Expert Systems with Applications 42 (2015) 1223–1234

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A new fuzzy logic based ranking function for efficient Information Retrieval system

Yogesh Gupta, Ashish Saini *, A.K. Saxena

Department of Electrical Engineering, Faculty of Engineering, Dayalbagh Educational Institute, Agra 282110, Uttar Pradesh, India

ARTICLE INFO

Article history:
Available online 19 September 2014

Keywords:
Information Retrieval
Fuzzy Logic Controller
Precision
Recall
Ranking function

ABSTRACT

The relevant documents from large data sets are retrieved with the help of ranking function in Information Retrieval system. In this paper, a new fuzzy logic based ranking function is proposed and implemented to enhance the performance of Information Retrieval system. The proposed ranking function is based on the computation of different terms of term-weighting schema such as term frequency, inverse document frequency and normalization. Fuzzy logic is used at two levels to compute relevance score of a document with respect to the query in present work. All the experiments are performed on *CACM* and *CISI* benchmark data sets. The experimental results reveal that the performance of our proposed ranking function is much better than the fuzzy based ranking function developed by Rubens along with other widely used ranking function *Okapi-BM25* in terms of precision, recall and F-measure.

1. Introduction

In recent times, Information Retrieval (IR) has become an important area of research in computer science. IR systems are used in several application domains such as web search, digital library search, blog search, information filtering, recommender system and social search, etc. The major concern of IR is to find "relevant" information or documents with respect to user need, modeled through a query from large data corpus in appropriate time interval (Salton & McGill, 1983; Yates & Berthier, 1999). IR system uses ranking function to retrieve relevant documents by computing relevance score between a query and a document. Although the conventional statistical ranking functions such as *Cosine, Jaccard* (Salton, 1998), *Euclidean* and *Okapi* (Robertson, Walker, & Beaulieu, 1999) have been extensively used but these measures fail to capture inherent features of documents and queries due to subjectivity involved in natural language text.

Natural language is often vague and uncertain (Subtil, Mouaddib, & Faucout, 1996). It is very difficult to determine something that is uncertain and vague with crisp formulas and crisp logics. Therefore, fuzzy logic (Zadeh, 1965) is found very suitable, to handle this uncertainty, vagueness and impreciseness. It transforms vagueness and uncertainty of documents, queries and their characteristics into fuzzy membership functions (Zadeh, 1997). The documents are retrieved by query with the help of the rules framed in Fuzzy Inference System (FIS) (Abraham, Lihong, & Zhiqiang, 1992; Jang & Sun, 1997; Ross, 1997; Sugeno, 1985a; Zadeh, 1997). Fuzzy logic uses degrees of memberships to express relevance unlike the Binary/Boolean model which is based on binary decision criterion i.e. {relevant, not relevant}.

In the present paper, a new fuzzy logic based ranking function is proposed. The performance of proposed ranking function is compared with *Okapi-BM25* and Rubens' ranking function (Rubens, 2006). Vector Space Model (VSM) is used as an IR model to develop proposed ranking function due to its strengths over other models, which are explained in Section 2. The main contributions of this paper are following:

- The proposed ranking function is based on composite *FIS*, which has two levels: *first level FIS* and *second level FIS*. *First level FIS* consists of two Fuzzy Logic Controllers (*FLCs*). First *FLC* is for structuring the features of documents and second *FLC* is for structuring the features of queries. *Second level FIS* consists of one *FLC*.
- The proposed ranking function retrieves the relevant documents on the basis of different variables; those capture the features of documents and queries as well.
- New fuzzy rules are framed in this paper for each *FLC* at each level of *FIS*.
- New linguistics variables are used to transform existing knowledge and information into fuzzy rules.

The rest of the paper is structured as follows. In Section 2, a brief description of VSM and work related to the already developed

* Corresponding author. Tel.: +91 562 2801224; fax: +91 562 2801226.
E-mail address: ashish7119@gmail.com (A. Saini).

http://dx.doi.org/10.1016/j.eswa.2014.09.009
0957-4174/

ranking functions model are presented to form the necessary theoretical foundation for this work. The details of proposed fuzzy logic based ranking function and comparison of its important features with Rubens' approach are presented in Section 3. In Section 4, the experimental results and analysis are discussed. Finally, conclusion and future directions are drawn in Section 5.

2. Related work and theoretical foundation

There are different factors, which affect the performance of an IR system, but ranking function is one which affects the most (Lancaster & Warner, 1993). Ranking functions match the documents or information to a user's query and rank them according to the relevance score in descending order. The documents and queries both need to be transformed into a model that can be effectively processed by computers to facilitate this relevance estimation process. VSM (Cordon, Moya, & Zarco, 2004; Haase, Steinmann, & Vejda, 2002; Harman, 1993; Jones & Furnas, 1987; Mercier & Beigbeder, 2005; Robertson, 1997; Salton & Buckley, 1988; Witten, Moffat, & Bell, 1999; Yap & Wu, 2005; Yates & Berthier, 1999) is considered as one of the most successful IR models.

This section describes various advantages of VSM using as an IR model and its important features followed by the discussion on the literature of ranking functions.

2.1. Vector Space Model

Vector Space Model is used as an IR model in present paper to develop the proposed ranking function because of following advantages:

- It is simple and fast model as documents and queries are represented in the form of vectors in n-dimensional space, where n is the number of unique terms used to describe the contents of documents and queries (Cordon, Viedma, Pujalte, Luque, & Zarco, 2003). Therefore, the properties of these vectors such as similarity and closeness can be studied easily.
- It can handle weighted terms.
- It produces a ranked list as output and that the indexing process is automated which means a significantly lighter workload for the administrator of the collection.
- It is easy to modify individual vectors, which is essential for the query expansion technique and logic based ranking functions.

VSM is based on the assumption that the relevance of a document with respect to a query is correlated with the distance between that query and document. A block schematic of queries and documents represented as vectors in VSM, is shown in Fig. 1.

The representation of documents and queries can be extended by including their features. An empirically validated document feature is the number of term occurrences within a document (term frequency or *tf*) (Salton, 1968). The intuitive justification for this feature is that a document that notifies a term more often is more likely to be relevant for that term. Another important feature is the potential for a term to discriminate between documents, named as inverse document frequency (or *idf*) (Jones, 1972). This particular feature (*idf*) has been observed to be inversely proportional to the number of term occurrences in a data corpus. The terms, those are common in a corpus, less likely to be used to discriminate relevant and irrelevant documents.

2.2. Ranking function

Many researchers have developed different ranking functions using VSM as IR model in the past. The major contributions in developing such type of ranking functions are categorized and discussed under following subsections.

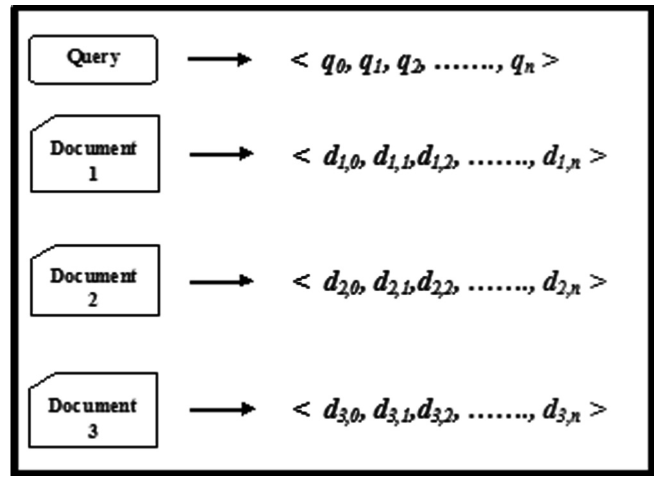


Fig. 1. Vector Space Model.

2.2.1. Statistical ranking functions

There are different conventional ranking functions in literature such as *Cosine*, *Jaccard* (Salton, 1998) and *Okapi* (Robertson, Walker, & Beaulieu, 1999), etc. *Cosine* ranking function computes cosine of the angle between the query and document vector. The assumption used in the *Cosine* is that the document length has no impact on relevance but later on Singhal, Salton, Mitra, and Buckley (1996) found that more documents judged to be relevant actually were found in longer documents. *Jaccard* is defined as the intersection of document and query vectors divided by the union of document and query vectors. Subsequently *Okapi* is developed as ranking function to overcome shortcomings of *Cosine* and *Jaccard*. This ranking function not only considers the term frequency, but also the length of the document and average length of the whole collection. *Okapi-BM25* (Christopher, Raghavan, & Schutze, 2009) is another latest variant of *Okapi* which enhances the performance of IR system. The mathematical representation of *Okapi-BM25* is given by (1)–(3).

$$Okapi-BM25(Q, D_i) = \sum_{T \in Q} W \frac{(k_1+1)tf}{K+tf} \times \frac{(k_3+1)qtf}{k_3+qtf} \tag{1}$$

where,

$$K = (k_1(1-b) + b.dl/avdl) \tag{2}$$

$$W = log(N - n + 0.5)/(n + 0.5) \tag{3}$$

Q is a query that contains the words T. D_i is a document in data set D. k_1, b and k_3 are constant parameters. *tf* is the term frequency of the term with a document, *qtf* is the term frequency in the query. N is the number of documents and n is the number of documents containing the term. *dl* and *avdl* are the document length and average document length respectively.

Unfortunately, the ranking functions mentioned above are not able to capture all the features of queries and documents due to the reasons already explained in previous section.

2.2.2. Evolutionary algorithm based and/or hybrid ranking functions

Some researchers used evolutionary algorithms such as Genetic Algorithm (GA) and Genetic Programming (GP) to construct ranking function for enhancement of IR system. Pathak, Gordon, and Fan (2000) propose a new weighted matching function to overcome the limitations of statistical ranking functions, which is a linear combination of different statistical ranking functions (*Cosine*, *Jaccard* and *Okapi*). The weighting parameters are determined by GA. This process overcomes the limitations of statistical ranking functions but suffers two drawbacks. First, it is a time consuming process. Secondly, it is not able to capture vagueness and uncertainty of documents and queries.

Afterword the same authors (Fan, Gordon, & Pathak, 2004) tried to develop ranking function using GP based learning fitness function. These authors propose a new GP based framework for ranking function that helps to automate the design of a ranking function. Tuomo, Jorma, and Martti (2007) used a connection between the *Cosine measure* and the *Euclidean* distance in association with Principal Component Analysis (PCA) and grounded searching on the latter. After retrieving relevant documents, PCA is run to cluster these documents, which increases the performance of IR. Yeh, Lin, Ke, and Yang (2007) propose an automatically generated ranking function for IR called *RankGP*. It employs GP to address the task of learning to rank by combining various types of evidences in IR, such as structure features, content features and query-independent features. Radwan, Latef, Ali, and Sadek (2008) present a new GA based ranking function for IR. They compare their approach with Cosine ranking function and find satisfactory results. Wang, Ma, and He (2010) propose the first immune programming based ranking function discovery approach. They use immune programming to the learning to the rank problem. The performance of this approach depends on its control parameters and there is no theoretical method to verify the values of these control parameters. Jiyin, Edgar, and Maarten (2011) present a result diversification framework based on query-specific clustering and cluster ranking, in which diversification is restricted to documents belonging to clusters that potentially contain a high percentage of relevant documents. Usharani and Iyakutti (2013) present a GA based ranking function for finding similarity of web documents based on *Cosine ranking function*. Bade, Bhat, and Borate (2014) present the new approach towards matching function based on GA for improving the performance of IR.

Although above researchers have made attempts to enhance the performance of IR, but certainly they have missed to focus upon one important aspect of addressing vagueness and uncertainty of queries and documents as well.

2.2.3. Fuzzy logic based ranking function

Fuzzy logic can be used to model uncertainties and vagueness of documents, queries and their characteristics (Zadeh, 1997). Therefore, Rubens (2006) proposes a fuzzy logic based approach to define a new ranking function. Three input variables namely *term frequency* (*tf*), *inverse document frequency* (*idf*), *overlap* and one output variable namely *relevance* are used in this ranking function. The ranges of input and output variables are represented by two linguistic terms as high and low. Triangular membership function is used to map inputs to a fuzzy set and fuzzy rules are derived from *tf.idf* weighting scheme e.g. if a query term has high *tf* and high *idf* in a document, then *relevance* is likely to be high. Besides this, another criterion is also used such as if many terms of the query are found in the document (*overlap*), then *relevance* is likely to be high.

Chen (2011) presented a new ranking function based on the geometric mean averaging operator to handle the similarity problems of generalized fuzzy numbers. Although, this research work is limited to fuzzy Information Retrieval and fuzzy datasets (Chen & Chen, 2003; Chen, Horng, & Lee, 2001; Chiang, Chow, & Hsien, 1997; Devedzic & Pap, 1999; Frigui, 2001).

It is evident that, very few works are reported in literature to develop fuzzy logic based ranking function during last one decade only despite of advantages already mentioned above. Therefore, there is an enough scope to develop fuzzy logic based ranking functions to enhance the performance of IR.

3. The proposed ranking function

A new fuzzy logic based ranking function is presented in this paper. The intuition behind using fuzzy logic is that it provides a convenient way to transform knowledge expressed in a natural language into fuzzy logic rules. The fuzzy logic based ranking function could be easily viewed, extended and verified. Fuzzy logic also allows combining the logic based model with the VSM. Therefore, the resulting model possesses simplicity and formalism of the logic based model, and the flexibility and performance of the VSM.

The proposed ranking function is based on term weighting scheme having different IR evidences which capture the features of documents and queries. Mamdani type Fuzzy Inference System (Mamdani & Assilian, 1975) is used in proposed ranking function. Fig. 2 represents a block diagram of composite *FIS* used for proposed ranking function. The notations of different entities, input and output variables used in Fig. 2 are described in Table 1.

In this paper, fuzzy logic is used to compute similarity score using fuzzy rules of composite *FIS*. As shown in Fig. 2, composite *FIS* has two levels such as *first level FIS* and *second level FIS*. *First level FIS* is based upon two *FLC*, FLC_{doc} and FLC_{que}. *Second level FIS*

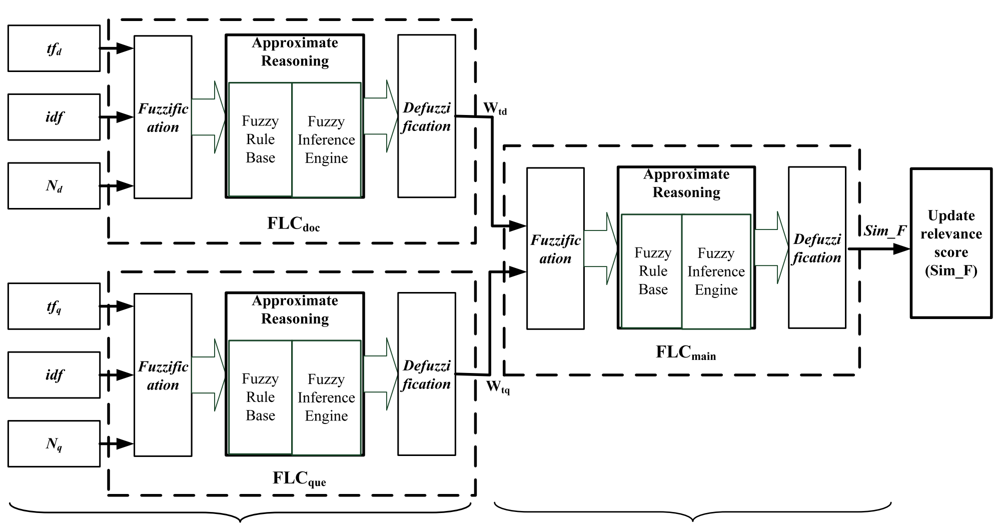


Fig. 2. Block diagram for Composite Fuzzy Inference System.

Table 1
Description of different entities, input and output variables used in composite *FIS*.

	Notation	Description
Variable	tf_d	Term frequency of a term in document
	tf_q	Term frequency of a term in query
	idf	Inverse document frequency that can be mathematically expressed as $idf = \log(N/n_t)$ where, N is the total number of documents in data corpus and n_t is the number of documents those are containing the term t
	N_d	Inverse of document length
	N_q	Inverse of query length
	w_{td}	Intermediate output and term weighting unit for document
	w_{tq}	Intermediate output and term weighting unit for query
	Sim_F	Final output i.e. relevance score
Entity	FLC_{doc}	Fuzzy Logic Controller for documents
	FLC_{que}	Fuzzy Logic Controller for queries
	FLC_{main}	Main Fuzzy Logic Controller, which gives final output

contains only one *FLC* i.e. FLC_{main}. In *first level FIS*, FLC_{doc} accepts values of input variables as tf_d, *idf* and N_d whereas FLC_{que} accept values of input variables as tf_q, *idf* and N_q. The outputs of FLC_{doc} and FLC_{que} are obtained in the form of values of w_{td} and w_{tq}, respectively. These values of w_{td} and w_{tq} are treated as input variables for *second level FIS*. FLC_{main} in *second level FIS* computes *sim_F* (relevance score) as final output. Each one the three *FLCs* used in composite *FIS*, has three processes – *fuzzification, approximate reasoning* and *defuzzification* as described in following subsections.

3.1. Fuzzification

This process converts crisp input values into degree of membership using membership function. The proposed ranking function uses linguistic terms to represent all input and output variables. These linguistic terms are represented by membership functions, obtained from domain knowledge, as shown in Fig. 3. A membership function is a curve that defines how each point in the input space is mapped to a degree of membership of fuzzy set (Zadeh, 1965).

The triangular type membership function is used to model the membership degrees of all the variables of composite *FIS*. The range of input variables tf_d, tf_q, *idf*, N_d and N_q for *first level FIS* are represented as *very high* (*VH*), *high* (*H*), *medium* (*M*), *low* (*L*) and *very low* (*VL*). The range of w_{td} and w_{tq} are represented as *high* (*H*), *medium* (*M*) and *low* (*L*) membership functions. The range of final output variable *Sim_F* is also represented by *high* (*H*), *medium* (*M*) and *low* (*L*) membership functions.

3.2. Approximate reasoning

An approximate reasoning is established for inference in order to deal with uncertainty and vagueness. It includes two

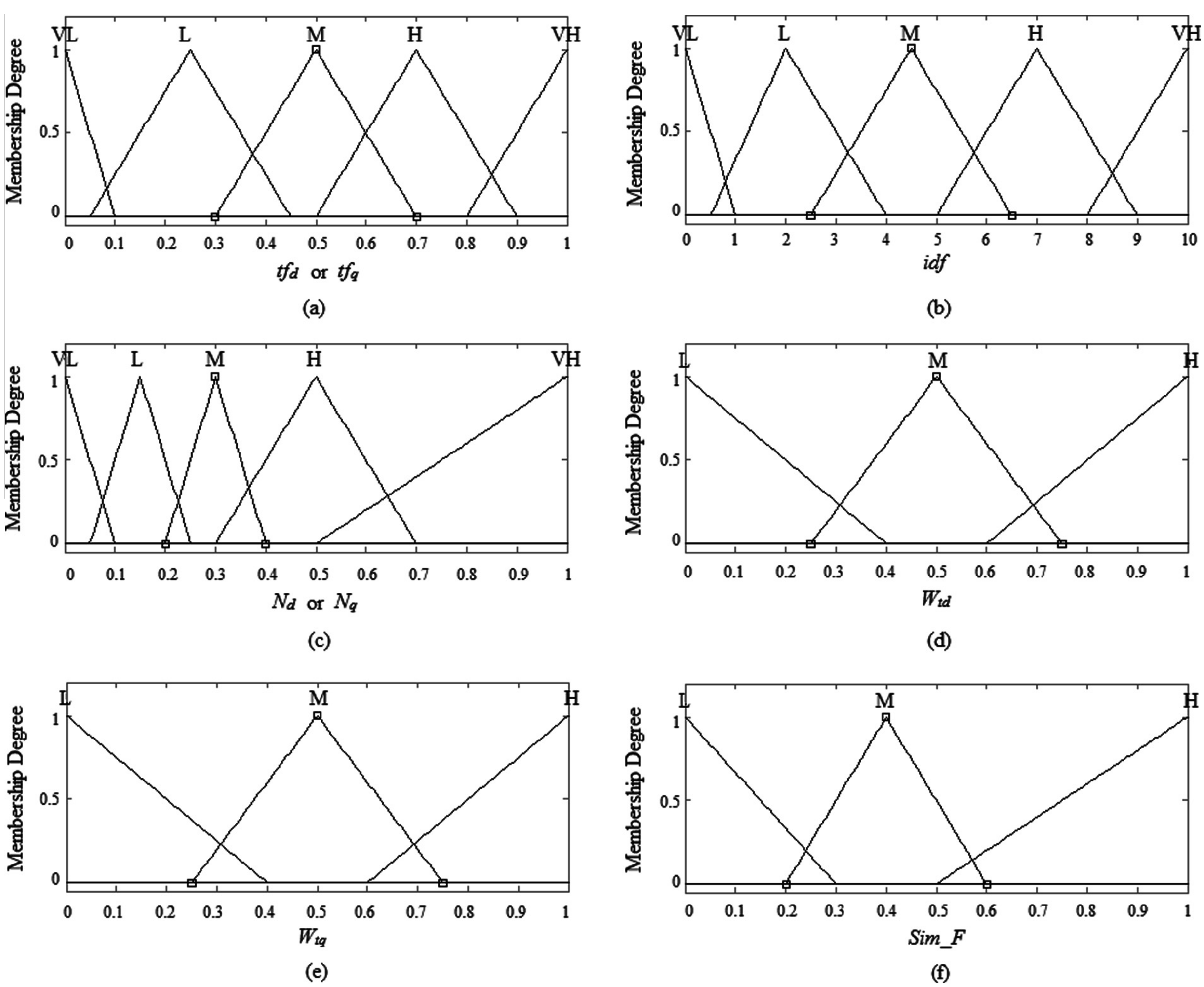


Fig. 3. Membership functions for the values of (a) tf_d or tf_q, (b) *idf*, (c) N_d or N_q, (d) W_{td}, (e) W_{tq} and (f) *Sim_F*.

sub-processes i.e. fuzzy rule base and fuzzy inference engine as described below:

3.2.1. Fuzzy rule base

The most common way to represent human knowledge is to form it into natural language expressions of the type

$$\text{IF premise (antecedent), THEN conclusion (consequent)} \tag{4}$$

$$\text{IF premise1 (antecedent) AND premise2 (antecedent), THEN conclusion (consequent)} \tag{5}$$

The form expressed in Eq. (4) is commonly referred to as the IF–THEN *fuzzy rule-based* form. It typically expresses an inference such that if we know a fact (premise, hypothesis, antecedent), then we can infer, or derive, another fact called a conclusion (consequent) (Ross, 1997). Another form expressed in equation (5) is referred as IF–THEN *fuzzy rule-based* form *with multiple conjunctive antecedents*. These forms of knowledge representation, characterized as *shallow knowledge*, are quite appropriate in the context of linguistics because it expresses human empirical and heuristic knowledge in our own language of communication.

In present work, fuzzy rules are derived from the common knowledge of IR system and term weighting scheme. Total 250 fuzzy rules are framed for *first level FIS* and 9 fuzzy rules are framed for *second level FIS*. The fuzzy rules include linguistic terms as well as linguistic hedges such as *very high, high, medium, low* and *very low* for *first level FIS.* Linguistic terms like *high, medium* and *low* are considered for *second level FIS*. Rubens found that the performance of IR system improves if the negated (using *not high*) or low featured fuzzy rules (using *low* and *very low*) are added in fuzzy rule base (Rubens, 2006). Therefore, low featured fuzzy rules are also framed in present work. These low featured fuzzy rules are also part of 250 rules at *first level FIS* and 9 rules at *second level FIS*. The order of the fuzzy rules does not affect the output. The domain knowledge used to frame fuzzy rules for proposed ranking function is tabulated in Table 2. The examples of some of the fuzzy rules based on this domain knowledge are also listed in the same table.

3.2.2. Fuzzy inference engine

The functionality of fuzzy inference engine for proposed ranking function can be understood in terms of following steps as listed in Table 3.

Another way to interpret the entire fuzzy inference process at once is by means of fuzzy inference rule view diagram. Fig. 4 shows a fuzzy inference rule view diagram in MATLAB™ Fuzzy Logic Toolbox (The mathworks Inc, 2004) for FLC_{main}, which contains three small plots in each row (antecedent and consequent of each fuzzy rule). Hence, each rule is represented as a row of plots and each column is represented as a variable. The first two columns of plots show the membership functions with reference to the antecedent or the *if*-part of the fuzzy rule. The third column of plot shows the membership function with reference to consequent or *then*-part of the fuzzy rule. The bottom most plot in third column represents the aggregate decision value of *sim_F*. Fuzzy inference rule view diagram may be used as a diagnostic for the performance of all nine fuzzy rules. The activation of rules and influence of individual membership function shapes on the results can be analyzed from Fig. 4. The names and the current values of variables are displayed on top of the each column. A yellow colored patch under the actual membership function curve is used to make the fuzzy membership value visually apparent. As the values of w_{td} and w_{tq} increase, the value of *Sim_F* also increases (i.e. more the document

Table 2
Domain knowledge of IR system used to frame the fuzzy rule base with some examples.

FIS	Fuzzy Logic Controller	Domain Knowledge	Examples of fuzzy rules
First level FIS	FLC_{doc}	If a term has high tf_d, *idf* values in a document and inverse of the length of that document (N_d) is also high (i.e. document length is small), then term weighting unit w_{td} is likely to be high	If (tf_d is *high*) and (*idf* is *high*) and (N_d is *high*) then (w_{td} is *high*)
		If a term has low tf_d, *idf* values in a document and inverse of the length of that document (N_d) is also low, then term weighting unit w_{td} is likely to be low	If (tf_d is *low*) and (*idf* is *low*) and (N_d is *low*) then (w_{td} is *low*)
	FLC_{que}	If a term has high tf_q in processed query, high *idf* in a document and inverse of the length of that query (N_q) is also high, then term weighting unit w_{tq} is also likely to be high	If (tf_q is *high*) and (*idf* is *high*) and (N_q is *high*) then (w_{tq} is *high*)
		If a term has low tf_q in processed query, low *idf* in a document and inverse of the length of that query (N_q) is also low, then term weighting unit w_{tq} is likely to be low	If (tf_q is *low*) and (*idf* is *low*) and (N_q is *low*) then (w_{tq} is *low*)
Second level FIS	FLC_{main}	If term weighting units, w_{td} and w_{tq} are high, then the document is likely to be more relevant (i.e. high *sim_F* score)	If (w_{td} is *high*) and (w_{tq} is *high*) then (*sim_F* is *high*)
		If term weighting units, w_{td} and w_{tq} are low, then the document is likely to be less relevant (i.e. low *sim_F* score)	If (w_{td} is *low*) and (w_{tq} is *low*) then (*sim_F* is *low*)

Table 3
Description of functional steps for fuzzy inference engine.

S.No.	Steps	Description
1	Fuzzy operator application	The membership degree to which each part of antecedent has been satisfied for each rule is known after fuzzifying the inputs. If the antecedent for a fuzzy rule has more than one part then fuzzy operator is applied to obtain a number which represents the result of the antecedent for that particular rule (The mathworks Inc, 2004). The *AND* fuzzy operator could be seen as an aggregation applied locally to the rule in this case
2	Implication method application	After applying fuzzy operator application, antecedent gives a number as output. This number is used as input in implication process and gives a fuzzy set as an output. Implication is implemented for each rule. There are two built-in methods in MATLAB™ Fuzzy Logic Toolbox (The mathworks Inc, 2004): *min* (minimum), which truncates the output fuzzy set, and *prod* (product), which scales the output fuzzy set In present work, *prod* method is used as Implication method
3	Output aggregation	The entire fuzzy rules must be combined in order to take the decision. Therefore aggregation process combines all the fuzzy sets into a single fuzzy set. There are three built-in aggregation methods available in MATLAB™ Fuzzy Logic Toolbox (The mathworks Inc, 2004): *max, sum* and *probabilistic OR* Since, as per the domain knowledge, all the fuzzy rules must be included in determination of ranking of documents. Therefore, *sum* aggregation method is used in proposed ranking function

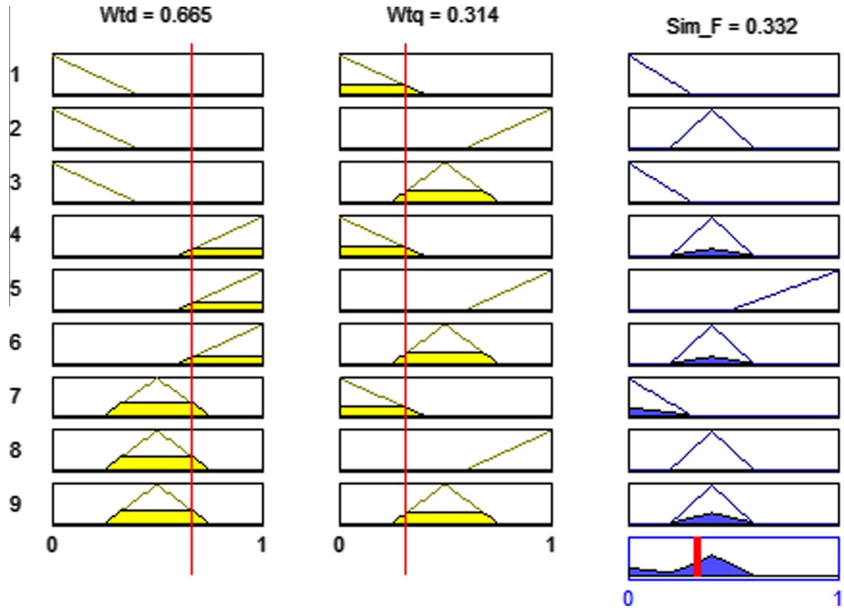


Fig. 4. Fuzzy inference rule view diagram for *second level FIS*.

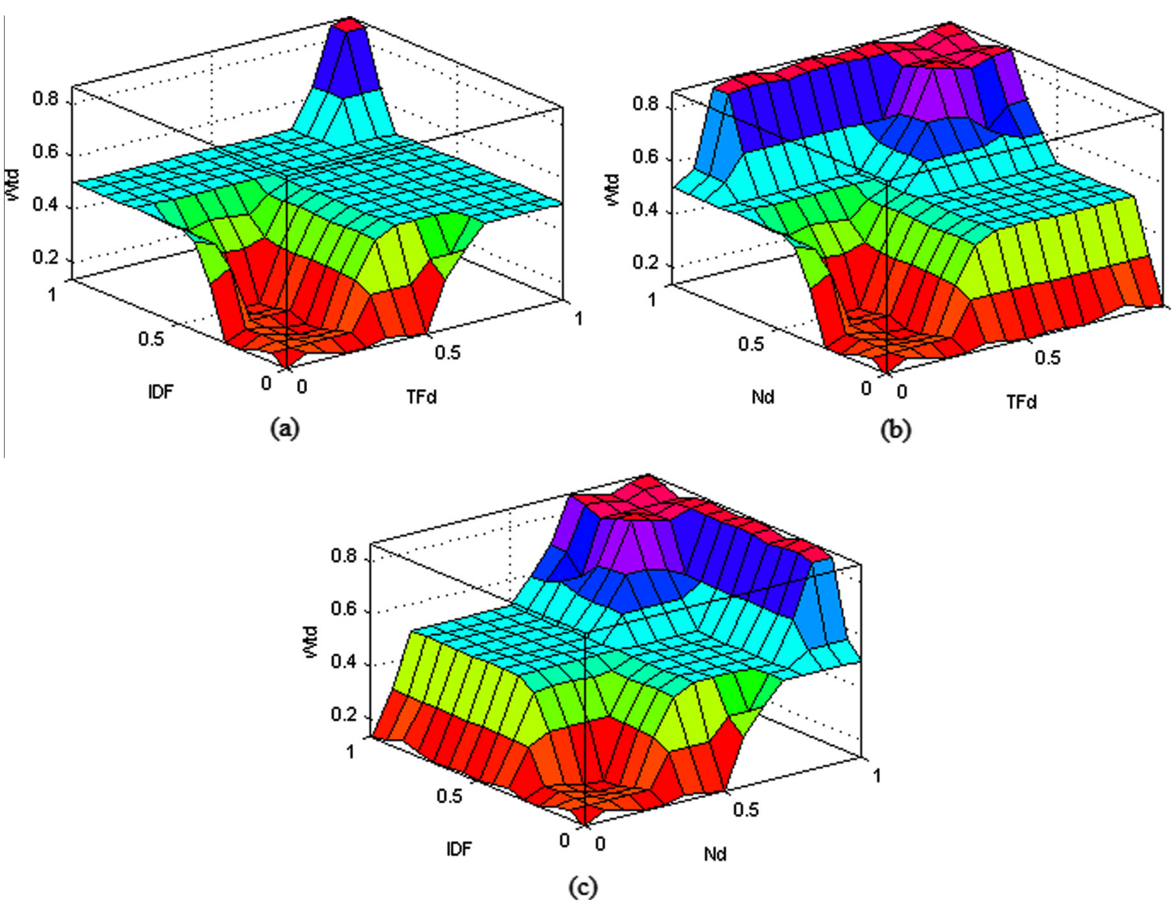


Fig. 5. Fuzzy inference surface view diagram for w_{td}.

relevant). Similar fuzzy inference rule view diagram can be obtained for *first level FIS* comprising of 250 fuzzy rules.

The rule view diagram shows one calculation at a time in detail. In this sense, it represents a sort of microscopic view of the whole fuzzy inference process. The three dimensional rule surfaces are plotted in Figs. 5 and 6 to understand the entire output surface of fuzzy inference process. Fig. 5(a)–(c) display the dependency of output (i.e. w_{td}) on any two of the three inputs (i.e. tf_d, idf and N_d). As the values of tf_d, idf and N_d increase then w_{td} also increases i.e. more relevant document will be retrieved. The similar analysis

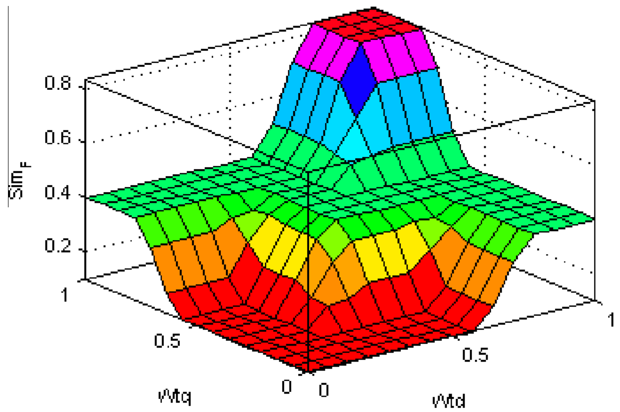


Fig. 6. Fuzzy inference surface view diagram for *Sim_F*.

can be done for w_{tq} with respect to tf_q, idf and N_q to understand the dependencies of variables. Fig. 6 demonstrates that *sim_F* increases very fast for the higher values of w_{td} and w_{tq}.

3.3. Output defuzzification

This process gives a single crisp value as an output after defuzzifying the aggregate fuzzy set. There are different methods available for defuzzification: *middle of maximum* (the average of the maximum value of the output set), *largest of maximum*, *smallest of maximum*, *centroid* and *bisector*. In this proposed ranking function, the *centroid method* or *center of sums method* (Sugeno, 1985b; Lee, 1990) is used for defuzzification method as defined by (6).

$$Y = \int_y \sum_{i=1}^{n} y.\mu_{Bi}(y)dy \Big/ \int_y \sum_{i=1}^{n} \mu_{Bi}(y)dy \qquad (6)$$

where the input for the defuzzification process is a fuzzy set $\mu_{Bi}(y)$ (the aggregate output fuzzy set) and the output is a single number Y.

At the end of this section the main features of our proposed fuzzy logic based ranking function are highlighted and compared with Rubens' approach (Rubens, 2006) in Table 4 as given below.

4. Experimental results and analysis

The experiments are performed on two benchmark datasets, *CACM* and *CISI*. *CACM* dataset is based on computers and communications, while *CISI* dataset is based on IR system. Both datasets are in English language. *CACM* and *CISI* contain *3204* and *1460* documents respectively.

The performance of proposed ranking function is evaluated in terms of *precision*, *recall* and *F-measure* as defined by (7)–(9) respectively (Yates & Berthier, 1999).

$$Precision = \frac{|R_a|}{|A|} \qquad (7)$$

$$Recall = \frac{|R_a|}{|R|} \qquad (8)$$

$$\text{F-measure} = \frac{2 * Precision * Recall}{(Precision + Recall)} \qquad (9)$$

where $|R_a|$ is a set of relevant documents retrieved, $|A|$ is a set of total documents retrieved and $|R|$ is a set of all relevant documents.

Twenty five queries are randomly chosen from each of the dataset. After submitting these queries to IR system, a set of documents is retrieved and sorted according to their relevance scores with respect to the queries using ranking function. The values of *precision, recall* and *F-measure* are calculated for top retrieved documents at rank ten, twenty and thirty cut-offs. The results are compared with Rubens' ranking function (Rubens, 2006) along with *Okapi-BM25.*

Above mentioned three cut-offs are considered to check the consistency of the performance of our proposed ranking function. It is not necessary that ranking function gives better *precision* and *recall* values for all three cut-offs. For example, a ranking function may give better *precision* and *recall* values for top ten retrieved documents but after that it may not retrieve any relevant document for next top ten retrieved documents. Therefore, the performance of ranking function falls (degrades) for top twenty retrieved documents in such case. The same argument holds true for top thirty retrieved documents also. The comparison of performance of the proposed ranking function with *Okapi-BM25* and Rubens' ranking function is discussed in following subsections.

4.1. Overall retrieval effectiveness

Average *precision* and average *recall* values are important indicators to check the performance of IR system at once. Therefore, the comparison of the average *precision* and the average *recall* values of the proposed ranking function with existing Rubens' ranking function (Rubens, 2006) along with widely used ranking function *Okapi-BM25* are presented in Tables 5 and 6 with respect to twenty five queries for both the datasets. It is evident from Tables 5 and 6 that the proposed ranking function gets higher average *precision* and higher average *recall* values than Rubens' ranking function (Rubens, 2006) and *Okapi-BM25* with respect to top retrieved documents at rank ten, twenty and thirty cut-offs respectively. These tables also highlight the percentage improvements of the proposed ranking function and Rubens' ranking function in comparison to *Okapi-BM25.*

4.2. Query-based retrieval effectiveness

The values of *precision, recall* and *F-measure* are calculated for top retrieved documents at rank ten, twenty and thirty as shown in Figs. 7–15. These figures clearly illustrate that higher *precision, recall* and *F-measure* values are obtained for the proposed ranking function in comparison to *Okapi-BM25* and Rubens' ranking function (Rubens, 2006).

Fig. 7(a) and (b) clearly show that higher values of *precision* are obtained using proposed ranking function for top ten retrieved documents for twenty four queries in comparison to other ranking functions, in case of both the datasets. Fig. 8(a) and (b) reveal that the proposed ranking function gives better *precision* values of top twenty retrieved documents for twenty three queries in case of *CACM* dataset and for twenty two queries in case of *CISI* dataset respectively. From Fig. 9(a) and (b), again it is observed that the *precision* values obtained from the proposed ranking function are better than *Okapi-BM25* and Rubens' ranking function (Rubens,

Table 4
Comparison of features of proposed ranking functions with Rubens' approach.

S.No.	Features	Proposed ranking function	Rubens' ranking function
1	Fuzzy controller	Three Fuzzy Logic Controllers FLC_{doc}, FLC_{que} and FLC_{main} are used	Only one Fuzzy Logic Controller is used
2	Input variables	tf_d, tf_q, idf, N_d, N_q, w_{td} and w_{tq}	*tf*, *idf* and *overlap* (terms of the query are found in document)
3	Membership functions	Very high, high, medium, low and very low	High and not high
4	Fuzzy rules	Not dependent on number of terms of query	Dependent on number of terms in query

Table 5
Comparison of the average *precision* and the average *recall* values of the proposed ranking function with *Okapi-BM25* and Rubens' approach for *CACM* dataset.

Method		Top ten retrieved documents		Top twenty retrieved documents		Top thirty retrieved documents	
		Average precision	Average recall	Average precision	Average recall	Average precision	Average recall
Okapi-BM25		0.2268	0.0988	0.2065	0.1604	0.1892	0.2089
Rubens' approach	Value	0.2592	0.1197	0.2181	0.2609	0.1924	0.2862
	Improvement with respect to *Okapi-BM25*	14.29%	21.15%	5.62%	62.65%	1.69%	37.00%
The proposed ranking function	Value	**0.3574**	**0.1602**	**0.2850**	**0.2972**	**0.2472**	**0.3214**
	Improvement with respect to *Okapi-BM25*	**57.58%**	**62.14%**	**38.01%**	**85.28%**	**30.65%**	**53.85%**

Bold values show the better results obtained by our proposed method in comparison to others.

Table 6
Comparison of the average *precision* and the average *recall* values of the proposed ranking function with *Okapi-BM25* and Rubens' approach for CISI dataset.

Method		Top ten retrieved documents		Top twenty retrieved documents		Top thirty retrieved documents	
		Average precision	Average recall	Average precision	Average recall	Average precision	Average recall
Okapi-BM25		0.1246	0.0278	0.1142	0.0594	0.1004	0.0828
Rubens' approach	Value	0.2747	0.0750	0.2173	0.1248	0.1868	0.1589
	Improvement with respect to *Okapi-BM25*	120.46%	169.78%	90.28%	110.10%	86.05%	91.90%
The proposed ranking function	Value	**0.3053**	**0.0801**	**0.2397**	**0.1365**	**0.2043**	**0.1739**
	Improvement with respect to *Okapi-BM25*	**145.02%**	**188.12%**	**109.89%**	**129.79%**	**103.48%**	**110.02%**

Bold values show the better results obtained by our proposed method in comparison to others.

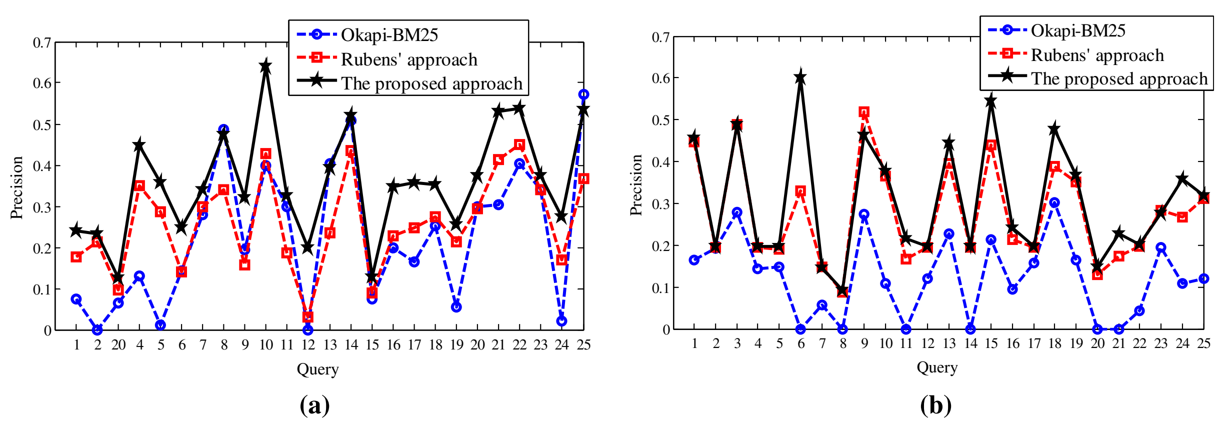


Fig. 7. The *precision* values of the top ten retrieved documents with respect to twenty five queries for different methods for (a) *CACM* dataset, (b) *CISI* dataset.

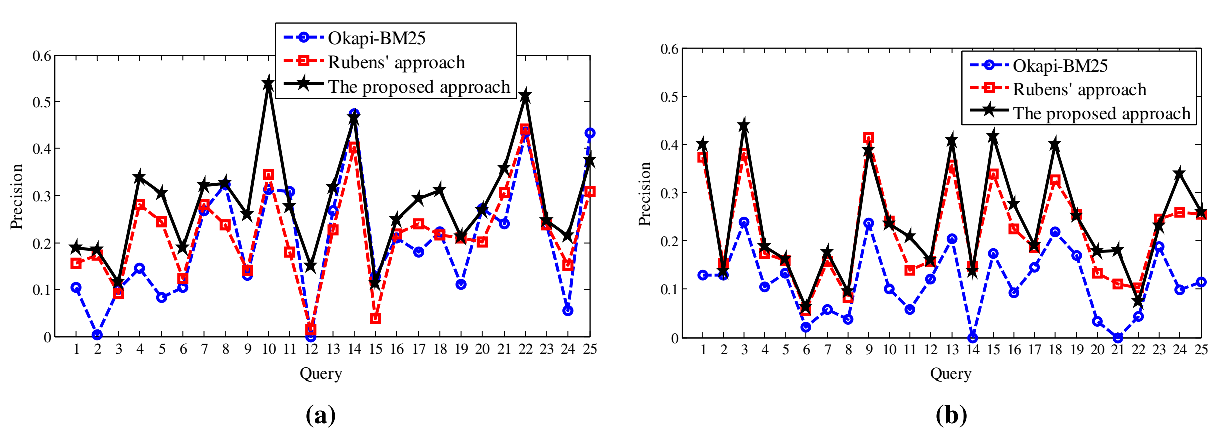


Fig. 8. The *precision* values of the top twenty retrieved documents with respect to twenty five queries for different methods for (a) *CACM* dataset, (b) *CISI* dataset.

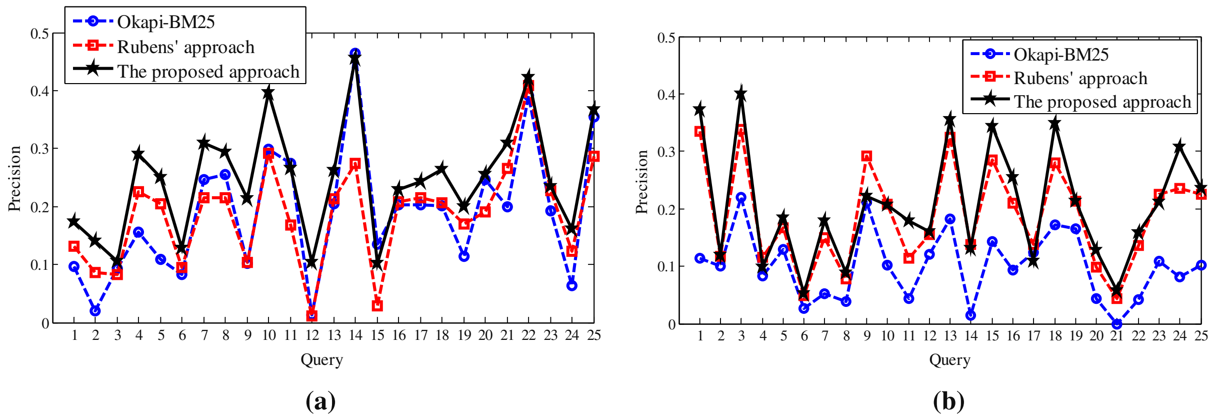


Fig. 9. The *precision* values of the top thirty retrieved documents with respect to twenty five queries for different methods for (a) *CACM* dataset, (b) *CISI* dataset.

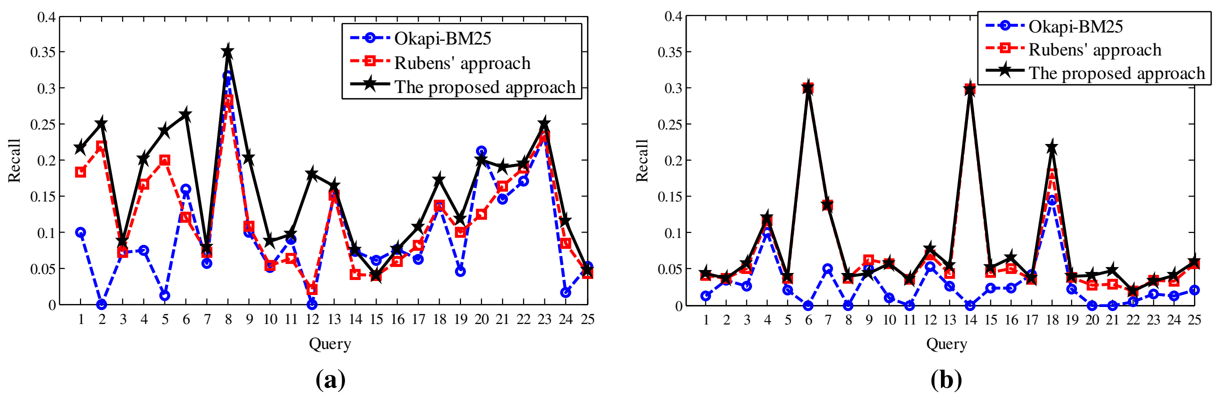


Fig. 10. The *recall* values of the top ten retrieved documents with respect to twenty five queries for different methods for (a) *CACM* dataset, (b) *CISI* dataset.

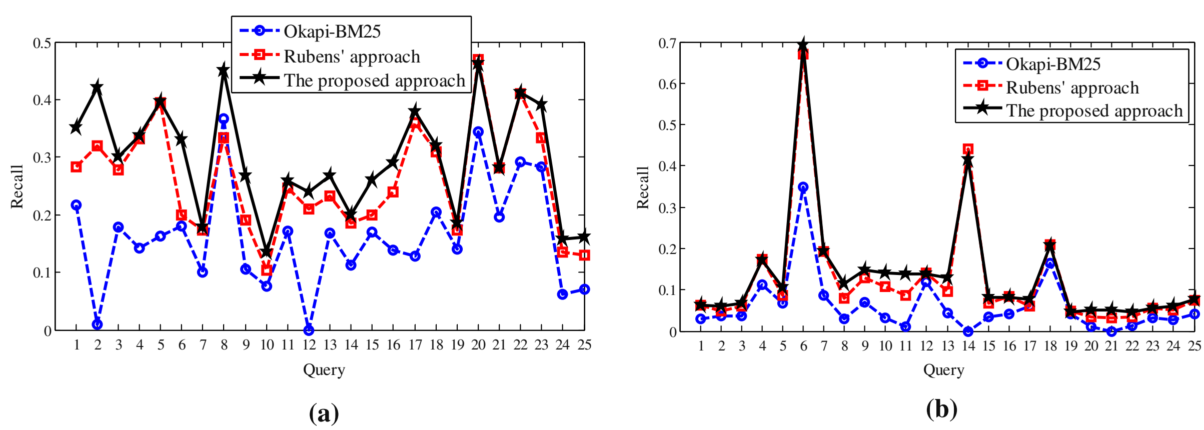
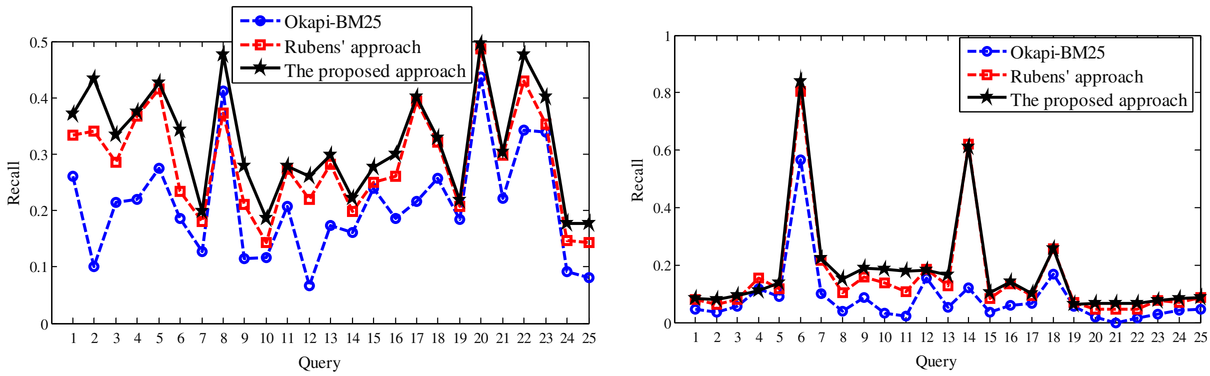


Fig. 11. The *recall* values of the top twenty retrieved documents with respect to twenty five queries for different methods for (a) *CACM* dataset, (b) *CISI* dataset.

2006) for twenty four queries in case of *CACM* dataset and for twenty three queries in case of *CISI* dataset respectively.

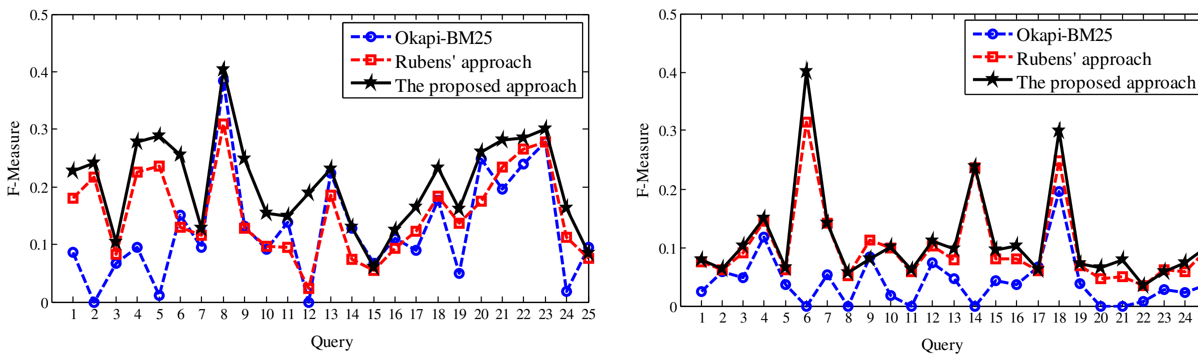
Fig. 10(a) and (b) clearly show that higher values of *recall* using proposed ranking function for top ten retrieved documents are obtained for twenty four queries in comparison to other ranking functions, in case of *CACM* dataset and for all twenty five queries in case of *CISI* datasets respectively. Fig. 11(a) and (b) reveal that the proposed ranking function gives better *recall* values of top twenty retrieved documents for all twenty five queries in case of *CACM* dataset and for twenty four queries in case of *CISI* dataset respectively. From Fig. 12(a) and (b), again it is observed that the *recall* values obtained from the proposed ranking function are better than *Okapi-BM25* and Rubens' ranking function (Rubens, 2006) for all twenty five queries in case of *CACM* dataset and for twenty four queries in case of *CISI* dataset respectively.

The results are also compared in terms of *F-measure* as shown in Figs. 13–15. *F-measure* is calculated to get a single measure of effectiveness. From the figures, it can be seen that the proposed ranking function outperforms *Okapi-BM25* and Rubens' ranking function. Fig. 13(a) shows that the proposed ranking function obtains better values of *F-measure* for almost all of the queries for top ten retrieved documents in case of *CACM* dataset, however



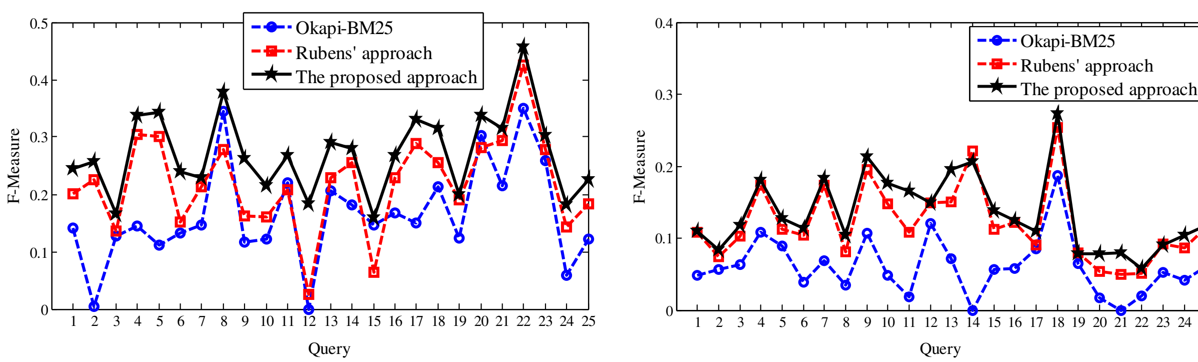
(a) (b)

Fig. 12. The *recall* values of the top thirty retrieved documents with respect to twenty five queries for different methods for (a) *CACM* dataset, (b) *CISI* dataset.



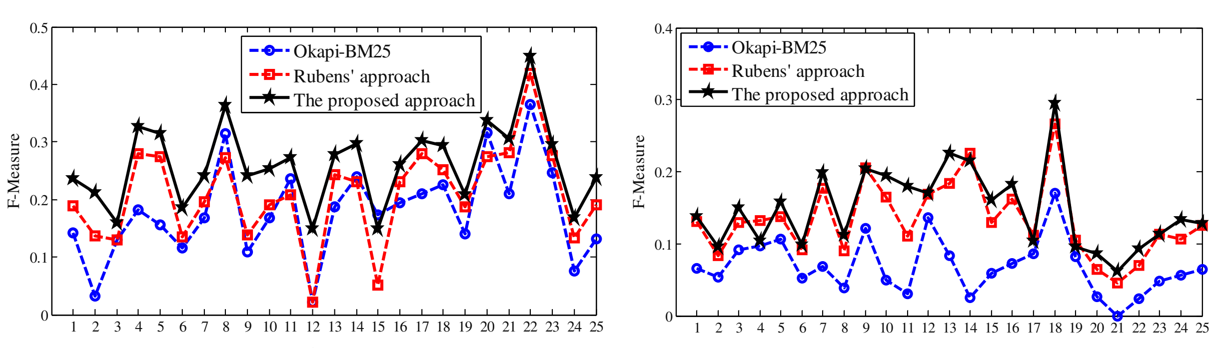
(a) (b)

Fig. 13. The *F-measure* of the top ten retrieved documents with respect to twenty five queries for different methods for (a) *CACM* dataset, (b) *CISI* dataset.



(a) (b)

Fig. 14. The *F-measure* of the top twenty retrieved documents with respect to twenty five queries for different methods for (a) *CACM* dataset, (b) *CISI* dataset.



(a) (b)

Fig. 15. The *F-measure* of the top thirty retrieved documents with respect to twenty five queries for different methods for (a) *CACM* dataset, (b) *CISI* dataset.

Table 7
Paired t-test results on *CACM* and *CISI*.

	Data set	*Okapi-BM25*			Rubens' ranking function		
		h-Value	*p*-Value	CI	*h*-Value	*p*-Value	CI
The proposed ranking function	*CACM*	1	0.0000	[−0.1740, −0.0871]	1	0.0000	[−0.1198, −0.0776]
	CISI	1	0.0000	[−0.2309, −0.1304]	1	0.0198	[−0.0558, −0.0053]

the values of couple of queries are equal to the proposed ranking function. Fig. 13(b) shows the results of *CISI* dataset for top ten retrieved documents. Our approach lags behind only for one query (ninth query) from Rubens' approach. From Fig. 14(a) and (b), again it can be observed that the values of *F-measure* obtained by the proposed ranking function are better than other ranking functions for all twenty five queries in case of *CACM* dataset and for twenty four queries in case of *CISI* dataset. These diagrams are drawn for top twenty retrieved documents. Fig. 15(a) and (b) also reveal that the proposed ranking function gives better values of *F-measure* for top thirty retrieved documents for all twenty five queries in case of *CACM* dataset and for twenty three queries in case of *CISI* dataset respectively.

4.3. Statistical analysis

The statistical *paired t-test* results obtained for *CACM* and *CISI* datasets are tabulated in Table 7. A *paired t-test* is the most commonly used hypothesis test in IR. In the present work, the *paired t-tests* are conducted to determine whether the proposed ranking function is statistically different from *Okapi-BM25* and Rubens' ranking functions or not. These *paired t-tests* return the results in terms of *h-value, p-value* and *CI* as shown in Table 7. The *p-value* = 0 indicates that the null hypothesis is rejected and that the mean of our data is significantly different from other approaches with 95% certainty and therefore the null hypothesis ("means are equal") cannot be rejected at the 5% significance level ($\alpha = 0.05$). If the *p-value* = 1 then the performances are not statistically different and therefore the null hypothesis ("means are equal") can be rejected at the 5% significance level ($\alpha = 0.05$). The *CI* is the 95% confidence interval of the mean based upon the *t-distribution*. Table 7 clearly indicates that the improvement of the proposed ranking function over *Okapi-BM25* is statistically significant at $\alpha = 0.05$ (p is almost *zero* for both the datasets). This table also presents that the proposed ranking function is more statistically significant then Rubens' approach at $\alpha = 0.05$ ($p = 0.00$ and $p = 0.01$ for *CACM* and *CISI*, respectively).

4.4. Analysis of success/failure in the retrieval effectiveness

The performance of any ranking function depends upon capability of capturing the features of queries and documents as well. Each query and each document have different features and different IR evidences are used to capture these features. The features of all the queries and documents cannot be captured completely and therefore, best results for all types of datasets cannot be assured by any ranking function.

Although our proposed ranking function performs much better than *Okapi-BM25* and Rubens' approach for most of the queries but it lags behind above mentioned two ranking functions for few queries. It is noticed from Figs. 12–14 that *Okapi-BM25* gives better precision only for query No. 25 in comparison to our proposed ranking function for top ten and top twenty retrieved documents in case of *CACM* dataset but for top thirty retrieved documents, our proposed ranking function is better. As already explained in Section 2.2, there are three constants in *Okapi-BM25* i.e. k_1, b and k_3. k_1 and b control term frequency scaling and document length normalization respectively. If $k_1 = 0$, the ranking function is binary based model and raw term frequency based model otherwise. Similarly, if $b = 0$, there is no length normalization and if $b = 1$, there is relative frequency (fully scale by document length). After a number of experiments, k_1 and b are set to 1.2 and 0.75, respectively, so as to realize *Okapi BM-25* ranking function with features of raw term frequency based model and 75% document length normalization. Therefore, the features of query No. 25 are captured by *Okapi-BM25* for above settings of constants k_1 and b, in more efficient way than Rubens' and our proposed ranking function.

The performance of proposed ranking function slightly lags behind Rubens' approach for 9th query as shown in Figs. 12–14, Rubens used three input variables *tf, idf* and *overlap* as already discussed in Section 2.2. In case of 9th query, *overlap* plays an important role. Its value is high for 9th query as compared to other queries, because most of the terms of this query are also in documents. The associated weights of *tf.idf* rules and *overlap* rules are also set differently by Rubens. The different IR evidences (tf_d, tf_q, idf, N_d and N_q) are used to compute similarity scores and equal associated weights are considered for all the rules in our proposed ranking function. Moreover, *overlap* is not included as an input in our proposed ranking function because some degree of overlap is covered by *tf.idf schema* (Rubens, 2006). Therefore, in particular case of query No. 9, Rubens' approach captures the features of query in a better manner as compared to proposed ranking function.

5. Conclusion and future directions

A new method is proposed in this paper to construct a ranking function based on fuzzy logic for IR. The proposed ranking function is based on composite *FIS* structure which improves the performance of IR system due to the extension of fuzzification of IR evidences at two levels. The main strength of composite *FIS* lies in *fuzzy rule bases* (first two at *first level FIS* and last one at *second level FIS*) which transform the domain knowledge into fuzzy sets using total *259* fuzzy rules. The different IR evidences (tf_d, tf_q, idf, N_d and N_q) are used for proposed ranking function in order to capture more features of queries and documents represented in the form of vectors using VSM.

CACM and *CISI* benchmark datasets are used to validate our proposed ranking function. It is clear from the experiments that our proposed ranking function increases the values of *precision*, *recall* and *F-measure*. The higher average *precision* and average *recall* values are also obtained by proposed ranking function in comparison to *Okapi-BM25* and Rubens' ranking functions. A paired *t-test* is conducted to perform statistical analysis. This statistical analysis confirms that the proposed ranking function significantly improves the retrieval of relevant documents as compare to *Okapi-BM25* and Rubens' ranking functions.

The present work is a significant effort to apply fuzzy logic in developing ranking function after Rubens' work and the results are appreciating. In future, the work can be extended in some of the directions as pointed out herewith. The robustness of proposed ranking function may be further tested on other large sized dataset such as *TREC*. Additional IR evidences may be included to improve the performance of the IR system. The different associated weights of fuzzy rules may be analyzed to capture the features of queries and documents more effectively. The different membership functions, aggregation operators and linguistic quantifiers may be considered in order to improve the performance of fuzzy logic based ranking function proposed in this paper.

Acknowledgments

The authors are very grateful to the anonymous reviewers for their valuable and constructive comments with helpful suggestions to improve the paper's quality.

References

Abraham, K., Lihong, L., & Zhiqiang, C. (1992). Fuzzy inference and its applicability to control systems. *Fuzzy Sets and Systems, 48*(1), 99–111.

Bade, Y., Bhat, R., & Borate, P. (2014). Optimization techniques for improving the performance of information retrieval system. *International Journal of Research in Advent Technology, 2*(2), 263–267.

Chen, S. J. (2011). Fuzzy information retrieval based on a new similarity measure of generalized fuzzy numbers. *Intelligent Automation and Soft Computing, 17*(4), 465–476.

Chen, S. J., & Chen, S. M. (2003). Fuzzy risk analysis based on similarity measures of generalized fuzzy numbers. *IEEE Transactions on Fuzzy Systems, 11*(1), 45–56.

Chen, S. M., Horng, Y. J., & Lee, C. H. (2001). Document retrieval using fuzzy-valued concept networks. *IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics, 31*(1), 111–118.

Chiang, D. A., Chow, L. R., & Hsien, N. C. (1997). Fuzzy information in extended fuzzy relational databases. *Fuzzy Sets and Systems, 92*(1), 1–20.

Christopher, D. M., Raghavan, P., & Schutze, H. (2009). *An introduction to information retrieval*. Cambridge University Press.

Cordon, O., Moya, F. & Zarco, C. (2004). Fuzzy logic and multi-objective evolutionary algorithms as soft computing tools for persistent query learning in text retrieval environments. In *Proceedings of the IEEE international conference on fuzzy systems* (pp. 571–576).

Cordon, O., Viedma, E., Pujalte, C., Luque, M., & Zarco, C. (2003). A review on the application of evolutionary computation of information retrieval. *International Journal of Approximate Reasoning, 34*, 241–263.

Devedzic, G. B., & Pap, E. (1999). Multicriteria-multistages linguistic evaluation and ranking of machine tools. *Fuzzy Sets and Systems, 102*(4), 451–461.

Fan, W., Gordon, M., & Pathak, P. (2004). A generic ranking function discovery framework by genetic programming for information retrieval. *Information Processing and Management, 40*, 587–602.

Frigui, H. (2001). Interactive image retrieval using fuzzy sets. *Pattern Recognition Letters, 22*(9), 1021–1031.

Haase, V. H., Steinmann, V., & Vejda, S. (2002). Access to knowledge: Better use of the internet. In *Proceedings of the informing science and it education conference* (pp. 618–627).

Harman, D. K. (1993). Overview of the first text retrieval conference (TREC-I). In *Proceedings of the first text retrieval conference, (TREC'93)* (pp. 1–20). NIST Special Publication.

Jang, J. S. R., & Sun, C. T. (1997). *Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence*. Prentice Hall.

Jiyin, H., Edgar, M., & Maarten, R. (2011). Result diversification based on query specific cluster ranking. *Journal of the American Society for Information Science and Technology, 62*(3), 550–571.

Jones, S. K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation, 28*(1), 11–20.

Jones, W. P., & Furnas, G. W. (1987). Pictures of relevance: A geometric analysis of similarity measures. *Journal of American Society of Information Sciences, 38*, 420–442.

Lancaster, F. W., & Warner, A. I. (1993). *Information retrieval today*. USA: Information Resources Press.

Lee, C. (1990). Fuzzy logic in control systems: Fuzzy logic controller, Parts I and II. *IEEE Transaction on System, Man and Cybernetics, 20*, 404–435.

Mamdani, E. H., & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man–Machine Studies, 7*, 1–13.

Mercier, A., & Beigbeder, M. (2005). Fuzzy proximity ranking with boolean queries. In *Proceedings of the 14th text retrieval conference (TREC 2005)* (pp. 433–442).

Pathak, P., Gordon, M., & Fan, W. (2000). Effective information retrieval using genetic algorithms based matching functions adaption. In *Proceedings of 33rd hawaii international conference on science (HICS)*. Hawaii, USA.

Radwan, A. A. A., Latef, B. A. A., Ali, A. M. A., & Sadek, O. A. (2008). Using genetic algorithm to improve information retrieval systems. *World Academy of Science, Engineering and Technology, 2*, 748–754.

Robertson, S. E. (1997). The probabilistic character of relevance. *Information Processing & Management, 13*, 247–251.

Robertson, S. E., Walker, S., & Beaulieu, M. (1999). Okapi-BM25 at TREC–7: Automatic ad hoc, filtering, VLC and filtering tracks. In *Proceedings of the seventh text retrieval conference (TREC-7)* (pp. 253–264).

Ross, T. J. (1997). *Fuzzy logic with engineering applications*. Singapore: McGraw-Hill.

Rubens, N. O. (2006). The application of fuzzy logic to the construction of the ranking function of information retrieval system. *Computer Modeling and New Technologies, 10*, 20–27.

Salton, G. (1968). *Automatic information organization*. New York: McGraw-Hill.

Salton, G. (1998). *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management, 24*(5), 513–523.

Salton, G., & McGill, M. (1983). *Introduction to modern information retrieval*. New York: McGraw-Hill.

Singhal, A., Salton, G., Mitra, M., & Buckley, C. (1996). Document length normalization. *Information Processing and Management, 32*(5), 619–633.

Subtil, P., Mouaddib, N., & Faucout, O. (1996). A fuzzy information retrieval and management system and its applications. In *Proceedings of the ACM symposium on applied computing*. USA.

Sugeno, M. (1985a). *Industrial applications of fuzzy control*. Elsevier Science Publication Co..

Sugeno, M. (1985b). An introductory survey of fuzzy control. *Information Science, 36*, 59–83.

The mathworks Inc. (2004). Fuzzy logic toolbox user's guide.

Tuomo, K., Jorma, L., & Martti, J. (2007). On principal component analysis, cosine and euclidean measures in information retrieval. *Information Science, 177*(22), 4893–4905.

Usharani, J., & Iyakutti, K. (2013). A genetic algorithm based on cosine similarity for relevant document retrieval. *International Journal of Engineering Research and Technology, 2*(2), 1–5.

Wang, S., Ma, J., & He, Q. (2010). An immune programming based ranking function discovery approach for effective information retrieval. *Expert Systems with Applications, 37*, 5863–5871.

Witten, I., Moffat, A., & Bell, T. (1999). *Managing gigabytes: Compressing and indexing documents and images*. Morgan Kaufmann.

Yap, K. H., & Wu, K. (2005). A soft relevance framework in content-based image retrieval systems. *IEEE Transactions on Circuits and Systems for Video Technology, 15*(12), 1557–1568.

Yates, R. B., & Berthier, R. (1999). *Modern information retrieval*. Addisson Wesley.

Yeh, J. Y., Lin, J. Y., Ke, H. R., & Yang, W. P. (2007). Learning to rank for information retrieval using genetic programming. SIGIR'07, Amsterdam, Netherlands.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control, 8*, 338–353.

Zadeh, L. A. (1997). Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets and Systems, 90*(2), 111–127.