, we have to devise a solution to achieve a document ranking algorithm with higher effectiveness that is also adaptive to document content and user preferences.

## 2. Motivation

With the growth of the WWW the need for tools to address problemswith information excesshas become more obvious. However, inmany situations the information seeking experience is less than satisfactory: oftensearchers have difficulty finding relevant information from the huge number of information sources that has not matched the rapid growth. The main reason forthis is the lack of effective search interaction and retrieval tools. The existingtools are often ineffective for all but the most simple search tasks [5]. We develop an effective search tool to address the problems of query formulation by modelling information needs to improve retrieval effectiveness.

The contextual relevance of IR element (documents) to include in a context-aware application is vital from the users' and from a computational perspective. However, context information only

ISSN 1512-1232
becomes meaningful when it is judged with respect to the user needs. From theoretical perspective, anticipated contextual information must have an optimal value that falls above an overall fitness mean relevance weight value; otherwise it is considered irrelevant. Research on user interaction with IR results has been of growing importance since the emergent proliferations of Web search engines. Understanding how users work with search interfaces is important whenever large collections of texts are made accessible. Providing users with a perceptive interface for retrieval is essential for organizing information. Such research that concentrates on the presentation of the results to the users and their interaction with them has been coined human-computer information retrieval (HCIR) [6]. Thus, developing HCIR applications is particularly challenging when it comes to context-aware retrieval that adapts the results to personal interests and preferences, the user's current location and other context information relevant to given task [7]. Consequently, we explore how changes of context can cause an adaptation of the result ranking for a given query, so that the same query does not necessarily always lead to the same results.

Existing human-computer traditional systems are aimed to pro-actively find andfilter relevant information that matches user's interests. New interests are stored ina simple profile, containing terms related to different interests and hence resultedin poor performance. In this paper, we propose an adaptive algorithm to explore the use of context information relevance in document ranking based on user actions to their information needs.

## 3. Existing Problem and Our Approach

Generally an Internet surfer does not bother to scan through more than 10 to 20 document shown by a search engine. Therefore the web page ranking should focus on giving higher rank to the relevant documents. In spite of all sophistication of the existing search engine, sometimes they do not give satisfactory result [8 and 9]. The reason is that most of the time a surfer wants a particular type of page like an index page to get the links to an article to know details about a topic. What is lacking in the existing search engines is a suitable categorization of the search documents and ranking according to that. As a result of the proliferation and abundance of information on the Web, ranking algorithms play an important role in Web search. Exactly what information the user wants is unpredictable. So the web page ranking algorithms are designed to anticipate the user requirements from various static (number of links, textual contents) and dynamic (popularity). They are important factors for making one search engine better than another. In this regards, most search engines currently use two major categories of ranking algorithms based on content (classical IR) and link (the web graph).In classical IR [10], the system tries to find documentscorresponding to the user query. Examples of the content-based ranking algorithms are TF-IDF [11] and BM25 [12]. These algorithms are suitable for well formed and structured environmentssuch as digital libraries and collections of scientific articles. In these environments queries are long and wellspecified and the vocabulary is small and relatively well understood.

However, the web consists of a large number of unstructured documents linked together, creating a massivegraph. Furthermore, queries are generally short [13] and vocabulary is huge. This poses new challenges to IR. In addition, since the contents of the web are published in a distributedmanner, this content is often inconsistent and includes a lot of misinformation. Therefore, application of classical IRmethods to web content will result in problems such as low precision and recall, as well as the rank spamming problem [14].To address these limitations, a novel DROPT technique that retrieves the most relevant document though limited and rank them according to their relevance weight at the top of the search list has been proposed. From this point of view, our approach will capture the underlying semantic context terms of the retrieved documents rather that just on weighted words in the TF-IDF specifically in the domain of IR. Previous studies indicate that algorithms using hyperlinks for ranking yield satisfactory results [15]. Their main strength comes from using the links to convey information which can be used to evaluate the importance of documents and therelevancy of documents to the user query. Instances of link-based ranking algorithms are PageRank [16], HITS [17] and DistanceRank [18].Although these algorithms are

ISSN 1512-1232
appropriate in some situations, on average their precision is low compared to content based algorithms [19]. Furthermore, they suffer from shortcomings like the "rich-get-richer' problem [20] that causes young high quality and relevance pages to take a long time tobecome popular. In other words, popular pages are ranked higher and have a higher chance to be browsed by userswhile young pages are likely to be neglected regardless of their quality and relevance. In web information retrieval, the user playsthe most important role in the system and the basic objective is to satisfy him by a good ranking. However, in the TF-IDF ranking algorithmsthere is not any position for the user; directly or indirectly. Thus, there seems to be roomfor better ranking algorithms that take the role of the user into account.

Thus far we have identified two factors that can be used to produce relevance rankings of web content,namely content and user preferences. For this purpose we use relevance judgment for a query to explicitly take user preferences into consideration about retrieved documents to meet their information needs. For combination assessment, we will use content-based methods such as TF-IDF and BM25 [21] for comparison with our developed DROPT algorithmto bring the relevance documents to the top of the result list. A major property of our ranking technique is its adaptability.Depend on the user need and context, the method adapts itself with the environment to present an appropriateranking for the user's satisfaction.We use a fitness function for our ranking technique which shows the satisfaction degree of an average user of the algorithm. The fitness function is computed for each retrieved document via an iterative process using reinforcementlearning. In the reinforcement learning problem, the learning is done by interaction with the prototype WampServer site search engine back end discussed in Section 6.

## 4. The DROPT Technique

The adaptive DROPT algorithm is based on result rankings, independent of the applied knowledge representation, and retrieval method respectively. The IR optimization algorithm proposed analyse the document contents based on users' actions and preferences for ranking. The DROPT algorithm looks at result rankings based on the document relevance weights. Hence, the measure takes normalization to the interval [0,1]. The most important aspect is the introduction of foundation of the weighting relevance scheme based on equation (5) to compute the relevance weight of individual document.The presentation of the search results to the user is an important aspect in human-computer information retrieval (HCIR). The presentation method lists result rankings in ascending order according to relevance weights in response to a given query request. An assumption that suggests itself is that presentation of information about the relevance of documents also influences user's judgments in result rankings. The mathematical definition of the DROPT technique isintroduced in the following subsection.

### 4.1. Formalization of Mathematical Model Definitions

The DROPT technique is based on IR result rankings, where a ranking R consists of an ordered set of ranks. Each rank consists of a relevance value $v \in [0, 1]$ where $v$ represents the relevance of the ranking results. Each rank is assigned an ascending rank number $n$, such that:

$$R = [\{1, v_1, \}, \{2, v_2, \}, ..., \{n, v_n, \}], \textit{where } v_1 > v_2 > ... > v_n. \quad (1)$$

Based on equation (5), a DROPT technique for documents retrieved from a corpus is developed with respect to document index keywords and the query vectors. This based on calculating the weight ($w_{ij}$) of keywords in the document index vector, calculated as a function of the frequency of a keyword $k_j$ across a document $d_i$. Our technique, DROPT is composed of six steps.

**Step 1: Initialization of Parameters**

(a) Let a query vector, Q, be defined as:

$$Q = [q_1 \quad q_2 \quad q_3 \cdots q_l] \quad (2)$$

ISSN 1512-1232

Where, $q_i = (x_i, 1)$, $x_i$ being a term string with a weight of 1.

(b) Let the indexed document corpus be represented by the matrix:

$$D = \begin{bmatrix} d_{11}\ d_{12}\ d_{13} \cdots\cdots d_{1l} \\ d_{21}\ d_{22}\ d_{23} \cdots\cdots d_{2l} \\ \vdots \\ \vdots \\ d_{N1}\ d_{N2}\ d_{N3} \cdots d_{Nl} \end{bmatrix}, \quad (3)$$

where $d_{jk} = (y_{jk}, w_{jk})$, $y_{jk}$ being an index string, with weight $w_{jk}$.

(c) We compute the convolution matrix, representing:

$$W = D \times Q = \begin{bmatrix} w_{11}\ w_{12}\ w_{13} \cdots w_{1l} \\ w_{21}\ w_{22}\ w_{23} \cdots w_{2l} \\ \vdots \\ \vdots \\ w_{n1}\ w_{n2}\ w_{n3} \cdots w_{nl} \end{bmatrix} \quad (4)$$

where $w_{ij} = w_{kl}$ *iff IsEqualStringIgnoreCase*$(q_p, d_{rs})$*; 0, otherwise;* $|l| \leq |n|$, $l$ being the number of terms in the query vector and $n$ the number of retrieved documents that are indexed by at least one keyword in the query vector.

**Step 2: Search String Processing (Matching Mechanism)**

This is a matching process that depends on the relevance judgment rules for matching users' interest with feedback values. When a match is identified the relevance weight ($w_{ij}$) of keywords in the document index vector, is calculated as a function of the frequency of a keyword $k_j$ across a document $d_i$, otherwise no relevance weight is calculated.

A user must specify some information, considered as preferences, pertaining to the query. This context (preferences) provides a high-level description of the users information need and eventually control the search strategy used by the system. In this paper, we focus on modelling the information using rules that best matches user's interest to judge the relevance of competing information need models. Such rule states, among a set of conditions, a particular ***YES*** or ***NO*** together with a weight. The rules are shown in Table 1.

| Threshold score exceeded? | Y | N |
|---|---|---|
| P | HR | HI |
| E | HR | HI |
| G | HR | HI |
| F | MR | MI |
| B | LR | LI |
| H | LR | LI |

*Table 1: Relevance Judgments Model (RJM) Table for User Model Judgments*

ISSN 1512-1232

**Matching values:** Yes (Y), No (N)

**Feedback values:** Perfect (P), Excellent (E), Good (G), Fair (F), Bad (B), Harmful (H)

**Relevance Judgment values:** Highly Ranked (HR), Moderately Ranked (MR), Lowly Ranked (LR), Lowly Ignored (LI), Moderately Ignored (MI) and Highly Ignored (HI).

Each of the cells in Table 1 represent ***IF***<*CONDITION*>***THEN***<*ACTION*>*Statement*. Users can express conditions regarding the values of a preference. For example, the first cell in the Table 1 above is a statement IF <Matching = Y; Feedback = P> THEN < Judgment = HR>, where Y represents matching condition value "YES", P represents feedback value "Excellent" and "HR" represents relevance judgment value "Highly Ranked" respectively. These judgment rules rely on obtaining information from a domain of expert by scoring each of the retrieved documents calculated as a function of the frequency of the keyword across a document. Users provide a judgment of the documents over a scale of [0...30], and the matching is calculated over a scale [0.0...1.0] with feedback values  [0.0…1.0] and relevance judgment (output values) were performed on a non-binary manner, where documents were judged on a six-level scale: Highly Ranked (HR), Moderately Ranked (MR), Lowly Ranked (LR), Highly Ignored (HI), Moderately Ignored (MI), or Lowly Ignored (LI).

**Step 3: Calculate Relevance Weight**

[22] Studied weighted relevance of terms in a document by considering term frequency (*tf*) and term document frequency (*idf*). Term frequency is the number of times a given term occurs in a given document, while document frequency is the number of times the term occurs in all documents. The author argued that the more a term occurs in one document, but less in other documents, the more relevant it is to that document. Consequently the relevance weight is proportional to the term frequency and inverse document frequency. In [10] the relevance weight is given by,

$$w_{ij} = tf_i \times idf_j \tag{5}$$

Where *tf* is the term frequency in the query-document, $idf = \log\left(\frac{N}{n_i}\right)$, $n_i$ is the number of documents indexed containing term $j$ ; $N$ is the total number of documents containing.

We calculate the mean weight score using the weighted root mean squares (RMS) to determine the overall fitness of all documents with respect to a given querycalculated as:

$$\varpi = \prod_{i=1}^{n} \frac{1}{l} \sqrt{\sum_{j=1}^{l} w_{ij}^2} \tag{6}$$

**Step 4: User Feedback from Retrieved Documents**

(a) In order to prevent the relevance of contextual model representation to increase without bounds, the overall relevance judgment is given by:

$$G = \left[g_{ij}\right]_{n \times l} \tag{7}$$

$where\ g_{ij} = \min\left(w_{ij}, \overline{q_{ij}}\right)$

$1 \le i \le n,\ 1 \le j \le l$ G is a query vector with a small-operator defined as a matrix.

(b) Therefore, any weight component of matrix G greater than the mean weight values will be retained to add to a matrix T given by:

$$T = \left\lfloor t_{ij} \right\rfloor_{n \times l} \tag{8}$$

ISSN 1512-1232

where $\begin{cases} t_{ij} = g_{ij}, & if\ g_{ij} \geq \varpi \\ t_{ij} = 0, & if\ g_{ij} < \varpi \end{cases}$ $1 \leq i \leq n, 1 \leq j \leq l$ .

(c) Based on matrix T, we calculate scores, $sco_i$ and generate $D$, for the set of retrieved documents which are the largest weighting value of each corresponding vector given by:

$$Sco_i = max \{ t_{ij} \},\ 1 \leq i \leq n \quad (9)$$

$$1 \leq j \leq l$$

(d) Document $d_i$ is retrieved if $sco_i$ is greater than zero ($sco_i > 0$) and then added into the retrieved document set $D$ shown in equation (9). Hence documents are sorted in ascending order of $Sco_i$, ranked and given to the users to meet their information needs. So, average score ranging between 0 and 1 is computed for each document given by:

$$D = \{d_i \mid if\ Sco_i > 0,\ 1 \leq i \leq n\} \quad (10)$$

**Step 5: Relevance Judgment**

(a) If a user feels that the document is relevant to his/her information needs, he finishes the search, GO to **Step 4** according to user's preference function.

(b) Else, user continues to search the document databases by reformulating the query or stop querying the document database until relevant documents are retrieved. GO to **Step 6**.

**Step 6: Update Tem Weight and Keywords Set**

The keyword set $K$ provided by the documents and the weight values will be updated by the feedback of the users.

(i)Any new query term not belonging to $K$ will be added and a new column of weight value will be computed and expanded for documents routinely.

(ii) If any retrieved document $d_i$ is retrieved by the users, the corresponding weight values with respect to the query keywords will be increased by equation (11). The default of β is set to increase the corresponding weight values.

$w_{ij} = \left( w_{ij} \right)^{\beta}$, where

$$0 < \beta < 1, i \in \{i \mid d_i \in D\} \text{ and } j \in \{j \mid q_j = 1\} \quad (11)$$

We coined the acronym DROPT to name our adaptive algorithm that provides a limited number of ranked documents in response to a given query. Also it improves the ranking mechanism for the search results in an attempt to adapt the retrieval environment of the users and amount of relevant information according to each user's request. Finally, the DROPT measure must be self-learning that can automatically adjust its search structure to a user's query behaviour.

## 5. Implementation Using Perl Programming

Perl an acronym for **P**ractical **E**xtraction and **R**eporting **L**anguage is designed to handle a variety of system administrator functions and provides comprehensive string handling functions. Originally developed by Larry Will at NASA's Jet Propulsion Laboratory in 1986, it has since been improved by hundreds of volunteer developers. It is widely used to write Web server programs for such tasks as automatically updating user accounts and newsgroup postings, processing and removal requests, synchronizing databases and generating reports. Conversely, **DBD: Mysql** is an interface between the Perl programming language and the Mysql programming API that comes with the Mysql relational database management system. Most functions provided by this programming API are supported. **LWP** (short for "Library for WWW in Perl) is a popular group of Perl modules for accessing data on the Web. **CAM::pdf** package reads and writes any document that conforms to the PDF specification generously provided by Adobe.

ISSN 1512-1232

The DBD: Mysql, LWP and CAM:: PDF are PerlActive modules of Perl programming language that extracts information as a function of the frequency of the keyword across a document from the document database collected and stores in the localhost database. The prototype is implemented as a traditional Web based application, with a WampServer site localhost engine back end. The results are shown in Tables 2, 5 and 8. This proposal has been tested with 30 documents;10 for each of the domain of system user expert and the IR system developed gives promising results. This is verified using the evaluation measures in an experimental setting in Section 7. WampServer is a Windows Web development environment. It allows us to create Web applications with Apache2, PHP and a Mysql database. Alongside, PhP Myadmin allows us to manage easily our databases. The WampServer localhost search engine back end allows context-aware agents to pro-actively act on behalf of user based on environment and context to perform several information-related tasks.

### 5.1 Experimental Setting of Generated Data and Discussions

A WampServer localhost search engine back end is created for Domain of system user experts 1, 2, and 3 collectively. Information retrieval and Text Processing, Wireless Sensor Networks and Grid and Distributed Computing, for Domain 1, 2, and 3 respectively.

The formal definition of our fitness function is described below. For any chromosome (document) $w = (w_1, w_2, ..., w_n)$ in the current document collection **N**, its fitness function of the chromosome used is calculated by:

$$F(w) = 1 - \frac{n}{N} \quad , \tag{12}$$

where $n$ is the number of times the keywords are appearing in the whole document, $w$ is the numerical weight of each document, while $N$ is the total number of documents present in the document collection.

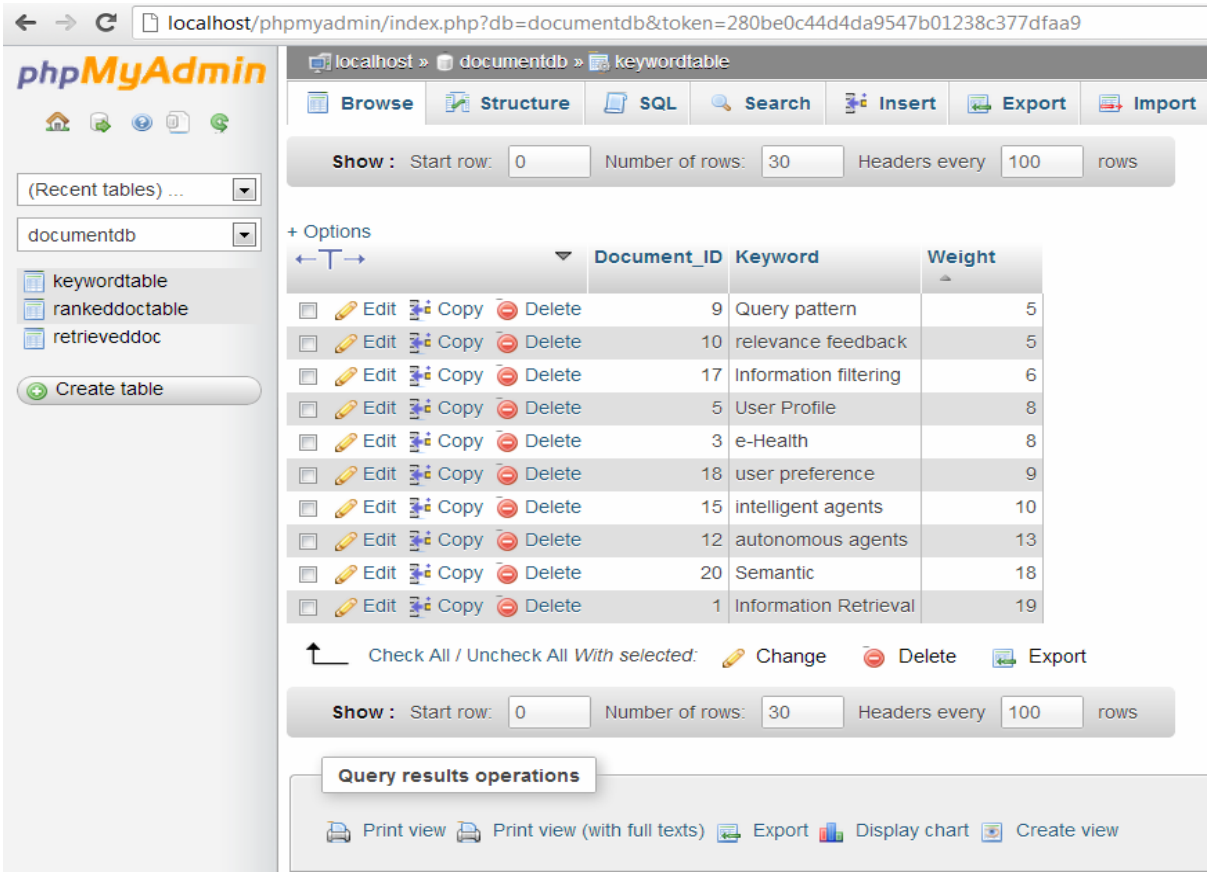


*Figure 1: Screenshot displaying keyword matching based querying results from Domain of System User Expert 1*

ISSN 1512-1232

| Information | | |
|---|---|---|
| **Doc_Id** | **Keywords** | **Weight** |
| 9. | Query pattern | 5 |
| 10. | Relevance Feedback | 5 |
| 17. | Information filtering | 6 |
| 5. | User Profile | 8 |
| 3. | e-Health | 8 |
| 18. | User Preference | 9 |
| 15. | Intelligent agents | 10 |
| 12. | Autonomous agents | 13 |
| 20. | Semantic | 18 |
| 1. | Information Retrieval | 19 |

*Table 2: Derived Information Using CAM::PDF, DBD: Mysql and LWP Modules from Domain of System User Expert 1*

Based on equation (12), the fitness function of each of the chromosomes is shown in Table 3.

| Doc_Id | Chromosome | Fitness Score |
|---|---|---|
| 9. | Query pattern | 0.83 |
| 10. | Relevance Feedback | 0.83 |
| 17. | Information filtering | 0.80 |
| 5. | User Profile | 0.73 |
| 3. | e-Health | 0.73 |
| 18. | User Preference | 0.70 |
| 15. | Intelligent agents | 0.67 |
| 12. | Autonomous agents | 0.57 |
| 20. | Semantic | 0.40 |
| 1. | Information Retrieval | 0.37 |

*Table 3: Fitness Calculation for Domain of System User Expert 1*

From Table 3, the overall average fitness value **F = 0.626** is obtained according to equation (12) as shown in Table 4.

| Doc_Id | Chromosomes | Fitness Score | Overall fitness score |
|---|---|---|---|
| 9. | Query pattern | 0.83 | $\geq 0.626$ |
| 10. | Relevance Feedback | 0.83 | $\geq 0.626$ |
| 17. | Information filtering | 0.80 | $\geq 0.626$ |
| 5. | User Profile | 0.73 | $\geq 0.626$ |
| 3. | e-Health | 0.73 | $\geq 0.626$ |
| 18. | User Preference | 0.70 | $\geq 0.626$ |
| 15. | Intelligent agents | 0.67 | $\geq 0.626$ |
| 12. | Autonomous agents | 0.57 | |
| 20. | Semantic | 0.40 | |
| 1. | Information Retrieval | 0.37 | |

*Table 4: Overall Fitness Score for Domain of system user expert 1*

The values displayed in Table 4 shows the results of the search system for documents retrieved from a WampServer localhost search engine back end. Documents are sorted and were set in ascending order of Retrieval Status Values (RSV). Hence, $Doc\_Id_9$, $Doc\_Id_{10}$, $Doc\_Id_{17}$, $Doc\_Id_5$, $Doc\_Id_3$, $Doc\_Id_{18}$, and $Doc\_Id_{15}$ are ranked accordingly because their relevance scores

ISSN 1512-1232

are above the threshold score (**F=0.626**); hence considered as relevant documents. Results that fall below threshold value are not displayed to the user (irrelevant).

| Information | | |
|---|---|---|
| **Doc_Id** | **Keywords** | **Weight** |
| 4. | Swarm intelligent | 2 |
| 7. | Data gathering | 2 |
| 6. | Traffic load | 3 |
| 2. | Medium access control | 3 |
| 13. | Passive clustering | 3 |
| 16. | Intelligent sensors | 3 |
| 14. | Wireless telemedicine | 4 |
| 11. | Clustering algorithm | 4 |
| 8. | Ant colony optimization | 5 |
| 19. | Health care | 16 |

*Table 5: Derived Information Using CAM::PDF, DBD: Mysql and LWP Modules from Domain of System User Expert 2*

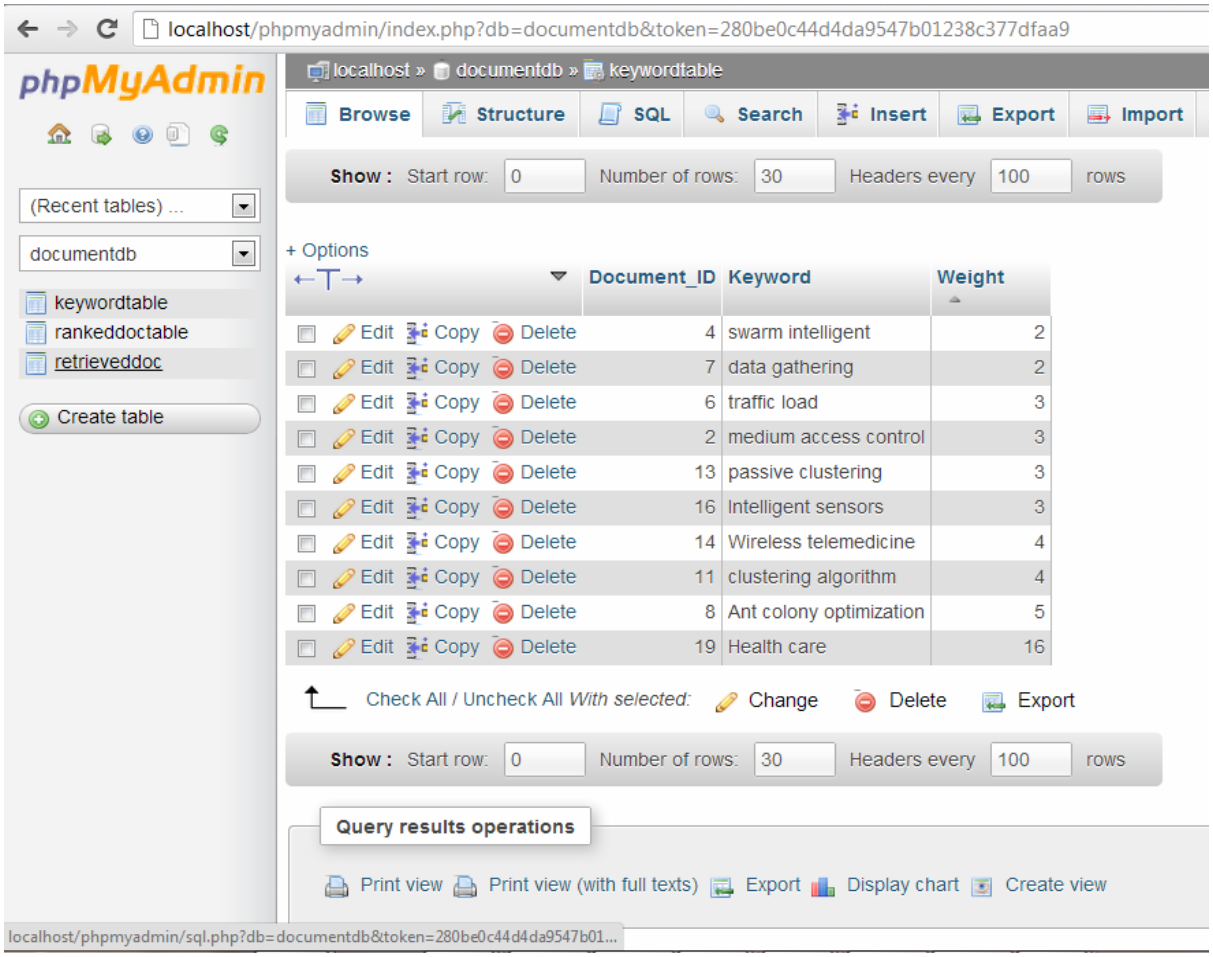


*Figure 2: Screenshot displaying keyword matching based querying results from Domain of System User Expert 2*

ISSN 1512-1232
Based on equation (12), the fitness function of each of the chromosomes is shown in Table 6.

| Doc_Id | Chromosome | Fitness Score |
|---|---|---|
| 4. | Swarm intelligent | 0.93 |
| 7. | Data gathering | 0.93 |
| 6. | Traffic load | 0.90 |
| 2. | Medium access control | 0.90 |
| 13. | Passive clustering | 0.90 |
| 16. | Intelligent sensors | 0.90 |
| 14. | Wireless telemedicine | 0.87 |
| 11. | Clustering algorithm | 0.87 |
| 8. | Ant colony optimization | 0.83 |
| 19. | Health care | 0.47 |

*Table 6: Fitness Calculation for Domain of System User Expert 2*

From Table 6, the overall average fitness value (**F = 0.85)** is obtained according to equation (12) as shown in Table 7.

| Doc_Id | Chromosomes | Fitness Score | Overall fitness score |
|---|---|---|---|
| 4. | Swarm intelligent | 0.93 | ≥ 0.85 |
| 7. | Data gathering | 0.93 | ≥ 0.85 |
| 6. | Traffic load | 0.90 | ≥ 0.85 |
| 2. | Medium access control | 0.90 | ≥ 0.85 |
| 13. | Passive clustering | 0.90 | ≥ 0.85 |
| 16. | Intelligent sensors | 0.90 | ≥ 0.85 |
| 14. | Wireless telemedicine | 0.87 | ≥ 0.85 |
| 11. | Clustering algorithm | 0.87 | ≥ 0.85 |
| 8. | Ant colony optimization | 0.83 | ≥ 0.85 |
| 19. | Health care | 0.47 | |

*Table 7: Overall Fitness Score for Domain of System User Expert 2*

The values displayed in Table 7 shows the results of the search system for documents retrieved from a WampServer localhost search engine back end. Documents are sorted and were set in ascending order of Retrieval Status Values (RSV). Hence, *Doc_Id*$_4$, *Doc_Id*$_7$, *Doc_Id*$_6$, Doc_*Id*$_2$, *Doc_Id*$_{13}$, *Doc_Id*$_{16}$, and *Doc_Id*$_{14}$, *Doc_Id*$_{11}$, and *Doc_Id*$_8$ are ranked accordingly because their relevance scores are above the threshold score (**F=0.85**); considered as relevant.

| Information | | |
|---|---|---|
| **Doc_Id** | **Keywords** | **Weight** |
| 21. | Workflow scheduling | 13 |
| 22. | Grid environment | 2 |
| 23. | Efficient security | 4 |
| 24. | Authorization | 2 |
| 25. | Grid portals | 4 |
| 26. | Homomorphic encryption | 14 |
| 27. | Medical grid | 2 |
| 28. | Authorization | 2 |
| 29. | Trust | 2 |
| 30. | Interoperation | 8 |

*Table 8: Derived Information Using CAM::PDF, DBD: Mysql and LWP Modules from Domain of System User Expert 3*

ISSN 1512-1232

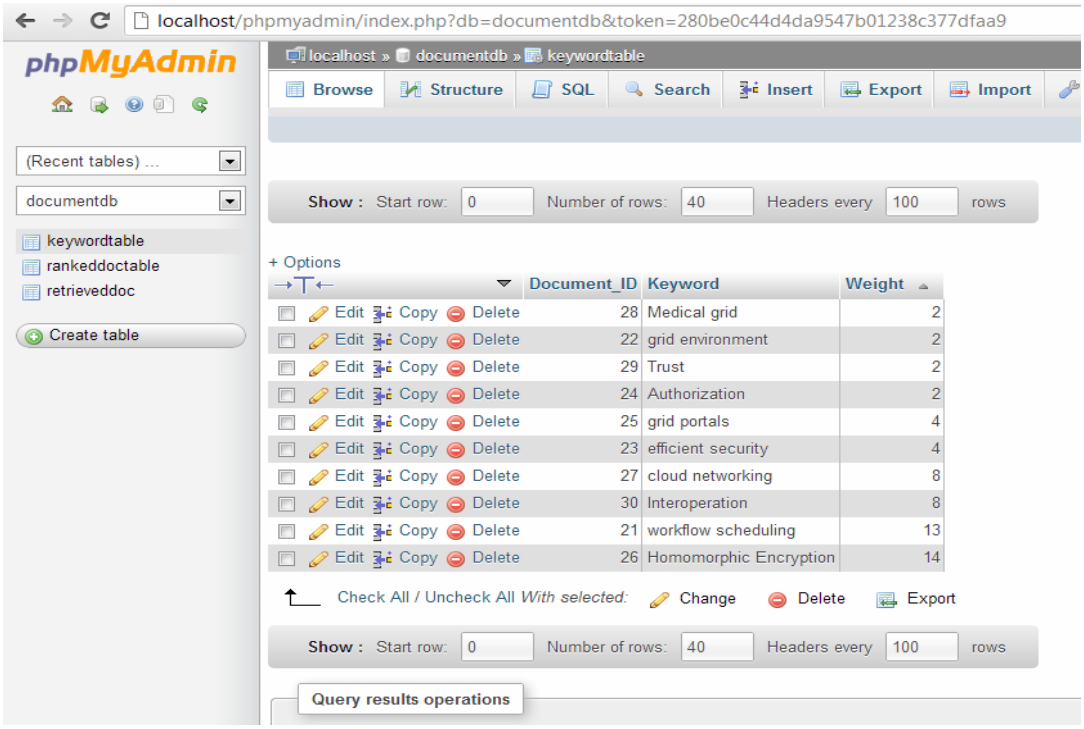


*Figure 3: Screenshot displaying keyword matching based querying results from Domain of System User Expert 3*

Based on equation (12), the fitness function of each of the chromosomes is shown in Table 9.

| Doc_Id | Chromosome | Fitness Score |
|---|---|---|
| 21. | Workflow scheduling | 0.57 |
| 22. | Grid environment | 0.93 |
| 23. | Efficient security | 0.87 |
| 24. | Authorization | 0.93 |
| 25. | Grid portals | 0.87 |
| 26. | Homomorphic encryption | 0.53 |
| 27. | Medical grid | 0.93 |
| 28. | Authorization | 0.93 |
| 29. | Trust | 0.93 |
| 30. | Interoperation | 0.73 |

*Table 9: Fitness Calculation for Domain of System User Expert 3*

From Table 9, the overall average fitness value **F = 0.735** is obtained according to equation (12) as shown in Table 10.

| Doc_Id | Chromosomes | Fitness Score | Overall fitness score |
|---|---|---|---|
| 21. | Workflow scheduling | 0.57 | |
| 22. | Grid environment | 0.93 | $\geq$ 0.735 |
| 23. | Efficient security | 0.87 | $\geq$ 0.735 |
| 24. | Authorization | 0.93 | $\geq$ 0.735 |
| 25. | Grid portals | 0.87 | $\geq$ 0.735 |
| 26. | Homomorphic encryption | 0.53 | |
| 27. | Medical grid | 0.93 | $\geq$ 0.735 |
| 28. | Authorization | 0.93 | $\geq$ 0.735 |
| 29. | Trust | 0.93 | $\geq$ 0.735 |
| 30. | Interoperation | 0.73 | |

*Table 10: Overall Fitness Score for Domain of System User Expert 3*

ISSN 1512-1232

The values displayed in Table 10 shows the results of the search system for documents retrieved from a WampServer localhost search engine back end. Documents are sorted and were set in ascending order of Retrieval Status Values (RSV). Hence, *Doc_Id*$_{22}$, *Doc_Id*$_{23}$, *Doc_Id*$_{24}$, Doc_*Id*$_{25}$, *Doc_Id*$_{27}$, *Doc_Id*$_{28}$, and *Doc_Id*$_{29}$, are ranked accordingly because their relevance scores are above the threshold score (**F=0.735**); hence considered as relevant.

The indexed keywords represent domain of knowledge of the system users (3 PhD students in an experimental setting at Ice Box Research Lab. UWC; where comparisons are conducted between different experimental runs of the system users). The retrieval effectiveness will be measured using well known metrics in the IR community [23 and 1]: (1) *Precision*, which is the number of retrievedrelevant documents over the total number of retrieved documents; (2) *Recall*,which is the number of relevant documents that are retrieved over the totalnumber of known relevant documents in the document collection. Ranking performance result is discussed between the relevance judgment values during performance evaluation in Section 6.

## 6. Ranking Performance Results

With the intention of measure ranking performance, the DROPT technique for ranking search results list was tuned by experimenting with the system for relevance judgment. Each query produced a document based on the matching conditions and the retrieval was repeated for 10 query reformulations from the domain of system user experts. The underlying philosophy of the relevance judgment rules for user model judgment using DROPT technique is to rank those documents, which exceeded the overall weighted fitness score that the system user judges to be relevant to his/her information needs, and ignore those documents the system users judges to be irrelevant (less preferred). In the present study, we explain the six different information needs of the system users. Hence, if the system user's judges a document to be relevant then the weight (significance) value of the query used for retrieval of the document should be highly ranked (HR) and particularly more so if the matching value is No, lowly ignored (LI). Conversely, if the user feedback is bad (not relevant) but the matching value between query and document exceeded relevance value, highly ranked (HR), and then the relevance value of the document should be ignored completely. System users provide a judgment of the documents over a scale of [0…30] and the matching value is calculated over a scale of [0.0… 1.0]. Figure 4 shows a better ranked list that help the user fill their information needs. Table 11 shows the MAP results and Table 12 shows the precision results at known relevant documents for ranking performance from the domain of experts.

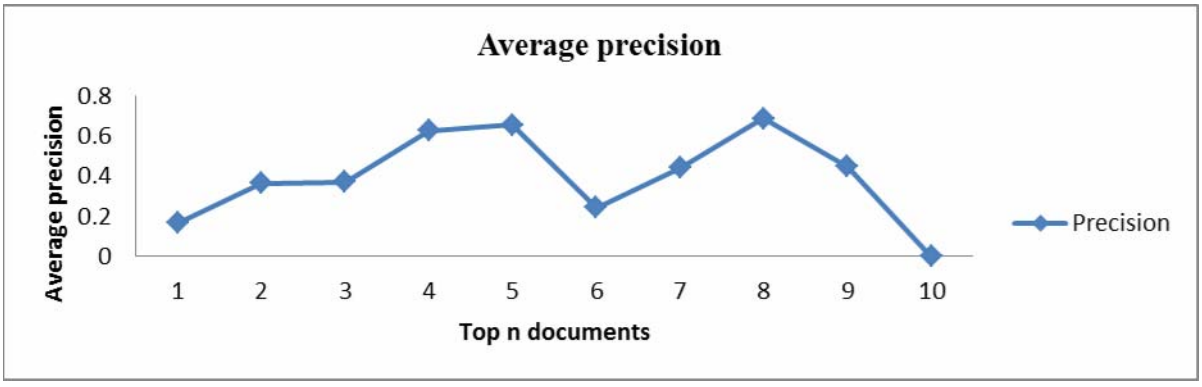


*Figure 4: Average precision Graph for ranking performance results*

| **Generations** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **MAP** | 0.167 | 0.364 | 0.370 | 0.626 | 0.655 | 0.242 | 0.441 | 0.687 | 0.448 | 0.000 |

*Table 11: Mean average precision results for ranking performance from 3 domain of experts*

ISSN 1512-1232

| Document # | Queries | Relevant | Tf | Precision | Fitness score |
|---|---|---|---|---|---|
| **1** | Information retrieval | | 19 | 0.000 | 0.37 |
| **2** | Medium access control | X | 3 | 0.500 | 0.90 |
| **3** | E-health | | 8 | 0.000 | 0.73 |
| **4** | Swarm intelligent | X | 2 | 0.500 | 0.93 |
| **5** | User profile | | 8 | 0.000 | 0.73 |
| **6** | Traffic load | X | 3 | 0.500 | 0.90 |
| **7** | Data gathering | X | 2 | 0.571 | 0.93 |
| **8** | Ant colony optimization | X | 5 | 0.625 | 0.83 |
| **9** | Query pattern | X | 5 | 0.667 | 0.83 |
| **10** | Relevance feedback | X | 5 | 0.700 | 0.83 |
| **11** | Clustering algorithm | X | 4 | 0.727 | 0.87 |
| **12** | Autonomous agent | | 13 | 0.000 | 0.57 |
| **13** | Passive algorithm | X | 3 | 0.692 | 0.90 |
| **14** | Wireless telemedicine | X | 4 | 0.714 | 0.87 |
| **15** | Intelligent agents | | 10 | 0.000 | 0.67 |
| **16** | Intelligent sensors | X | 3 | 0.688 | 0.90 |
| **17** | Information filtering | X | 6 | 0.706 | 0.80 |
| **18** | User preference | | 9 | 0.000 | 0.70 |
| **19** | Health care | | 16 | 0.000 | 0.47 |
| **20** | Semantic | | 18 | 0.000 | 0.40 |
| **21** | Workflow scheduling | | 13 | 0.000 | 0.57 |
| **22** | Grid environment | X | 2 | 0.591 | 0.93 |
| **23** | Efficient security | X | 4 | 0.609 | 0.87 |
| **24** | Authorization | X | 2 | 0.625 | 0.93 |
| **25** | Grid portals | X | 4 | 0.640 | 0.87 |
| **26** | Homomorphic Encryption | | 14 | 0.000 | 0.53 |
| **27** | Cloud networking | X | 2 | 0.630 | 0.93 |
| **28** | Medical grid | X | 2 | 0.643 | 0.93 |
| **29** | Trust | X | 2 | 0.655 | 0.93 |
| **30** | Interoperation | | 8 | 0.000 | 0.73 |
| **Average** | | | | 0.631 | 0.75 |

*Table 12: Precision results for ranking performance at known relevant documents*

As indicated in Table 12, scores that falls below overall weighted fitness values (0.75) for the ranking parameter do not show significant ranking improvement. This is because at low ranking scores below this value, irrelevant documents are rejected by the system user. On the other hand, documents whose scores fall above the overall weighted fitness score are retrieved and then marked 'X'. Hence ranked and given to the user to meet his/her information needs. This adapts and explore new domain for potentially relevant documents. Therefore, when the environment of the adaptive system changes the highest ranked documents of interest automatically adjust to the new environment. The best ranking performance of the system is given by medium values between (0.692-0.727) of the precision values. As shown in Figure 5 the system is more stable for ranking parameter value of 0.727 from domain of system user expert 2 and, the number of ranked relevant documents in the search result is also noticeably higher than for the other ranking parameter values from domain of experts 1 and 3. Also considering Figure 6, which shows the total ranked relevant documents retrieved in the 19 search processes, the ranking performance of 0.727 has the highest number of ranked documents retrieved from domain of expert 2.

ISSN 1512-1232

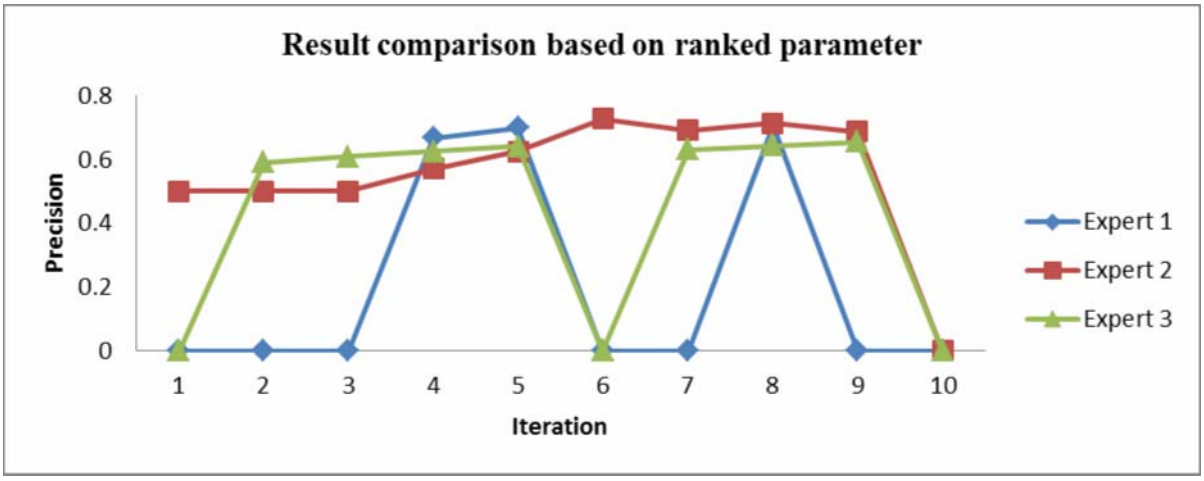


*Figure 5: Precision Graph for ranking performance results with all values from the domain of system users' experts*

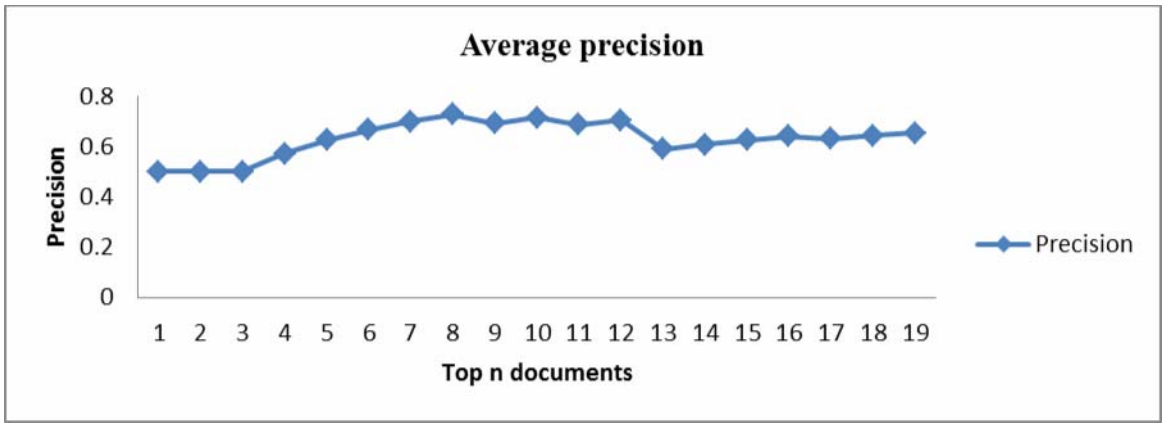


*Figure 6: Precision Graph for ranking performance results at known relevant documents*

## 7. Experimental Results of DROPT Technique

We compare our ranking algorithms with selected well known baseline algorithms such as TF-IDF and BM25 to evaluate the performance of our ranking technique in standard P@n measure. For the information needs (requests) and document collections of the experiment, relevance was assessed by different experienced system users in their domain of experts (3 PhD students). They are vast in their domain and were asked to judge the relevance of the retrieved documents on a six level scales: (0=Harmful, 1=Bad, 2=Fair, 3=Good, 4=Excellent and 5=Perfect) with respect to a given query. For comparison of algorithms, we have used "Precision at position n" (P@n) metrics [24]. Precision at *n* measures the relevancy of the top *n* results ofthe ranking list with respect to a given query (equation 13).

$$P@n = \frac{No\ of\ relevant\ documents\ in\ top\ n\ results}{n} \tag{13}$$

P@n can only handle cases with binary judgment "relevant" or "irrelevant" with respect to a given query at rank n. To compute P@n, 30queries were judged in these 6 levels by users.

For the evaluation of our algorithm we conducted testing the system. The test process involves using the 30 queries provided by the system users. The measure (P@n) is used for evaluation. Naturally, we compute them for each query and then take the average dimension (*n*) for all queries. Figure 7 shows comparison of the DROPT algorithm with other algorithms in the P@n

ISSN 1512-1232
measure. As the figure shows, our adaptive algorithm outperforms the others. DROPT algorithm achieves a 25% in P@n compared to BM25 which is the best one of the other. The figure compares the precision for these 10 queries set between the TF-IDF, BM25 and DROPT. It shows that the precision value of the proposed ranking technique is comparatively higher for all the query sets. This achievement resides in the combination of content-based algorithms using user preferences in query reformulations. In this regards, the number of top *n* results showed to users will depicts the relevancy degree of the retrieved documents with respect to a given query with rank *n* (judged by the system users).

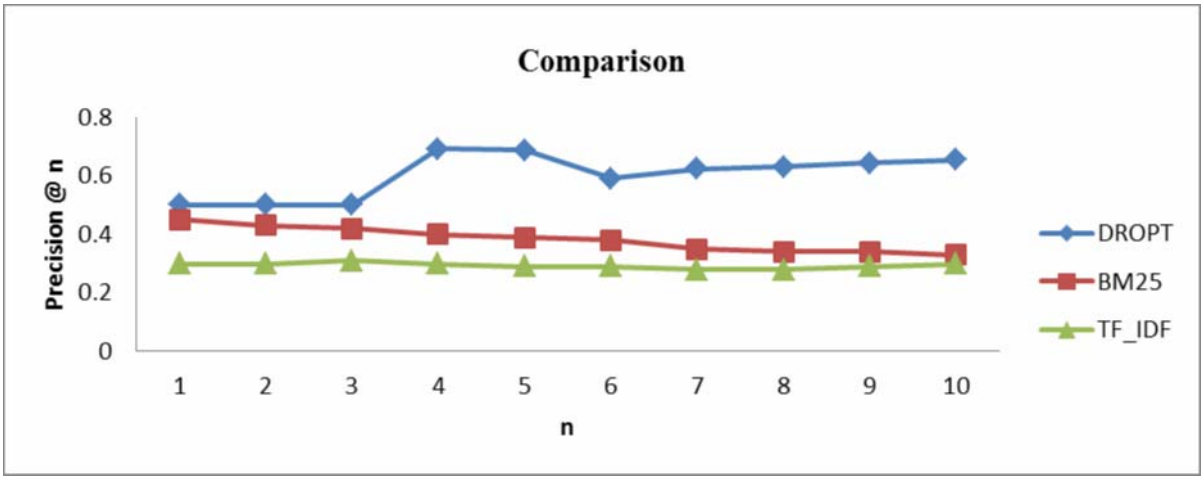


*Figure 7: Comparison of DROPT with BM25 and TF-IDF in the P@n measure.*

## 8. Conclusion

Present ranking algorithms (content-based) suffered from low precision and recall. In addition they are not suitable for some situations and dependent on the context, they will work differently. This paper proposed an adaptive DROPT technique based on content and user preferences to improve the relevance of retrieved documents. We have given this algorithm an acronym DROPT. DROPT algorithm tries to adapt itself with user information needs. We have used the relevance assessment comparison in our algorithm for ranking to improve retrieval effectiveness based on the user's feedback. Our algorithm judge the relevance of the search results based on the relevance weight of the retrieved documents. We have used content-based algorithms such as TF-IDF and BM25 for user preferences in comparison with our algorithm. Our ranking algorithm defined a factor called fitness function to compute the weight of each retrieved document using interactive reinforcement learning method. We used the created WampServer site localhost search engine back end (which was wrapped around Google) for document collections and 30 related queries from the domain of the different system user experts for evaluation of DROPT technique. We have compared DROPT technique with other single algorithms using P@n measure and found interesting improvements. The proposed algorithm has some interesting features like scalability and adaptability. It is scalable in that we can add any new algorithm easily and also adaptable in that it adapts itself with user information needs within the environment.

## References

1. Manning, C.D., Raghavan, P., Schutze, H. (2008): Introduction to Information Retrieval. Cambridge University Press, Cambridge.
2. Luhn, H. P. (1958). The automatic creation of literature abstracts, IBM Journal of Research and Development, Vol. 2, pp. 159-168.

ISSN 1512-1232
3. Jansen, B.J., Spink A. and Saracevic, T. (2000): Real life, real users, and real needs: a study and analysis of users on the Web. Information Processing and Management. Vol. 36, no.2, pp. 207-227.
4. Jansen, B.J., Spink, A., Saracevic, T. (2000): Real life, real users, and real needs: a study and analysis of user queries on the Web. Inf. Process. Manag. Vol. 36, pp: 207–227.
5. Dennis, S., McArthur, R. and Bruza, P. (1998). Searching the WWW made easy? The Cognitive Load imposed by Query Refinement Mechanisms. Proceedings of the 3rd Australian Document Computing Symposium.
6. Marchionini, G (2006): Toward Human-Computer Information Retrieval. June/July 2006 Bulletin of the American Society for Information Science and Technology, available from http://asis.org/Bulletin/Jun-06/marchionini.html, 2006.
7. Brown, P. J and Jones, G. J. F (2001): Context-Aware Retrieval: Exploring a New Environment for Information Retrieval and Information Filtering. Personal and Ubiquitous Computing, Vol. 5, pp. 253-263.
8. Baeza-Yates, R, and Emilio, D. (2004): "Web page ranking using link attributes," Proceedings of the 13th international World Wide Web conference on Alternate track papers and posters, May 2004.
9. Xing, W, Ghorbanifar, A. (2004): Weighted PageRank algorithm;" Proceedings of the Second Annual Conference on Communication Networks and Services Research, 19-21 May 2004; pp. 305 – 314.
10. Baeza-Yates, R., & Ribeiro-Neto, B. (1999). Modern Information Retrieval. ACM Press/ Addison-Wesley.
11. Salton, G., and Buckley, C (1988). Term-Weighting approaches in automatic text retrieval, Information Processing and Management, Vol. 24, Issue 5, pp. 513-523.
12. Robertson, S. E., Walker, S., Hanocock Beaulieu, M. M., Gatford, M., and Payne, A. (1995). Okapi at TREC-4. In NIST Special Publication. The fourth Text Retrieval Conference (TREC-4), pp. 73-96.
13. Zhang, Y., & Moffat, A. (2006). Some Observations on User Search Behavior.11th Australasian Document Computing Symposium pp.1-8.
14. Henzinger, M., Motwani, R., & Silverstein, C. (2002). Challenges in web search engines, SIGIR Forum Vol.36, no.2.
15. Henzinger, M. (2001). Hyperlink analysis for the web. IEEE Internet Computing, Vol. 5, Issue 1, pp.45–50.
16. Page L., Brin S., Motwani, R. & Winograd T. (1998) "The PageRank citation ranking: Bringing order to the web", In Technical report, Stanford Digital Libraries, pp. 1-17.
17. Kleinberg J. M. (1999) "Authoritative sources in a hyperlinked environment", Journal of the ACM, Vol. 46, No. 5, pp. 604 –632.
18. Zareh Bidoki, A. M., & Yazdani, N: (2007) Distance Rank: An intelligent ranking algorithm for web pages, Information Processing and Management, doi:10.1016/j.ipm.2007.06.004.
19. Najork, M., Zaragoza, H., & Taylor, M. J. (2007). Hits on the web: how does it compare? In Proceedings of SIGIR'07 pp. 471- 478.
20. Cho, J., Roy, S., & E. Adams, R. (2005). Page Quality: In Search of an Unbiased Web Ranking. In Proceedings of ACM International Conference on Management of Data (SIGMOD).
21. He, B., & Ounis, I. (2005). A study of Dirichlet priors for term frequency normalization. In Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval pp. 465 – 471).
22. Salton, G. (1971). The SMART retrieval system: Experiments in automatic document processing. Prentice-Hall.
23. Baeza-Yates, R., and Ribeiro-Neto, B. (2011): Modern Information Retrieval: The Concepts and Technology Behind Search, 2nd edn. Addison-Wesley, Reading.

ISSN 1512-1232
24. Jarvelin, K and Kekalainen, J (2000). IR evaluation methods for retrieving highly relevant documents. Published in: Belkin, N. J., Ingwersen, P. and Leong, M. K. (eds). Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, NY: ACM, pp. 41-48.

Article received: 2013-03-27