ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering and Technology (ICIET'14)
On 21st & 22nd March Organized by

K.L.N. College of Engineering, Madurai, Tamil Nadu, India

Ranking Documents in IR Using Vector Based Ordering In E-Learning

T.Suganya [1], M.Ravichandran [2]

[1] Department of CSE-IT, Arunai Engineering College, Tiruvannamalai, Tamilnadu, India

[2] Department of Information Technology, Arunai Engineering College,
Tiruvannamalai, Tamilnadu, India

ABSTRACT— E-Learning provides enormous collection of e-learning materials to the users. But, Most of the retrieved learning materials may be irrelevant to the query posted by the users. The Users spent lot of time to retrieve the pertinent learning materials in the largest domain. Due to this the learning process of a learner is slowed down. Hence, there is a need to develop an efficient retrieval and ranking method in the information learning system. In classical information retrieval model, various strategies were used to rank the documents. These methods ranked the documents based on the retrieval status value which can be computed by using various aggregation operators. These methods rank order the documents without considering the importance of individual term relevance. This paper presents a technique called vector based possibility framework to enhance the performance of classical information retrieval method. This proposed system provides highly relevant learning materials to the learner and it recommends the items based on individual term relevance with respect to the query specified by the user

INDEX TERMS—E-Learning, Information Retrieval, Ranking Algorithm, Possibility vector, Necessity vector

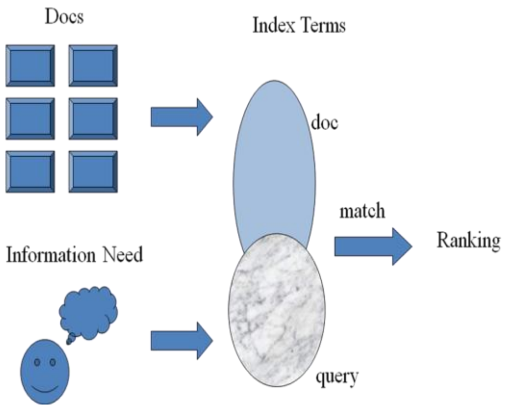
Introduction

E-Learning is an innovative way which enhances the traditional learning system. It enables busy people to learn new technologies at anytime and anywhere. It can include training, the delivery of just-in-time information and guidance from experts. E-learning includes various types of media that deliver text, audio, images, animation, and includes technology applications. Content is a core component of e-learning and includes issues such as pedagogy and learning object re-use.

Pedagogical element is the basic unit of e-learning material. These elements are the educational content that is to be delivered. These pedagogical structures themselves are not the textbook, web page, Podcast, lesson, assignment, multiple choice questions, quiz, discussion group or a case study.

An information retrieval system is an application that stores and manages information on documents, often textual documents but possibly multimedia. The goal of information retrieval (IR) is to provide users with those documents that satisfy their information need. We use the word "document" as a general term that could also include non-textual information, such as multimedia objects.

Traditional information retrieval systems usually adopt index terms to index and retrieve documents. An index term is a keyword (or group of related words) which has some meaning of its own (usually a noun).Ranking is an ordering of the documents retrieved that (hopefully) reflects the relevance of the documents to the user query.Ranking algorithms are at the core of information retrieval systems (predicting which documents are relevant and which are not).The ranking is based on fundamental premises regarding the notion of relevance, such as common sets of index terms,sharing of weighted terms, likelihood of relevance.Each set of premises leads to a distinct IR model.



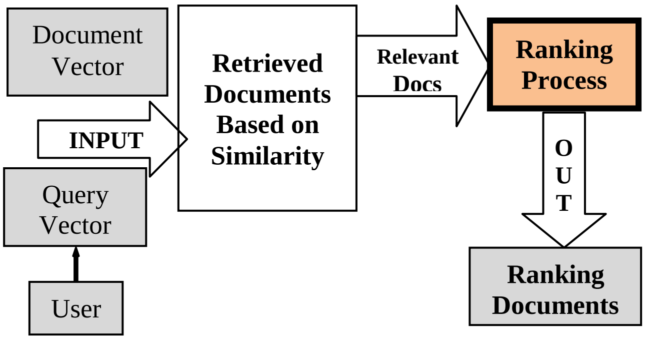
Document & Index Terms

In Classical IR methods each document is represented as a set of representative keywords or index terms.An index term is a document word useful for remembering the document main themes. Not all terms are equally useful for representing the document contents, less frequent terms allow identifying a narrower set of documents.The importance of the index terms is represented by weights associated to them.Document may be of any type of learning item (Journal paper, book, e-mail messages, web page).

The Generic information retrieval systems select and return the desired documents to the user from a large set of documents in accordance with criteria specified by the user. It performs various retrieval functions. They are document search and document routing (filtering).The document search (ad-hoc) method selects the documents from an existing collection of documents where as document routing (filtering) distributes incoming documents to appropriate users on the basis of user interest profiles. The central problem regarding IR systems is the issue of predicting which documents are relevant and which are not. Ranking algorithms are at the core of IR systems. The ranking algorithm orders the learning materials based on the relevance according to distinct IR model**.** The documents and queries are represented as vectors. These vectors are compared to retrieve the relevant documents having the query terms as shown in fig1.

The main aim of the information retrieval system in e-learning is to provide the learning materials what exactly the users specified in their query. Most of the users prefer a top most material from a list of materials which is exposed to them. Hence, efficient ranking should be performed to order the relevant materials.

The input to the IR system is query vector and document collection (shown in fig2). Query terms are usually weighted in order to allow the user to express their preferences and assess the importance of each term. Therefore, the result of query evaluation on a document is a vector. This vector has a set of weight values of terms in the retrieved document, usually modified for taking into account preferences about the importance of the terms in the query. The document vector has set of weighted terms. These weight values are computed by term weighting scheme.



Information Retrieval System

The query and document vectors are compared to find similarity between them. Documents having the terms present in query are retrieved. The documents having all the terms or any of the terms with respect to the query are considered as relevant documents (based on the possible condition stated by user). After retrieving the relevant documents, they are given as input to the ranking process. Then the ranked documents are provided to the user.

PRIOR WORK

In classical information retrieval system, various aggregation schemes were used to rank order the documents. This type of approach frequently drops the valuable information and reduces the discriminating power between the documents. These methods combine all the individual keyword values together while ranking. In classical information retrieval systems, documents and user queries are represented by sets of weighted terms. Term weights are computed by statistical analysis.

To evaluate to what extent a document is relevant to the query, the retrieval status value (rsv) is computed by aggregating the weights for the terms present in the query. Then the documents are ranked on the basis of the *rsv*. Various aggregation operators used in finding the rsv value. The candidate operators are similarity-based evaluation, average (mean value), *p*-norms [2, 3,4]. Since these operators combine all the individual keyword weights of the retrieved relevant documents, it is impossible to discriminate the documents having same global relevance value.

As an example, let us consider a three terms query. To evaluate the query, average aggregation method is used and the rsv value is computed. This is only an example, and remarks similar to the ones below apply to other aggregation operators including *min* and other fuzzy logic connectives. Let us assume that the evaluation of the query q=*t1 ^ t2 ^ t3* on two documents *d*1 and *d*2 gives the following results.

$$
\begin{aligned}
rsv(q,d1) &= w(t1,d1)+w(t2,d1)+w(t3,d1)/3 \\
&=0.2+0.8+0.8/3 \\
&=0.6 \\
rsv(q,d2) &= w(t1,d2)+w(t2,d2)+w(t3,d2)/3 \\
&=0.6+0.6+0.6/3 \\
&=0.6
\end{aligned}
$$

In the above example, both documents having same retrieval status value with different weighted terms. If the user mostly prefers the first term (t1 than others) then the ordering of documents d1 over d2 will not give up a best one to the user. Hence, problem arises when ranking document d1 before d2.Here, the main concern is to know whether the user prefers a document with a medium relevance for all her criteria, or having a high relevance for most of her criteria. This type of approach explained above does not yield any key to find the two aspects of relevance (possibly relevant or certainly relevant with respect to the query) and it does not offer an efficient implementation for millions of relevant document collection.

In this paper, instead of aggregating the weights of the keywords, the individual term relevance degree vectors are used to find *rsv*. The *rsv* is computed by aggregating the two vectors namely, possibility vector and necessity vector.

Criteria Formulation

The first attempt to retrieve relevant information is to formulate a query. A query is composed of keywords and the documents containing such keywords are searched for. Keyword based queries are very popular, easy to express and allows the system to do fast ranking. Thus a query can be simply a word or combination of complex words. Users can express their criteria using the keywords with some quantifiers and conditions (conjunction or disjunction operators).

For an example, if user wants to retrieve the documents associated with the terms database, sql and index, then he can express his query as, *Q= most of (database) ^ sql ^ index*. In this example, user stated that the documents having the three terms with non zero frequency should be obtained. The *most of* quantifier allows the user to express his preferences.

At least two approaches can be used to compare objects according to multiple criteria. The first one is to aggregate these criteria, then to compare the obtained values. This corresponds to the classical information retrieval approach, considering individual query term relevance as a criterion to fulfill. The second method amounts to compare the criteria evaluation vectors directly by using a refinement of Pareto ordering ((t_1. . . t_n) >Pareto (t'_1, . . . , t'_n) if $\forall$i, $t_i \geq t'_i$ and $\exists_j$, $t_j > t'_j$ *)*.

In the proposed model, possibility & necessity vector values are aggregated to find relevance of a document.

Refinement of Pareto ordering

After formulating the multi criteria function, the full weight vectors are compared using the pare to ordering. The two refinements of Pareto ordering are *discrimin* & *leximin* [5, 11]. They are used to differentiate the vectors having same minimal value.

Discrimin: Two evaluation vectors are compared using only their distinct components. Thus, identical values having the same place in both vectors are dropped before aggregating the remaining values with a conjunction operator. Thus only discriminating term weights are considered. In the context of information retrieval, given two vectors representing the weights of terms in query *q* for document *d*1 and *d*2. For instance,

$$
\begin{aligned}
rsv\ (q, d1) &= (1, 0.5, 0.1, 0.2), \\
rsv\ (q, d2) &= (0.2, 0.7, 0.1, 1)
\end{aligned}
$$

The *discrimin* procedure “drops” the third term and ranking these documents based on remaining values.

Leximin: It is a *discrimin* applied on vectors with increasingly re-ordered components. Considering two vectors,

$$
\begin{aligned}
rsv\ (q, d1) &= (1, 0.5, 0.1, 0.2), \\
rsv\ (q, d2) &= (0.2, 0.7, 0.1, 1).
\end{aligned}
$$

Leximin sorts the values before comparing them. The discrimin drops the identical values in same place. The values (0.1, 0.2 and 1) are dropped. The evaluation of result becomes rsv (*q*, *d*2) = 0.7 and *rsv*(*q*, *d*1) = 0.5 and then ranking the document d2 before d1.

$$
\begin{aligned}
rsv\ (q, d1) &= 0.5 \\
rsv\ (q, d2) &= 0.7
\end{aligned}
$$

PROPOSED MODEL

The proposed system focuses on the individual terms in relevant documents by applying possibilistic logic for ranking documents. Possibility and Necessity vectors are framed by setting threshold value called $\alpha(\alpha \in \{0,1\})$. Our approach tries to distinguish the terms which are possibly representative of documents and those which are necessarily representative of documents, i.e. terms which suffice to characterize documents. This is a mathematically based approach and easy to implement for larger collections.

Retrieving relevant documents

Both the query vector and document vector are compared to find similarity between them. If a document fulfills all the criteria stated in the query, then it is considered as a relevant document to the query and the relevant document is retrieved from the collection. For these relevant documents, threshold value is computed for each term corresponding to the query.

Setting Threshold Value

The improvement of this proposed method strongly depends on the factor α. The value can be estimated from the frequency *tf* [10],

$$\alpha = \mathrm{ntf}_{ij} = \frac{\mathrm{tf}_{ij}}{\mathrm{Max}\ \forall_{tk\epsilon dj}(\mathrm{tf}_{kj}).} \qquad (1)$$

Where ntfij is *normalized term frequency*, tf_{ij} is the term frequency for term *i* in the document *j* and *max* $\forall_{tk\epsilon dj}(tf_{kj})$ is the maximum term frequency of that document.

Possibility & Necessity Vectors

In this paper, we will use the possibilistic model [1].In this approach, the document relevance for the query is given by a pair of possibility and necessity degrees computed from α. The retrieval status value is then a pair,

$$rsv\ (q, d) = (\Pi(q, d), N(q, d))$$

rsv(q,d) represents to what extend it is possible or certain that *d* is relevant with respect to *q*. To use this possibilistic model, the possibility and necessity vectors are framed for the matching terms by taking into account the statistical weights of the terms in the document. A simple, parameterized, way to assess the possibility and the necessity degrees (resp. *Π* and *N*) is to use the following piecewise linear transformation[1]

$$\Pi\ (t, d) = \begin{cases} 0 & \text{if wt} = 0 \\ 1 & \text{if wt} \geq \alpha \\ \mathrm{Wt}/\alpha & \text{otherwise} \end{cases} \qquad (2)$$

$$N\ (t, d) = \begin{cases} 1 & \text{if wt} = 1 \\ \mathrm{wt}-\alpha/1-\alpha & \text{if } \alpha < 1 \text{ and wt} \geq \alpha \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

Thus, the evaluation of a conjunctive query *q* involving $t_1, \ldots, t_n$ amounts to compute the pair of vectors $(\Pi(t_1, d), \ldots, \Pi(t_n, d))$ and $(N(t_1, d), \ldots, N(t_n, d))$. Then documents are ordered by applying first the leximin/discrimin ranking procedure on the *N*-vectors, and in case of ties, the leximin/discrimin is applied to the corresponding *Π*-vectors to try to refine the ordering.

EXPERIMENTS AND RESULTS

In this section, we present results of some experiments on a simple document collection in order to evaluate the merit of the vector-based ranking of documents. Moreover, the impact of the possibilistic encoding of the term weights in the document is first discussed.

Description of the Experiments

The goal of the experiment is to enhance the global performance of the information retrieval system, and to compare the results that are obtained using several ranking methods with the proposed one. The first experiment compares results obtained with conjunction aggregation operator, namely the weighted sum aggregation underlying the classical approach with the vector based ranking method. The second experiment analysis the performance of possibility framework suggested in [1] with the proposed approach, in order to determine the best *α* value. This experiment is used to improve the overall performance of classical information retrieval system.

Dataset Information

To analysis the performance of this proposed model, a small dataset with non binary term frequency term-document matrix is taken as input. Weight for the terms is computed using a formula derived from Okapi system [6, 7].

$$w_t = \frac{(\mathrm{k1}+1).*\mathrm{tf}}{\mathrm{k1}\ (\ (1\text{-}b) + (b * \mathrm{dl}/\Delta l)) + \mathrm{tf}.} * \mathrm{idf}$$

where *tf* is the term frequency in the document, *dl* is the document length, ∆*l* is the average document length in the collection, idf is the inverse document frequency computed from log (N/ni) (N - total number of documents in the collection, ni – the number of documents having the term ti) and k1 and b are global parameters which are constant values. it can be tuned on the basis of evaluation data. The k1value is 0 if it is a binary (tf ϵ {0, 1}) model and 1 if non binary (tf is in between o and 1) model.

For a given query q, the relevant documents are retrieved based on the similarity degree. If the query q is in the following form,

		α=0.2	α=0.6	α=0.8	
t1^ t2	D5,D1, D3,D2,D4	CR D5,D1 D3,D2 D4	CR D5,D1 D3,D2 PR D4	CR D5,D1 D3, PR D2,D4	CR D5, D2, D1 PR D3, D4
t1^ t3	D5,D1, D2,D3,D4	CR D5,D1 D2,D3 PR D4	CR D5 PR D1,D2, D3,D4	PR D5,D1 D2,D3 D4	CR D5, D2 D1, D3 PR D4
t2^ t3	D5,D1,D3 D2,D4	CR D5,D1 D3,D2, D4	CR D5,D1 D3,D2 PR D4	CR D5,D1 D3 PR D2,D4	CR D5, D2 D1, D3 PR D4

q = t1 ^ t2

Then t1, t2 represents two unique terms and the possible condition is min. Hence, documents having both the terms with non-zero term frequency must be obtained.

The similarity degree S_{qd} between a query q and a document d giving the relevance of the document for the query, is computed as

$$Sqd = \Sigma_{t\epsilon q} \quad \lambda_t \times w_{td}$$

Where λt is an importance weight for the term in the query (here always 1) and w_{td} is the index term weight for the document d.

Comparison of ranking methods

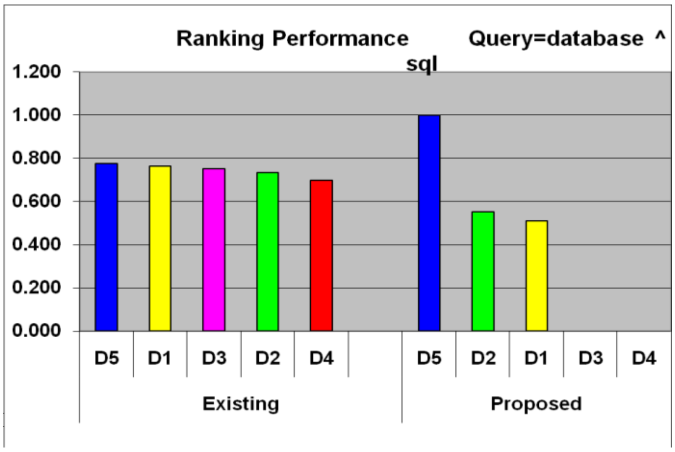
Q	Existing System	Proposed System	
	Weight Aggregation	Possibilistic logic & Vector based ordering	
	Average Operator	**α value chosen for all documents as stated in [1]**	***α= normalized term frequency***

We will now evaluate the ranking method used in existing & proposed stated in this paper. To apply these ordered weightings, the vectors containing the weights of each query term in the document are decreasingly ordered. The queries considered here introduce further preference levels with the help of *most of- like* operator. This type of operator gives more importance to the highest term weights minimizing the impact of the lowest ones. The weighting vector is computed according to the query length. Results are then sorted using (*Π, N*) values modified by the weight *wi*.

The Table 1 shows the rank ordering of documents relevant to the user queries, obtained by the three different methods. In three methods are, weight aggregation using averaging operator (existing system approach), same α values for all document suggested in [1] and formula for α stated in equation (1). The equation (1) provides different α values for each term in the document and their value varies with documents.

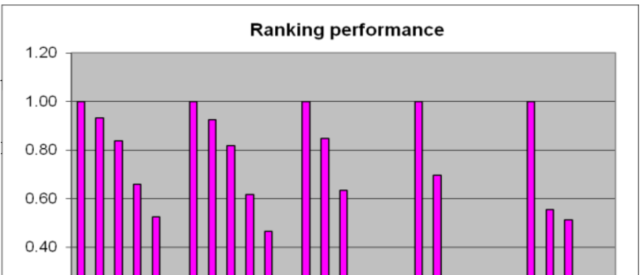
Table 1: Rank Ordering of Documents to User Query

PR – possibly relevant CR - Certainly relevant Q – Query



The experiment results shown in fig. 3 and fig. 4.The first experiment compare the performance of proposed against existing (Classical-weight aggregation method). For a given query, the proposed one show the certainly relevant documents in order (D5, D2, D1), the documents D3 and D4 are only possibly relevant to the query.

The second experiment is used to study the performance of proposed against possibility framework used in[1]. For a given query, the proposed one shows the certainly relevant documents in the order (D5, D2,D1), the documents D3 and D4 are only possibly relevant to the query.



Various **α values Vs normalized frequency**

CONCLUSION

The new approach stated in this paper, rank ordering the documents according to their individual term relevance degree using possibility approach and vector-based technique. This approach was evaluated on a subset of a document collection. We compared the refined rank-ordering approach with the classical approach based on relevance scores aggregated by a weighted sum. These experiments suggest the effectiveness of the refined rank-ordering approach, as it outperforms weight aggregation methods to some extent. These first preliminary results indicate that ranking documents can take advantage of the individual term weight vector of a document, rather than using an aggregated value.

The improvement of this approach depends on the term weighting scheme and the threshold value. In future works, we plan to evaluate this approach on larger collections such as TREC collections, and secondly to explore other variants of the flexible aggregation ranking techniques. This approach is not restricted to textual IR, but could be applied to any document retrieval system using several criteria for describing them, such as in video or audio resources in e-learning.

REFERENCES

[1] Mohand Boughanem, Yannick Loiseau, and Henri Prade, “Rank-Ordering Documents According to TheirRelevance in Information Retrieval Using Refinements of Ordered-Weighted Aggregations**”** LNCS 3877, pp. 44–54, 2006.

[2] lton, G., Fox, E., Wu, H.: “Extended boolean information retrieval”. Communications of the ACM **26** (1983) 1022–1036

[3] Robertson, S.E.: The probability ranking principle in IR. Journal of Documentation **33** (1977) 294–304

[4] Yager, R.: “On ordered weighted averaging aggregation operators in multicriteria decision making. IEEE Transactions on Systems”, Man and Cybernetics **18** (1988) 183–190

[5] Dubois, D., Fargier, H., Prade, H.: *Beyond min aggregation in multicriteria decision: (ordered) weighted min, discri-min, and leximin*. In Yager, R., Kacprzyk, J., eds.: The Ordered Weighted Averaging Operators. Kluwer (1997) 181–192

[6] Boughanem, M., Dkaki, T., Mothe, J., Soule-Dupuy, C.: Mercure at TREC-7. In: Proc. of TREC-7. (1998) 135–141

[7] Robertson, S.E., Walker, S.: Okapi-keenbow at TREC-8. In: Proc. 8th Text Retrieval Conf., TREC-8 (1999) 60–67

[8] Porter, M.: “An algorithm for suffix stripping”. Program **14** (1980) 130–137

[9] J¨arvelin, K., Kek¨al¨ainen, J.: Ir evaluation methods for retrieving highly relevant documents. In Belkin, N., Ingwersen, P., Leong, M.K., eds.: Proc. of the 23rd ACM Sigir Conf. on Research and Development of Information Retrieval, Athens, Greece, ACM Press, N.Y (2000) 41–48

[10] A Model for Information Retrieval based on Possibilistic networks Asma H. BRINI and Mohand Boughanem and Didier Dubois Irit 118, route de Narbonne, Cedex 4 Toulouse, FRANCE

[11] “Refinements of minimum based ordering in between discrimin and leximin” – Henri prade