International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:10, No:9, 2016

Algorithm for Information Retrieval Optimization

Kehinde K. Agbele, Kehinde Daniel Aruleba, Eniafe F. Ayetiran

***Abstract*—**When using Information Retrieval Systems (IRS), users often present search queries made of ad-hoc keywords. It is then up to the IRS to obtain a precise representation of the user's information need and the context of the information. This paper investigates optimization of IRS to individual information needs in order of relevance. The study addressed development of algorithms that optimize the ranking of documents retrieved from IRS. This study discusses and describes a Document Ranking Optimization (DROPT) algorithm for information retrieval (IR) in an Internet-based or designated databases environment. Conversely, as the volume of information available online and in designated databases is growing continuously, ranking algorithms can play a major role in the context of search results. In this paper, a DROPT technique for documents retrieved from a corpus is developed with respect to document index keywords and the query vectors. This is based on calculating the weight (wij) of keywords in the document index vector, calculated as a function of the frequency of a keyword kj across a document. The purpose of the DROPT technique is to reflect how human users can judge the context changes in IR result rankings according to information relevance. Our technique for IR optimization is generic for ranking retrieved documents. This paper shows that it is possible for the DROPT technique to overcome some of the limitations of existing traditional $(tf \times idf)$ algorithms via adaptation. A context-based IRS is developed whose retrieval effectiveness is evaluated using precision and recall metrics. The results demonstrate how to use attributes from user interaction behaviour to improve IR effectiveness.

***Keywords*—**Internet ranking, $tf \times idf$, BM25, information retrieval, relevance feedback, information relevance, DROPT technique.

I. Introduction

RECENT years have witnessed ever-growing amount of online information. The development of the World Wide Web (WWW) led to an increase in the volume and diversity of accessible information. Today, the Internet is a huge information repository. The question that now arises is how access to this information can be effectively supported. Users require the assistance of tools aimed to locate documents that satisfy their specific needs. Therefore, it is important to understand and analyse the underlying data structure of the Internet for efficient IR. Web data mining techniques, along with other areas like database (DB), Natural Language Processing (NLP), IR, and Machine Learning etc., can be used to solve these challenges. Search engines like Google, Yahoo, Web Crawler, Bing etc., are used to find information from the Internet by users. The simple architecture of a search engine is shown in Fig. 1. There are three important components in a search engine, which are the Crawler, Indexer and Ranking mechanisms. The Crawler is also called a robot or spider that traverses the Internet and downloads documents. The downloaded documents are sent to an indexing module that parses the web documents and builds the index based on the keywords in those documents. An index is generally maintained using the keywords. When a user types a query using keywords on the interface of a search engine, the query processor component matches the query keywords with the index and returns the URLs of the documents to the user with particular emphasis on the ranking mechanism. Before users are presented with the documents, the search engines use a ranking mechanism to show the most relevant documents at the top and least relevant ones at the end. One step in optimizing the IR (search queries) is the deployment of our proposed DROPT algorithm.

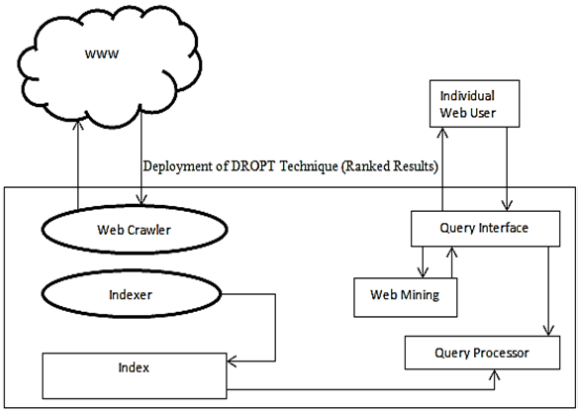

Fig. 1 Architecture of a Search Engine

II. Related Work

In this section, we briefly summarize related works, and in particular, research that has influenced the focus of this paper. Text-based IR is one of the oldest areas of research in Computer Science, and a number of approaches have been devised over the years [1], [2]. Methods based on 'bag-of-word' representations, where the frequency of terms in documents are used to define a vector space remain dominant, with variations of the Term Frequency-Inverse Document Frequency model (TF-IDF) being the most popular (e.g. the Pivot TF-IDF technique [1]. Boolean and Bayesian approaches have also been well-studied and find use in practice. Reference [19] proposed improvements of TF-IDF based text search using the DROPT technique to enhance effective ranking performance.

K. K. Agbele (Dr.) is a Senior Lecturer at the Department of Mathematics Computer Science, Elizade University, PMB 002 Ilara-Mokin, Ondo-State Nigeria (phone: 08088268717 or 08059616285, e-mail: agbelek@gmail.com).
Mr. K.D. Aruleba and Mr. F.E. Ayetiran are Assistant Lecturers at the Department of Mathematics and Computer Science, Elizade University, Ilara-Mokin, Ondo-State, Nigeria (e-mail: Kehinde.aruleba@elizadeuniversity.edu.ng, Eniafe.ayetiran@elizadeuniversity.edu.ng).

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. This is achieved by the IR system that matches the keywords against the document index in order to retrieve matching documents. The system then ranks the documents applying some algorithm that measures the similarity between the query and the documents representation. There are popular Internet search engines designed for retrieving online documents and are typically perceived to do an excellent job in finding relevant information. However, the studies reported in [3]-[5] highlighted that users interact only with a limited number of search results typically among the first page. The authors demonstrated that information seekers usually choose some relevant information within the first page of results having viewed very few documents. Uncertain about the availability of other relevant documents, most users end their search sessions after one or two iterations. The only way to satisfy user information needs is to search on a continuous basis; that is to continue looking for information often. This is a time-consuming task.

Relevance information is a vital factor for determining the relevance weight, but getting this information is crucial. We can achieve this by using the user feedback on retrieved documents, which indicates documents that are relevant and otherwise. Ranking has always been an important component of any IR system. In the case of an Internet search its importance becomes critical. Reference [6] proposed a ranking technique for multi-search projections on the WWW for a results aggregation model based on query words, search results, and search history to achieve the user's intention. To this end, the WWW can offer a rich context of information that can be expressed through the relevancy of document contents. Reference [7] proposed a model for online learning that is specifically adequate for user feedback. The experiment conducted shown retrieval effectiveness for Internet search ranking. In the context of an Internet search ranking, it is important these techniques aim at finding the best ordering function over the returned documents. The authors argue that regression on labels may be adequate and, indeed, competitive in the case of large numbers of retrievals. To make the WWW more interesting, there is a need to develop an effective and efficient ranking algorithm to deliver more suitable results for users. User information needs modelling is utilized in an effort to define a relevance model from the users' perspective to improve retrieval effectiveness [8].

Ideally, the relevance of documents should be defined based on user preferences. So the problem of ranking of retrieved documents is to sort documents based on user preferences. Relevance is a standard measure utilized in IR to evaluate effectiveness of an IR system based on the documents retrieved. The work reported in [9], [10] are studies on the concept of relevance and relevance assessment. The effectiveness of an IR system is determined primarily by the relevance assessment of the retrieved information [11]-[13]. Therefore, the concept of relevance, however, is one that is subjective and influenced by diverse factors. In the first instance, the queries posed to the IR systems are, most of the time, not optimal in terms of describing the required information with respect to an individual user's information needs. To this end, user perception and user knowledge level are factors that influence the relevance of a retrieved document. For example, [7] proposed a new model for online learning that is specifically adequate for user feedback. The experiment conducted shows the retrieval effectiveness for a WWW search ranking. Certainly, to make the Internet more interesting, there is need to develop an appropriate and efficient ranking algorithm to deliver more suitable results for individual 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, the focus of this paper is to provide a limited number of relevant ranked documents to individual users. To achieve this, a better ranking criterion is required and a more efficient mechanism can be used, which will enable the search engine to present the most relevant documents to users. Obviously, we can achieve a document ranking with higher retrieval effectiveness that is adaptive to semantic characteristics of the document itself and user preferences. The need for query operations arises from the user's difficulty to formulate queries without a full understanding of the underlying collection and the IR environment [14].

Ranking is the process of ordering the results of a query according to measures of relevance to the user. Although not essential to IR, ranking can greatly simplify the interaction with large document collections and is employed widely, especially in the context of an Internet search. Over the years, various techniques to deal with this problem have been proposed. Reference [15] divides these techniques into two broad categories: global methods that use information independent of the query, and local methods that adjust a query relative to the documents that initially seem to match it. Global methods use a thesaurus to expand or reformulate a query with similar terms. Thus, we propose a solution to this problem of ranking of retrieved documents based on our proposed DROPT technique.

III. Parameters Used for Ranking Principles

In this section we study the problem of ranking of retrieved documents from the search engine back end prototype developed in this study (as interaction between the information user and the information source), such that the most relevant documents d are retrieved at the start of the list, given a query q. For example, we desire to rank a set of scientific articles such that those related to the query 'information retrieval' are retrieved first. The basic assumption we make is that such a ranking can be obtained by a weighting function $w(tf \times idf)$ which conveys to us how relevant document d is for query q. The document ranking will be done by taking a weighted average of all determined parameters.

The necessity of being able to deal with each (document, query) pair independently arises from details of practicality on search engine back end prototype. To search through a large collection of documents efficiently it is preferable to assign a numerical weight to each document individually. In this respect, users are often only interested in the most relevant documents rather than the entire ranked list. For example, in the case of most Internet searches it is likely that users will only want to look at the first 10 retrieved search results. Similarly, when retrieving documents, a user may only be interested in considering viewing the best top n documents. The focus of this paper is to provide a limited number of ranked documents to the end user. Alternatively, the user's satisfaction with the system may depend on how many documents they need to scrutinize until they find a relevant one. Therefore, this paper is concerned primarily with the retrieval of the most relevant documents according to information relevance, rather than with all of them.

Our approach to ranking of retrieved documents is centered on self-learning, the weighting function $tf \times idf$ with required adaptivity properties. This is in contrast to past strategies in IR which rely on viewing the documents as information overloads to obtain weighting function without considerations for underlying semantic analysis. The semantic similarities between terms in documents, which attracted the interest of many researchers who realize that viewing query terms as relevant information is limiting. Therefore, this paper takes advantage of query terms occurrences and self-learning to guide us in finding a weighting function that can automatically adjust its search structure to a user's query behaviour. In this regard, a good ranking criterion remains the choice of an IR system expert.

A. The Problem Formulation

Let us define this problem in the semantics analysis of the documents itself by self-learning. Assume that for a query q we have a set of documents $D \coloneqq \{d_{1,} \dots, d_l\}$ with associated relevance numerical weight values $v = \{v_1, \dots, v_n\}$ where $v_i \in \{0,1\}$ as normalization interval which prompt a relevance order among the documents $d_{i.}$ Here 1 is the maximum relevance numerical weight value corresponding to *'highly relevant'* and value 0 corresponds to '*irrelevant'*. For example, using the relevance context information $v_i > v_j$ implies that document d_i is preferable to document d_j. This will express user's degree of interest by pairwise comparison of documents. It is our goal to rank retrieved documents according to relevance numerical weight, such that the documents with relevance value v_i will show up at the commencement of the ranked list, rather than documents with relevance value v_j. This optimization of IR is obtained by ranking the documents according to a relevance numerical weight value $\omega(tf \times idf)$, which is obtained from the weighting function w in descending order. Then we wish to return a relevance numerical weight subset v_i of v such that for each $d_i \in$ D, we optimize the following weighting function:

$$\omega = (tf \times idf) \tag{1}$$

where, tf is the term frequency in the query-document pair, idf = log (N/ni), ni is the number of documents indexed containing term j; N is the total number of documents in the corpus.

Based on work reported in [14], [16], (1) notations suggest diverse approaches to this weighting function problem involve statistics to enhance retrieval effectiveness. This paper proposes a solution to the problem of ranking retrieved documents based on our proposed DROPT technique, as discussed in Section IV.

IV. Formalization of Mathematical Model Definitions for DROPT Algorithm

Based on (1), a DROPT measure for documents retrieved from a corpus is developed with respect to document index keywords and the query vectors. Naturally, given the notation we present for the problem, the use of statistical methods has proven both popular and efficient in responding to the problem formulation [14], [17]. 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.

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 numerical weight value $v \in \{0,1\}$, where v represents the relevance numerical weights of the retrieved documents. Each rank is assigned an ascending rank number n, such that:

$$R = [\{1, v_1\}, \{2, v_2\}, \dots, \{n, v_n\}], \textit{ where } v_1 > v_2 > \dots > v_{n.} \tag{2}$$

Our technique, DROPT is composed of six steps.

Step 1: Initialization of Parameters

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

$$Q = [q_{1,} q_{2,} q_{3,} \dots, q_l] \tag{3}$$

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:

$$D = \begin{bmatrix} d_{11}\, d_{12}\, d_{13} \cdots\cdots d_{1j} \\ d_{21}\, d_{22}\ \ d_{23} \cdots\cdots d_{2j} \\ \vdots \\ \vdots \\ d_{N1}\, d_{N2}\, d_{N3} \cdots d_{Nj} \end{bmatrix} \tag{4}$$

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

(c) We compute the convolution matrix, representing:

$$W = D \times Q = \begin{bmatrix} w_{11}\ w_{12}\ w_{13} \cdots w_{1l} \\ w_{21}\ w_{22}\ w_{23} \cdots w_{2l} \\ \vdots \\ \vdots \\ w_{n1}\ w_{n2}\ w_{n3} \cdots w_{nl} \end{bmatrix} \quad (5)$$

$w_{ij} = w_{jn} \times$ is equal string ignore case $(q_{ij},\ d_{kj})$, where q_{ij} are query vectors, d_{kj} are document vectors, w_{ij} are weights of terms in the document vectors, and w_{jn} are weights of terms in the query vectors, while n is the number of retrieved documents that are indexed by at least one keyword in the query vector. The matrix W gives a numeric measure with no context information.

Step 2: Search String Processing (Matching Mechanism)

The comparison of the issued query term against the document representation is called the query process. The matching process results are a list of potentially relevant context information. Individual users will scrutinize this document list in search of the information they need. The goal of context information acquisition should be to determine what a user is trying to achieve while performing his\her matching tasks. The search agent is used to retrieve documents in response to an issued query and return the best matching documents according to specific matching rules.

Step 3:Calculate Relevance Weight

Retrieved documents that are more relevant are ranked ahead of other documents that are less relevant. It is important to find relevance numerical weights of the retrieved documents and provide a ranked list to the user according to their information requests.

(a) Based on (1), the relevance weight is obtained according to document content.

(b) Subsequently we calculate the average mean weight using the weighted root mean squares (RMS) to determine the overall fitness value of retrieved documents with respect to a given query calculated as:

$$\overline{\omega} = \prod_{i=1}^{n} \frac{1}{l} \sqrt{\sum_{j=1}^{l} w_{ij}^2} \quad (6)$$

where, w is the average relevance mean weight of each retrieved document, n is the number of keywords terms occurrences in each retrieved document, l is the total size of the keywords in the corpus, and w_{ij} are the sum weights of terms of the document vectors.

Step 4:User Feedback about Retrieved Documents

User feedback about retrieved documents is based on overall relevance weights $\overline{w}$ to construct a personalized user profiling of interests. We can achieve this when a user indicates the documents that are relevant; otherwise, from the designated DBs context.

(a) The overall relevance judgment is given by:

$$G = [g_{ij}]_{\text{nxl}} \quad (7)$$

where, $g_{ij} = min\left(w_{ij}, q_{ij}\right)$ and $1 \le i \le n$, $1 \le j \le l$ and G is a query vector with a small-operator defined as a matrix, w_{ij} are weights of terms of the document vectors, and q_{ij} are queries vectors.

(b) Any numerical weight component of matrix G greater than the average mean weight, $\overline{w}$ (6) will be retained to add to a matrix T given by:

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

where,

$$\begin{cases} t_{ij} = g_{ij}, & if\ g_{ij} \ge \varpi \\ t_{ij} = 0, & if\ g_{ij} < \varpi \end{cases} \quad 1 \le i \le n, 1 \le j \le l$$

(c) Based on matrix T (8) we calculate relevance numerical weight values, for all set of documents D, which are the largest weighting values for each corresponding vector given by:

$$Val_i = max\left\{t_{ij}\right\}, 1 \le i \le n \quad 1 \le j \le l \quad (9)$$

(d) Thus, any document whose value Val_i was higher than the overall average relevance weight would be predicted as a relevant document; any document with a lower value would be predicted as irrelevant document (9). Thus average relevance mean value within the normalization interval $v \in \{0,1\}$ is computed for each document given by:

$$D = \left\{d | if\ val_i > 0,\ 1 \le i \le n\right\} \quad (10)$$

Step 5:Relevance Judgment

The individual user is asked to judge contextual factor (e.g. information relevance) influence on ranking given a certain contextual dimension (*numerical weight* is *relevant or irrelevant*).

(a) If the ranked document is relevant to user information needs, the user finishes his/her query search context, then GO to **Step 4** according to the user's document preference.

(b) Otherwise, the user continues to search the document DBs by reformulating the query or stop querying the designated DB until relevant documents are ranked. GO to **Step 6**.

Step 6:Update Term Weight and Keywords Set

The keyword term set n provided by the ranked documents and the relevance numerical weight values will be updated by user feedback.

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

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

$$w_{ij} = (w_{ij})\beta, \quad (11)$$

where, $0 < \beta < 1, i \in \{i \,|d_i \in D\}$ and $j \in \{j \,|q_j = 1\}$

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 can improve the ranking mechanism for the search results in an attempt to adapt the retrieval environment of the users and amount of relevant context-aware 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.

V. Experimental Results

A Search engine back end prototype (which was wrapped around Google) is created for the domain of three systems user experts. To evaluate the proposed DROPT technique, we performed an experiment on small scale search of 30 queries to validate the effectiveness of the technique. In this section, we have presented the results obtained for the proposed technique. The experimental methodology is described in Section V-A. The results and discussions are presented in Section V-B, while Section V-C discussed the ranking performance results. Performance evaluations are given in Section V-D.

A. Methodology

We involved three system users (all PhD students) to collect data through the WampServer search engine back end prototype. The three study system user participants were given 10 search tasks each in their domain of knowledge. During the search context, the students' interactions with the search engine back end prototype were logged via the system log in menu with their "*student identification number*". In each task, the students were asked to obtain the frequency of keyword matching based querying across a document that were relevant to meet their information requests to achieve document ranking task based on individual users' preference, or ignore documents that were found to be irrelevant. The user behavioural measures we examine are the frequencies of the issued query (i.e. frequency of keyword matching based querying). The function of the frequency of the keyword across a document from the document DB collected and stores in the WampServer site localhost DB. WampServer is a Windows Internet environment that allows user to create Internet applications with Apache 2, PHP and a Mysql DB. PHP Myadmin allows user to manage easily our DBs. This measure was used to predict the 'relevant" documents marked 'X" for document ranking model. To evaluate the performance of the proposed technique, we have considered different 30 queries from the system users. Table I gives the statistics of the queries considered in the experiment.

B. Results and Discussions

In order to generate the prediction user model of the document ranking context, we used the $tf \times idf$ weighting method to calculate the most relevant document marked X as crucial predictors (relevance numerical weights) to generate data for the adaptation of retrieved documents. For the document ranking models generated according to relevance weights. The average relevance weights of individual users were obtained as ϖ = 0.663 for Domain 1, ϖ = 0.85 for Domain 2, ϖ = 0.735 for Domain 3, and the overall average relevance weight, ϖ = 0.75 was obtained for the three Domains of experts combined. Thus for Domain 1, any document whose value was higher than 0.663 would be predicted for ranking as a "relevant" document, and marked 'X'; and any document with a lower value would be predicted but ignored if found to be "irrelevant" later. Also, for Domain 2, any document whose value was higher than 0.850 would be predicted for ranking as a 'relevant' document and marked 'X'; and any document with a lower value would be predicted but ignored if found to be "irrelevant" later. Lastly, for Domain 3, any document whose value was higher than 0.735 would be predicted for ranking as a "relevant" document, and marked 'X'; and any document with a lower value would be predicted but ignored if found to be "irrelevant" later. We generated three prediction models, each from domain of experts with different generated data from the user behaviour measure context when the matching tasks were considered. This shows that any document whose value was higher than 0.75 would be predicted for ranking performance results at known "relevant" document, and marked 'X'; and any document with a lower value would be predicted but ignored if found to be irrelevant later. Our goal is to appropriately predict "relevant documents" for ranking performance results based on user preference. Therefore, we measured precision and recall of relevant documents, marked 'X'.

Our results on the indexed keywords represent the domain of the system user's experts (three PhD students) in an in-lab experimental setting. The results demonstrate that combining individual system user's behavioural measures can improve ranking prediction accuracy (according to relevance weights), for documents ranking, and however that individual users ranking performed much better than combining document rankings of the systems. This accomplishes the adaptation of retrieved documents for individual users as the focus of this paper. The retrieval effectiveness is measured using well-known metrics at known relevant documents.

C. Ranking Performance Results

With the intention of measure ranking performance, the DROPT technique for ranking search results list was tuned by experimenting with the prototype 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 the 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 judge to be irrelevant (less preferred).

According to Table I, the values displayed in Fig. 1 shows the 30 search results of the proposed technique for documents retrieved from a localhost DB search engine back-end. Documents are sorted and were set in ascending order of Retrieval Status Values (RSV). Hence, documents whose relevance weight fall above the (linear weight) with overall weighted fitness score (F=0.75), as shown in Fig. 1, are considered as relevant documents. Hence, 19 documents are ranked and given to users to meet their information needs. Conversely, the 11 search results that fall below (linear weight) are rejected by the user (not displayed), as shown in Fig. 2.

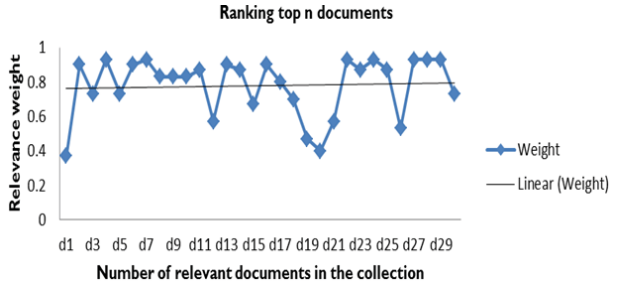

Fig. 2 Ranking performance graph results at the known relevant documents

D. Comparison of DROPT Technique with BM25 and TF-IDF Methods

This section presents the results that show the performance of our technique against some previously published methods. We compared 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 "Precision at position n" (P@n) measure. For the information needs and document collection of the experiment, relevance was assessed by different experienced system users in their domain of experts (three 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 results, we have used P@n metrics [18]. Precision at *n* measures the relevancy of the top *n* results of the ranking list with respect to a given query (12).

$$P@n = \text{Number of relevant document in top } n \text{ results} / n \quad (12)$$

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, 30 queries were judged in these six levels by users.

For the evaluation of the algorithm, testing of the prototype system was conducted. The test process involved using the 30 queries provided by the system users. The measure (P@n) is used for the evaluation. Naturally, this is computed for each query, and then takes the average dimension (*n*) for all queries. Fig. 3 shows the comparison of the DROPT algorithm with other algorithms in the P@n measure. As the figure shows, our adaptive algorithm outperforms the others. The 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 queries set between 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 context-based algorithms using user preferences for query reformulations. In this regard, the number of top *n* results showed to users depicts the relevancy degree of the retrieved documents with respect to a given query with rank *n* (judged by the system users).

TABLE I
PRECISION RESULTS FROM THE 3 DOMAINS OF EXPERT FOR RANKING PERFORMANCE AT KNOWN RELEVANT DOCUMENTS

Document #	Queries	Relevant	Tf	Precision	Fitness Score
1	IR		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

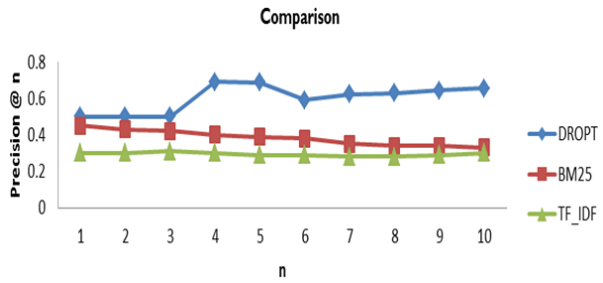

Fig. 3 Comparison of DROPT with BM25 and TF-IDF in the P@n measure

VI. Conclusion

In this paper, we have proposed a new document ranking technique in IR with the intention of retrieving context information ranked according to information relevance. The proposed adaptive DROPT technique is to improve the relevance of retrieved documents. The DROPT technique adapts itself with individual user information needs based on environment and search context. In this respect, our algorithm can judge the relevance of the current search results according to the relevance weight of the retrieved documents. The technique demonstrated in providing a limited number of ranked documents in response to a given users' query. The new DROPT technique combined approaches suitable for user profiling and mining of user interest for the enhancement of IR system performance, which satisfy the focus of this paper. The DROPT technique is designed purposely to overcome some of the limitations (e.g. low precision and recall, and not adaptive to users) of existing traditional ranking $tf \times idf$ algorithms that ignore the semantic analysis of the document itself. We have used content-based algorithms such as TF-IDF and BM25 as baselines in comparison with our DROPT algorithm for user preferences to predict our ranked documents. We used the search engine back-end prototype as an interaction between the information user and information source for document collections and 30 queries from the domain of the three system user experts for evaluation of our DROPT technique. The evaluation was carried out in an in-lab setting, whereby the number of relevant and non-relevant documents in the localhost site was known. Evaluation of the DROPT technique shows performance results improvement using 'precision at position n' i.e. P@n, over the chosen baseline algorithms methods. The proposed DROPT algorithm has some interesting features like scalability and adaptability. It is scalable in that any new algorithm can easily be added for comparison and also adaptable, in that it adapts itself based on user information needs within the environment.

References

[1] McGrayne, S. B. The Theory That Would Not Die: How Bayes' Rule Cracked the Enigma Code, Hunted Down Russian Submarines, and Emerged Triumphant from Two Centuries of Controversy. Yale University Press, 2012.

[2] Sasarak, C., Hart, K., Pospesel, R., Stalnaker, D., Hu, L., Livolsi, R., Zhu, S., and Zanibbi, R. min: A multimodal WWW interface for math search. In Symp. Human-Computer Interaction and Information Retrieval (HCIR) (Cambridge, MA, Oct. 2012).

[3] Lewandowski D. Search engine user behaviour: how can users be guided to quality content? *Information Services and Use*, 2008, Vol.28, No.3-4 pp. 261-268.

[4] Jansen BJ and Molina P. The Effectiveness of Web Search Engines for Retrieving Relevant Ecommerce Links, *Information Processing & Management*, 2006, vol. 42, no. 4, pp. 1075-1098.

[5] Jansen BJ, Spink A and Saracevic T. Real life, real users, and real needs: a study and analysis of user queries on the Web. *Inf. Process. Manag.*, 2000, Vol. 36, pp: 207–227.

[6] Shen C, Kim J, Wang L and van den Hengel A. Positive semidefinite metric learning using boosting-like algorithms. *Journal of Machine Learning Research*, 2012, Vol. 13, pp. 1007–1036.

[7] Shivaswamy PK and Joachims T. Online Learning with Preference Feedback. In NIPS workshop on Choice Models and Preference Learning, 2011 edition.

[8] Nyongesa HO and Maleki-dizaji S. User modelling using evolutionary interactive reinforcement learning. *Inf Retrieval*, 2006, vol. 9, no. 3, pp. 343-355. DOI: 10.1007/s10791-006-4536-3

[9] Hjorland B. The Foundation of the Concept of Relevance. *Journal of the American Society for Information Science and Technology*, 2010, Vol. 61, No. 2, pp. 217-237.

[10] Grady C and Lease M. Crowdsourcing Document Relevance Assessment with Mechanical Turk. In Proceedings of the NAACL 2010 workshop on Creating Speech and Language Data with Amazon Mechanical Turk, Los-Angelis, California, 2010, pp. 172-179.

[11] Saracevic T. Relevance: A review of the literature and a framework for thinking on the notion in information science. Part III: Behavior and effects of relevance. *Journal of the American Society for Information Science and Technology*, 2007, Vol. 58, No.13, pp. 2126-2144.

[12] Borlund P. The Concept of Relevance in IR. *Journal of the American Society for Information Science and Technology*, 2003, Vol. 54, No. 10, pp. 913-925.

[13] Setchi R, Tang Q and Stankov I. Semantic-based information retrieval in support of concept design. In: Proceedings of *Advanced Engineering Informatics*, 2011, pp.131-146.

[14] Baeza-Yates R and Ribeiro-Neto B. Modern Information Retrieval: The Concepts and Technology Behind Search, 2nd edn. 2011, Addison-Wesley, Reading.

[15] Manning CD, Raghavan P and Schutze H. Introduction to Information Retrieval. Cambridge University Press, 2008, Cambridge.

[16] Salton G and Buckley C. Term-Weighting approaches in automatic text retrieval, *Information Processing and Management*, 1988, Vol. 24, Issue 5, pp. 513-523.

[17] Agichtein E, Brill E and Dumais S. Improving Web search ranking by incorporating user behavior information. In: 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2006), 2006a, pp. 19–26. ACM, Seattle.

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

[19] Agbele K. Context-Awareness for Adaptive Information Retrieval Systems, Unpublished PhD Thesis, University of the Western Cape, 2014.