IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 3, Ver. IV (May-Jun. 2014), PP 65-74
www.iosrjournals.org

A New Web Document Retrieval Method Using Extended-IOWA (Extended-Induced Ordered Weighted Averaging) Operator on HTML Tags

Sukrati Pathak[1,*]and Sakshi Mitra[2]
[1]*Computer Science & IT Dept.*, *Banasthali Vidyapith, Banasthali, Rajasthan, India*
[2]*Computer Science & Engg. Dept.*, *IGDTUW, Delhi, India*

Abstract: *A new scenario has arisen into the information retrieval (IR) field with the increase in the use of mark-up languages. This paper targets structured IR and is focused on documents with structure. This assumption forces us to estimate the different weights which are applied to every field of structured web documents (designed using HTML). In this work a new ranking function based on fuzzy logic called Extended-IOWA operator for structured IR has proposed. Its purpose is to develop a competent IR system through Extended-IOWA operator with weighted HTML tags. We prioritized HTML tags into four classes and assign fuzzy weights to these classes according to their significance in text retrieval. Document weights are based on tags, which contain query terms. Consequently each class generates a matrix which describes document-document relationship using Linguistic terms which we represent using Trapezoidal Fuzzy Numbers. Document score is calculated in different classes and finally scores of documents are aggregated by Extended-IOWA which in turn returns result in the form of final ranked list of relevant documents.*

Keywords: *Document Retrieval, Fuzzy Logic, OWA Operators, HTML Tags, Multi-criteria decision making Problem, Trapezoidal Fuzzy numbers.*

I. Introduction

Presently most of the information retrieval systems still employ the Boolean logic model; however these information retrieval systems are not able to deal uncertain information, so in the present paper we proposed a model which uses linguistic terms to deal with uncertainty. These linguistic terms are represented by trapezoidal fuzzy numbers [1]. In comparison to Boolean model, fuzzy logic deals with vagueness/uncertainty of information (unlike the Boolean model which is based on binary decision criterion {relevant, not relevant}), and expresses relevance as degrees of memberships.

There is a huge involvement of HTML (Hyper Text Markup Language) in development of web pages available on the Internet. HTML contains a set of markup tags that represent the content, the presentation layout of the web documents. These HTML tags have different priorities in document structure, so traditional techniques that assign the term weights on the basis of frequency of occurrence only may not be provided adequate results. Deng have shown the fuzzy representation (FR) of WWW information based on Significance of HTML tags which is an effective alternative for characterizing Web documents [5]. Practically, Vector Space Model (VSM), Term Frequency (TF) and Inverse Term Frequency (IDF) are among other long-established models [3] [5] [15] [16] employed in conventional IR systems.

In presented paper we have focused on some of the issues which arrive while retrieving the HTML documents. Firstly, in HTML documents it is possible that a query term is present in only title tag, however title tag is consider as a highest prioritize tags. So document that contain query term in title tag but not in any other tag (in other part of document) is kept first by information retrieval system, which a not an efficient way of retrieval. Secondly, only term frequency (tf) & inverse-document frequency (idf) is not sufficient to make a powerful & highly scalable IR system. It is also possible that query terms are present in more than one tag (with different priorities) in same document and in more than one document, then it becomes difficult to merge all document weights and represent in a final ranked list proficiently.

Treating each tag independently is not an optimized way therefore we divide HTML tags (most of all tags, which are responsible for text retrieval) into finite number of classes. In this paper a new structure-sensitive web information retrieval model for HTML documents has been proposed. We firstly prioritize HTML tags and make four classes (i.e. C_1, C_2, C_3, & C_4) of HTML tags on basis of similar functionalities. These classes stand for title, header, emphasized and delimiters tags of HTML respectively. Afterwards, query terms are parsed on the basis of HTML tags and in this fashion each class generates a set of documents according to included tags that contain query terms. Since A term can be present in more than one document and in different tags and hence we aggregate all the weights of retrieved documents. For aggregation of these weights we propose Extended-Induced Ordered Weighted Operator (Extended-IOWA). Finally a ranked list of relevant documents is presented to the user. In this way a new scenario to use OWA in HTML web document retrieval has been implemented, which was not done in existing works.

This paper is organized as follows. Section 2 provides basic concepts of OWA and IOWA operators, which will be used throughout paper. Section 3 presents a brief review of related work models of the OWA operators and the models of various weighing schemes for HTML tags. In Section 4 we propose a new OWA operator namely Extended-IOWA. In section 5 we illustrate the proposed method using an example to generate final ranked list of relevant documents. Section 6 represents implementation results and at the last conclusion with future scope is described in Section 7.

II. Preliminaries Owa And Iowa Operators

In classical binary logic there are two extremes; “and”, where all the criteria should be met, and “or”, where at least one of the criteria should be met. Yager (1988) proposed OWA operator for aggregation in MCDM to form an overall decision function, which lies between the two extremes. This operator is different than the classical weighted average in that coefficients are not associated directly with a particular attribute but rather to an ordered position.

OWA operator of dimension n is defined as a mapping F: $I^n \rightarrow I$ (where I = [0,1]), with associated weighing vector W,

$$W = \begin{bmatrix} W_1 \\ W_2 \\ . \\ . \\ W_n \end{bmatrix}$$

Such that

1. $W_i \in [0,1]$
2. $\Sigma W_i = 1$
3. F $(a_1, a_2, . . . , a_n) = w_1 \times b_1 + w_2 \times b_2 + . . + w_n \times b_n$ where b_i is the i^{th} largest element in the collection $(a_1, a_2, . . , a_n)$.

The weighing vector can be obtained by two approaches. In the first approach some kind of learning mechanism is used. In the second approach some kind of semantics is given to each of the W_i’s. In this approach we calculate weights using the linguistic quantifier $Q(r) = r^{\alpha}$ where $\alpha \in [0,1]$, as

$$W_i = Q\left(\frac{i}{n}\right) - Q\left(\frac{i-1}{n}\right) \qquad (1)$$

For more details of OWA and its properties please refer [4].

Mitchell and Estrakh (1997) described a modified OWA operator in which the input arguments are not rearranged according to their values but rather using a function of the arguments [17]. Inspired by this work, Yager and Filev (1998) introduced a more general type of OWA operator, which they named the Induced Ordered Weighted Averaging (IOWA) operator [9]. Yager and Filev (1999) assume that we have n values that we want to aggregate using the OWA weighting vector W, these vectors are denoted as $(a_1, ... , a_n)$. In this more general framework each of a_i values is a component of a more complex object which is represent as a 2-tuple $<u_i, a_i>$ and is denoted as an OWA pair. In this approach for OWA aggregation, the arguments are ordered and form vector B based upon the u_i values. In particular, the OWA aggregation for these OWA pairs is calculated as follows:

$$F_w (<u_{1,} a_1 >, ..., <u_n, a_n>) = W^T B_u \qquad (2)$$

In this way an ordered argument vector B_u is formed, so that b_j is a value of OWA pair having the j^{th} largest u value. In these OWA pairs $<u_i,a_i>$, u_i is referred as the order inducing variable and a_i as the argument variable.

III. Related Work

Youssef basil and Paul Semaan (2012) have been used a vector model called SWVM (Semantic-Sensitive Web Vector Model) and a weighting scheme called BTF-IDF (boosted term frequency- inverse document frequency), particularly designed to support the indexing and retrieval of HTML web documents [3]. The chief advantage of the this model is that it assigns extra weights for terms that appear in certain pre-specified HTML tags that are correlated to the semantics of the document and the disadvantage is that this model is not applying fuzzy logic and not able to deal with uncertainty.

On basis of priorities, HTML tags can be categorized. Different classes contain a set of HTML tags according to characteristic/functionalities**.** Classification approach facilitates a simpler way to assign weights to tags. Jiawei Deng and Lihui Chen (2000) proposed automatic categorization of HTML Web Documents with similar contents, Categorization through Fuzzy Representation and HAC (Hierarchical Agglomerative

Clustering) [5]. The main drawback of this model is that it deals with single query terms. It fails if we have multiple query terms which are present in more than one class.

In 1988, Ronald R. Yager developed a new averaging operator i.e. OWA (ordered weighted averaging) operator in multi-criteria/group decision making, which is primarily concerned with the problem of aggregating Multi-criteria to form an overall decision function [4]**.** Yager then introduced a more general type of OWA operator called the Induced Ordered Weighted Averaging (IOWA) Operator [9]. These operators take as their argument pairs, called OWA pairs, in which one component is used to induce an ordering over the second components which are then aggregated. Consequently modified version of IOWA were Proposed viz. I-IOWA (Importance-Induced Ordered weighted Averaging Operator), P-IOWA (Preference-Induced Ordered weighted Averaging Operator), & C-IOWA (Consistency- Induced Ordered weighted Averaging Operator) to solve group decision-making problems based on fuzzy preference relations [7]. These operators are not able to work in a dynamic environment of multi-criteria decision making, which implies that in these operators we know the number of alternatives prior for which experts have to give their opinions, however in document retrieval we can’t estimate the number of retrieved documents (alternatives) for different queries in earlier stage.

Linguistic terms deal with uncertainty in information that can be represented by trapezoidal fuzzy numbers and used in decision making problems [1].

IV. Extended- Iowa

In Group Decision Making (GDM) or Multi-criteria Decision Making (MCDM) problems, an importance/priority weights are assigned to each alternative [7].

Molinari and G. Pasi [18] used fuzzy logic to develop a principled approach for assigning weights to different components, which are specified by tags, in a HTML document. The underlying principle is that a word in title carries much more weight than the same word appearing in other portion of the document. Therefore, we can sort tags based on their degree of importance. So fuzzy importance weight w_i can be calculated for each tag:

$$w_i = \frac{(n-i+1)}{\sum_{i=1}^{n} i} \qquad (3)$$

Where n is the total number of tags in the sorted list.

The numeric weights w_i associated with the tag_i can be computed by assuming that the tags are equidistant [5]. A numeric integer n-i+1 can be associated with tag_i, and its normalized importance weight w_i can be computed as:

$$w_i = \frac{(n-i+1)}{\frac{1}{2}n(1+n)} \qquad (4)$$

In our model, we calculate fuzzy priority weight for each class rather than each tag to reduce the overhead to assign priority weights to each and every HTML tag separately.

Yager [13] suggested a procedure for obtaining the OWA weights from a function f, called a Basic Unit-interval Monotonic (BUM) function. These functions are particularly useful in situation in which the imperative guiding the OWA aggregation is expressed linguistically by a quantifier. A BUM function is a mapping f : [0,1]→[0,1] defined such that f(0)=0, f(1)=1 and if x > y then f(x) ≥ f(y). An OWA aggregation guided by this BUM function can be obtained as:

$$F_f\,(a_1, a_{2,\,\ldots,}\, a_n) = \sum_{j=1}^{n} w_j b_j \qquad (5)$$

Where, b_j is the j^{th} largest of the arguments. Using the BUM function we obtain the n components of the OWA weighting vector as:

$$w_j = f\left(\frac{j}{n}\right) - f\left(\frac{j-1}{n}\right) \qquad (6)$$

Consequently, Yager [12] extended the use of these BUM functions in the IOWA environment. In the area of quantifier guided aggregations, Yager [13] presents a procedure to evaluate the overall satisfaction of important criteria (tag classes) by the alternative x_i (retrieve document). In this procedure, once the satisfaction values to be aggregated have been ordered, the weighting vector associated to an OWA operator using a linguistic quantifier Q is calculated using following the expression:

$$w_k^Q = Q\left(\frac{S(k)}{S(n)}\right) - Q\left(\frac{S(k-1)}{S(n)}\right) \qquad (7)$$

Being $S(k) = \sum_{l=1}^{k} w_{\sigma(l)}$, and σ the permutation used to produce the ordering of the values to be aggregated, where n is number of classes, k is class index, s(k) is importance weight of k^{th} class and s(n) is total weight of all classes.

In our work we have been used w_k^Q (obtained by eq (7)) instead of simple w_i (obtained by eq (4)) as an importance weights of classes because w_k^Q uses Linguistic Quantifier. Linguistic Quantifiers deals with vagueness or uncertainty in information, which is a core application of fuzzy logic. So, in our proposed E-IOWA (Extended-Induced Ordered Weighted Averaging) we have been used importance weights of classes (derived from eq.4 and eq.7 respectively) as order inducing variables. Consequently it generates a new method to calculate order inducing variables.

In our proposed model we use E-IOWA to aggregate the fuzzy linguistic ranking matrix $(C_{\text{ij}})_{m\times m}$ of all four classes, which describes preference of one document to another within a class. These preferences are measured as linguistic terms which are represented by trapezoidal fuzzy numbers. These linguistic terms are used to provide orders to documents within classes. Linguistic terms shows uncertainty, so we apply trapezoidal fuzzy numbers to deal with this uncertainty. The general procedure for the inclusion of importance weight values in the aggregation process involves the transformation of the preference values, C_{ij} under the importance weight w_k^Q to generate a new collective value C_{ij}^c such that:

$$\phi_Q(C_{ij}^1, \ldots, C_{ij}^n) = \sum_{k=1}^{n} w_{\sigma(k)}^Q \,.\, C_{ij}^k \qquad (8)$$

$(C_{\text{ij}})_{m\times m}$ is fuzzy linguistic ranking matrix of class, w_k^Q are importance weight of kth class, σ : {1,..., n} → {1,..., n} a permutation such that $w_{\sigma(k)}^Q > w_{\sigma(k+1)}^Q, \forall k = 1...$ n-1, i.e. $< w_{\sigma(k)}^Q, C_{ij}^{\sigma(k)} >$ is the 2-tuple with $w_{\sigma(k)}^Q$ the largest value in the set $\{w_1^Q, \ldots, w_n^Q\}$ where w_k^Q is defined in eq.7 and n is number of Tag classes (C_1, C_2, C_3 & C_4). Consequently the set of values to be aggregated, $\{C_{ij}^1, \ldots, C_{ij}^n\}$, is induced by the set of values $\{w_1^Q, \ldots, w_n^Q\}$ associated to them, which is based upon their magnitude.

V. Numerical Illustration

In this section we demonstrate our proposed method (section-4) to retrieve the web documents. We follow following steps to illustrate numerical example:

Step1: firstly, categorize HTML tags into different classes. This categorization is done on the basis of HTML tags priority. In this categorization we use HTML 5.0 version. We define four classes title (C_1), header (C_2), emphasized (C_3) and Delimiters (C_4) to cover the most of functionalities of tags that are mainly relevant to text retrieval.

Rank	**Class Name**	**Tags**
1st	Title (C1)	<title>, <meta>
2nd	Header (C2)	<h1>, <h2>, <h3>, <h4>, <h5>, <h6>, <header>
3rd	Emphasized (C3)	<b>, <strong>, <abbr>, <em>, <i>, <mark>, <form>, <map>, <figure>, <footer>, <summary>, <base>, <cite>, <u>, <q>, <blockquote>, <a>, <area>, <embed>, <link>, <lable>, <param>, <nav>, <source>, <span>, <sub>, <script>, <aside>, <article>
4th	Delimiters (C4)	<body>, <code>, <dfn>, <var>,<section>, <p>, <div>, <bdi>, <dl>, <ul>, <ol>, <option>, <table>

Table 1. proposed tag ranking table for the HTML 5.0

Suppose user entered a query: "use of OWA operators in information retrieval".
After applying pre-processing on query we get following query terms:

Use Owa Operator Information Retieval (5terms)

Let there are 8 documents which are relevant to our query. In present research work we consider only those documents in which query terms satisfy at least 2 classes out of 4 classes. The reason behind is that in some HTML documents terms are described in one tag only & there is no further description is present in other tags, these type of documents are useless for user.

E.g. A document in which term "owa" is present in title but no other description is given in any other tag.

In this way some of documents satisfy two classes, some satisfy three classes and some other may satisfy all four classes.

Suppose classes C_1, C_2, C_3 & C_4 categorize following set of documents based on included tags.

C_1=d_5, d_8, d_2, d_1, d_3
C_2=d_3, d_6, d_4, d_8, d_7
C_3=d_8, d_2, d_3, d_4, d_5, d_6, d_7
C_4=d_2, d_3, d_1, d_5, d_6, d_8

Ordering of documents in a class is based on occurrence of number of query terms (in case of multiple query terms) and term frequencies.

Since it can't predictable that how many documents are generated by classes for any query. So each class can generate different number of documents on basis of its tags.

Step 2**:** In this step we assign weights to classes as follows:

As discussed in section 3, weights for ith class is calculated by eq (4) as follows,

$$w_i = \frac{(n-i+1)}{\frac{1}{2}n(1+n)}$$

Since, we calculate fuzzy importance weight for each class rather than for each tag. In our example n=4 because we have 4 classes, so

$$w_1 = \frac{4-1+1}{\frac{1}{2}\times 4\times(1+4)} = 0.4$$

Similarly, for $w_2 = 0.3$, $w_3 = 0.2$ and $w_4 = 0.1$

We now produce important weight w_i^Q for each class using linguistic quantifier Q.

$Q(r) = r^{\alpha}$; α is a positive parameter and n is number of classes.

For class 1 put i=1, n=4 and α=.5 (“at most” quantifier) in eq (7)**,**

$$w_1^Q = \left(\frac{0.4}{1}\right)^{0.5} - (0) = 0.632 \qquad (9)$$

Similarly, w_2^Q=0.205, w_3^Q=0.112, w_4^Q=0.051

Step 3**:** In this step we assign linguistic terms to the documents (to deal with uncertainty) for each class C_i, where i=1, …, 4 as follows:

Let d = (d_1, d_2, …, d_8) represents the set of relevant set of documents to our query ,we choose the following linguistic scale to assign linguistic terms to documents on the basis of their order in each class by using following linguistic scale:

S = {EL = Extremely Low, VL = Very Low, L = Low, M = Medium, H = High, VH = Very High, EH = Extremely High}

We now construct a fuzzy linguistic ranking matrix $(C_{ij})_{m\times m}$, where the diagonal elements in C_{ij} are expressed as “M”, which means “medium”, and the other elements in C_{ij} are taken from set S accordingly. This matrix contains importance relations of one document to another for each class.

For mathematical computation we represent each linguistic terms by trapezoidal fuzzy numbers to provide numerical values to linguistic terms as follows:

EH = (0.7, 0.8, 0.9, 1), VH = (0.6, 0.7, 0.8, 0.9), H = (0.5, 0.6, 0.7, 0.8), M = (0.4, 0.5, 0.6, 0.7), L = (0.3, 0.4, 0.5, 0.6), VL = (0.2, 0.3, 0.4, 0.5), EL = (0.1, 0.2, 0.3, 0.4)

Where, EH > VH > H > M > L > VL > EL

$$\begin{array}{c|ccccc|}
 & d_1 & d_2 & d_3 & d_5 & d_8 \\
\hline
d_1 & M & L & H & VL & L \\
d_2 & H & M & H & L & L \\
d_3 & L & L & M & VL & VL \\
d_5 & VH & H & VH & M & H \\
d_8 & H & H & VH & L & M
\end{array}$$

Figure 1: Fuzzy linguistic ranking matrix for C_1 with Linguistic terms

Similarly for C_2, C_3 & C_4. Fig.1 shows preference of one document to another document.

Since these linguistic values are fuzzy values therefore we have to convert these values into crisp values (since these are argument values and these should be exact values). So for defuzzification, we apply center of gravity method (COG) [19].

$$\text{For, EH} = \frac{0.7+0.8+0.9+1}{4} = 0.85 \qquad (10)$$

Similarly, VH = 0.75, H = 0.65, M = 0.55, L = 0.45, VL = 0.35, EL = 0.25.

$$\begin{array}{c|ccccc|}
 & d_1 & d_2 & d_3 & d_5 & d_8 \\
\hline
d_1 & 0.55 & 0.45 & 0.65 & 0.35 & 0.45 \\
d_2 & 0.65 & 0.55 & 0.65 & 0.45 & 0.45 \\
d_3 & 0.45 & 0.45 & 0.55 & 0.35 & 0.35 \\
d_5 & 0.75 & 0.65 & 0.75 & 0.55 & 0.65 \\
d_8 & 0.65 & 0.65 & 0.75 & 0.45 & 0.55
\end{array}$$

Figure 2: Fuzzy linguistic ranking matrix for C_1 with defuzzified crisp values

Fig. 2 shows the fuzzy linguistic ranking matrix for C_1 with defuzzified crisp values.

Similarly, this matrix will be made for C_2 of dimension (5×5), C_3 of dimension (7×7) & C_4 of dimension (6×6) according to number of documents produced by classes. Fuzzy linguistic ranking matrix $(C_{ij})_{\times m}$, for every class may be heterogeneous in dimension, because of unpredictable number of documents fetched by classes for any dynamic query.

Step 4: In this step we apply the proposed E-IOWA for aggregating the fuzzy linguistic ranking matrix for C_i, where i=1, …, 4. This is accomplished in two phase viz. aggregation phase and exploitation phase.

Aggregation Phase:

In aggregation phase, we calculate $\tilde{C}_i = C_i \times w_i^Q$ where i=1, …, 4. This matrix contains importance relations of documents to each other by defuzzified trapezoidal fuzzy numbers.

For class C_1: $\tilde{C}_1 = [C_1] \times w_1^Q$ (From eq. 7)

$$\tilde{C}_1 = \begin{bmatrix} 0.55 & 0.45 & 0.65 & 0.35 & 0.45 \\ 0.65 & 0.55 & 0.65 & 0.45 & 0.45 \\ 0.45 & 0.45 & 0.55 & 0.35 & 0.35 \\ 0.75 & 0.65 & 0.75 & 0.55 & 0.65 \\ 0.65 & 0.65 & 0.75 & 0.45 & 0.55 \end{bmatrix} \times 0.632$$

Similarly, we find $\tilde{C}_2$, $\tilde{C}_3$ and $\tilde{C}_4$

Next, we calculate the collective fuzzy linguistic ranking matrix $(C_{ij}^c)_{8\times8}$ for all relevant documents (d1, …, d8) respectively as follows:

$$C_{ij}^c = [\tilde{C}_1] + [\tilde{C}_2] + [\tilde{C}_3] + [\tilde{C}_4] \qquad (11)$$

$$\begin{array}{c|cccccccc} & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 & d_7 & d_8 \\ \hline d_1 & 0.37565 & 0.30735 & 0.43375 & 0 & 0.25435 & 0.03315 & 0 & 0.32265 \\ d_2 & 0.44395 & 0.43725 & 0.51675 & 0.0728 & 0.40665 & 0.12225 & 0.0952 & 0.37815 \\ d_3 & 0.31755 & 0.35775 & 0.55000 & 0.20605 & 0.32715 & 0.25550 & 0.23775 & 0.46360 \\ d_4 & 0 & 0.0504 & 0.14265 & 0.17435 & 0.0728 & 0.16505 & 0.21725 & 0.17245 \\ d_5 & 0.49695 & 0.46785 & 0.54735 & 0.0504 & 0.43725 & 0.10595 & 0.0728 & 0.48315 \\ d_6 & 0.02295 & 0.05705 & 0.14930 & 0.18365 & 0.07335 & 0.20240 & 0.22655 & 0.19440 \\ d_7 & 0 & 0.028 & 0.10095 & 0.13145 & 0.0504 & 0.12215 & 0.17435 & 0.12025 \\ d_8 & 0.42865 & 0.493535 & 0.63640 & 0.17625 & 0.39135 & 0.21040 & 0.22845 & 0.55000 \end{array}$$

Figure 3: Collective fuzzy linguistic ranking matrix C_{ij}^c for all relevant documents

Exploitation Phase:

In the exploitation phase, the weights associated with an aggregation of degree n are obtained as follows:

$$w_j = Q\left(\frac{j}{n}\right) - Q\left(\frac{j-1}{n}\right) \qquad \text{For } j = 1\ldots n \qquad \text{[from eq (1)]}$$

Therefore we get $w_{1,\ldots,}$ w_n weights for n documents. (Where n = 8)

Using the fuzzy linguistic quantifier ''most of'' ($\alpha = 0.5$), $Q(r) = r^{\alpha}$,

$w_1 = 0.3535$, $w_2 = 0.1464$, $w_3 = 0.1124$, $w_4 = 0.0947$, $w_5 = 0.0835$, $w_6 = 0.0755$, $w_7 = 0.0694$, $w_8 = 0.0646$.

At this point, in order to select the 'Best' acceptable ranked list for the majority (Q) of the documents we apply simple weighted average process of OWA to the collective fuzzy linguistic ranking matrix.

$$F(a_1, a_2, \ldots, a_n) = w_1 \times b_1 + w_2 \times b_2 + \ldots + w_n \times b_n \qquad (12)$$

Where b_i is the i^{th} largest element in the collection $(a_1, \ldots, a_n)$ and $(w_1, \ldots, w_n)$ are the OWA weight of all documents and here $(b_1, \ldots, b_n)$ are values of collective fuzzy linguistic ranking matrix in such a way:

$d_1 = 0.43375\times0.3535 + 0.37565\times0.1464 + 0.32265\times0.1124 + 0.30735\times0.0947 + 0.25435\times0.0835 + 0.03315\times0.0755 + 0\times0.0694 + 0\times0.0646 = 0.29742$

Similarly, for d_2= 0.38743, d_3=0.40907, d_4=0.15823, d_5=0.41766, d_6=0.1724, d_7=0.12017, d_8=0.47727

Finally, we get a final ranked list of relevant documents,

$$d_8 > d_5 > d_3 > d_2 > d_1 > d_6 > d_4 > d_7$$

VI. Simulation Framework And Result Discussion

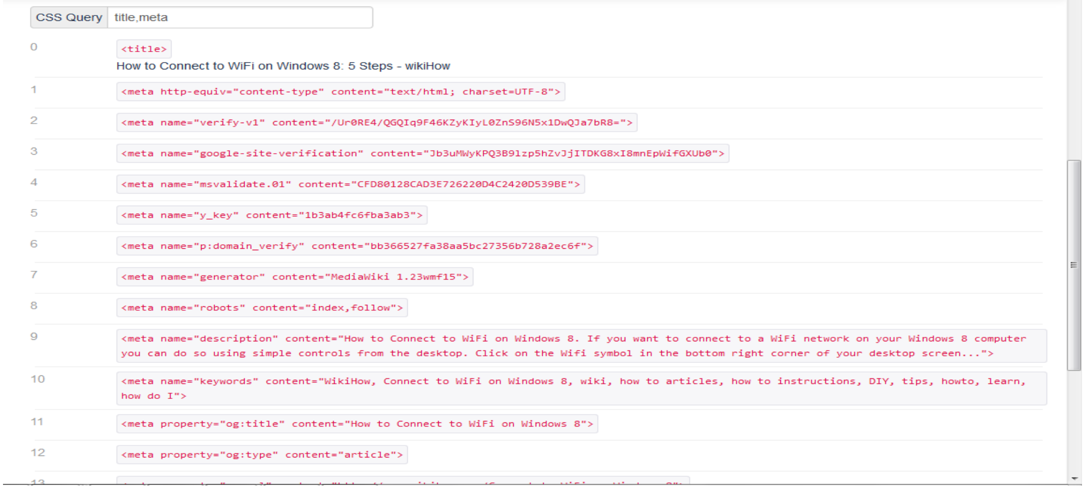
In our proposed system there is a requirement to track the HTML tags which contain query terms. For this purpose we have used Jsoup tool which shows all terms that contained by a HTML tag. Since our proposed concept of E-IOWA (Extended –Induced Ordered Weighted Averaging) basically use matrix operations, therefore we opted MATLAB to implement matrix manipulation. In this fashion we have implemented our system using Jsoup as a tool (jsoup: Java HTML Parser) and MATLAB as a programming language.

jsoup: Java HTML Parser

jsoup is a Java library for working with real-world HTML. It provides a very convenient API for extracting and manipulating data, using the best of DOM (document object model), CSS, and jquery-like methods. Jsoup is an open source project distributed under the liberal MIT license. We use jsoup API and modified its source code to fetch URLs of relevant documents from Google search engine.

MATLAB R2010a is made to interact with the java program developed in Netbeans IDE, using MatlabControl Java API [20-24] and is used to evaluate the ranking of documents by applying Extended -IOWA operator. The result of aggregation has been achieved using MATLAB. Finally, the re-ordered result is echoed in implemented system interface.

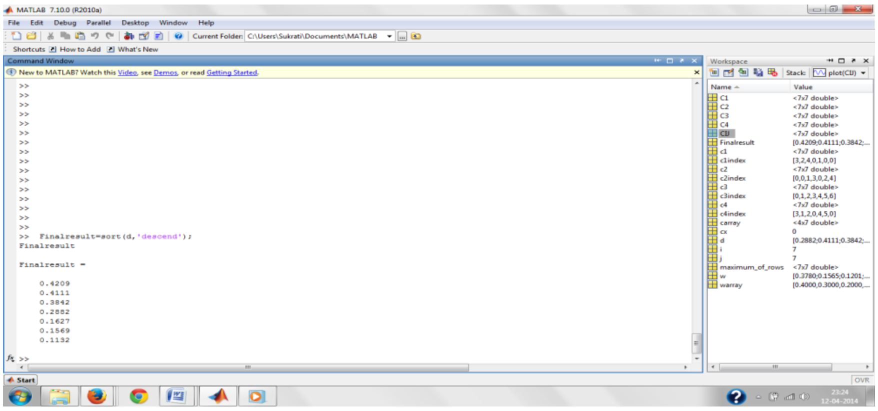
Use jsoup API and modified its source code to fetch top ten results from Google search engine. Jsoup tool saves these results as URLs. Each fetched document is parsed by jsoup tool on the basis of HTML tag classes. For the query “how to connect wifi in windows 8”, we select seven URLs out of top ten (Must be text/*, application/xml, or application/xhtml+xml) and parse them using jsoup tool. Fig. 4 shows result of jsoup tool as:



```
title,meta

0   <title>
    How to Connect to WiFi on Windows 8: 5 Steps - wikiHow
1   <meta http-equiv="content-type" content="text/html; charset=UTF-8">
2   <meta name="verify-v1" content="/Ur0RE4/QGQIq9F46KZyKIyL0ZnS96N5x1DwQJa7bR8=">
3   <meta name="google-site-verification" content="Jb3uMWyKPQ3B91zp5hZvJjITDKG8xI8mnEpWifGXUb0">
4   <meta name="msvalidate.01" content="CFD80128CAD3E726220D4C2420D539BE">
5   <meta name="y_key" content="1b3ab4fc6fba3ab3">
6   <meta name="p:domain_verify" content="bb366527fa38aa5bc27356b728a2ec6f">
7   <meta name="generator" content="MediaWiki 1.23wmf15">
8   <meta name="robots" content="index,follow">
9   <meta name="description" content="How to Connect to WiFi on Windows 8. If you want to connect to a WiFi network on your Windows 8 computer
    you can do so using simple controls from the desktop. Click on the Wifi symbol in the bottom right corner of your desktop screen...">
10  <meta name="keywords" content="WikiHow, Connect to WiFi on Windows 8, wiki, how to articles, how to instructions, DIY, tips, howto, learn,
    how do I">
11  <meta property="og:title" content="How to Connect to WiFi on Windows 8">
12  <meta property="og:type" content="article">
```

Figure 4: Document parsed in HTML tags of class c1

Matlab is mainly used for matrix manipulation and to support different matrix operations. Jsoup tool assigns different set of documents to all four classes on the basis of included tags that contain query terms. Afterward further computation is done by Matlab R2010a. Matlab R2010a generates Linguistic matrices for each class and performs all steps illustrated in section 5 and finally produces a final ranked list of relevant documents as shown in fig. 5.



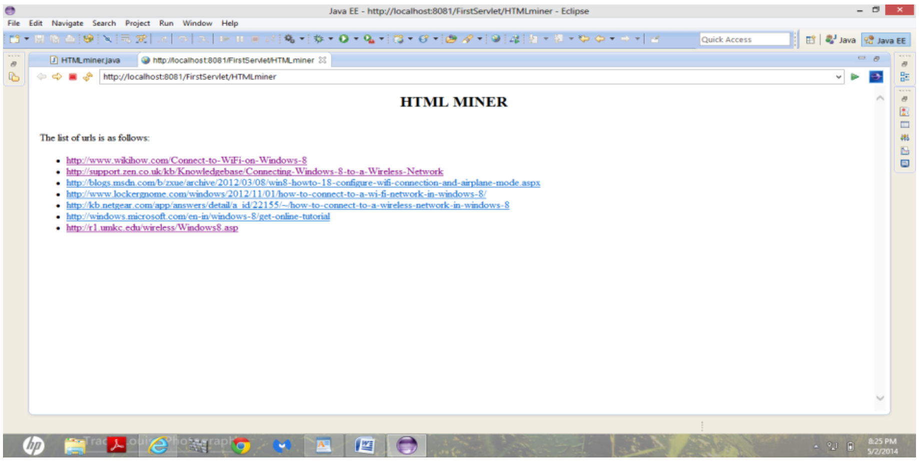
MATLAB 7.10.0 (R2010a)
File Edit Debug Parallel Desktop Window Help
Current Folder: C:\Users\Sukrati\Documents\MATLAB
Shortcuts How to Add What's New
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Workspace
Stack: plot(CIJ)
OVR
Start
23:24
12-04-2014

```
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>  Finalresult=sort(d,'descend');
Finalresult

Finalresult =

    0.4209
    0.4111
    0.3842
    0.2882
    0.1627
    0.1569
    0.1132

>>
```

Name	Value
C1	<7x7 double>
C2	<7x7 double>
C3	<7x7 double>
C4	<7x7 double>
CIJ	<7x7 double>
Finalresult	[0.4209;0.4111;0.3842;...
c1	<7x7 double>
c1index	[3,2,4,0,1,0,0]
c2	<7x7 double>
c2index	[0,0,1,3,0,2,4]
c3	<7x7 double>
c3index	[0,1,2,3,4,5,6]
c4	<7x7 double>
c4index	[3,1,2,0,4,5,0]
carray	<4x7 double>
cx	0
d	[0.2882;0.4111;0.3842;...
i	7
j	7
maximum_of_rows	<7x7 double>
w	[0.3780;0.1565;0.1201;...
warray	[0.4000,0.3000,0.2000,...

Figure 5: Final values of relevant documents in descending order

The final result of proposed system has been represented by fig. 6 in the form of URLs.


Java EE - http://localhost:8081/FirstServlet/HTMLminer - Eclipse
File Edit Navigate Search Project Run Window Help
Quick Access
Java
Java EE
HTMLminer.java
http://localhost:8081/FirstServlet/HTMLminer
http://localhost:8081/FirstServlet/HTMLminer

HTML MINER

The list of urls is as follows:

- http://www.wikihow.com/Connect-to-WiFi-on-Windows-8
- http://support.zen.co.uk/kb/Knowledgebase/Connecting-Windows-8-to-a-Wireless-Network
- http://blogs.msdn.com/b/zxue/archive/2012/03/08/win8-howto-18-configure-wifi-connection-and-airplane-mode.aspx
- http://www.lockergnome.com/windows/2012/11/01/how-to-connect-to-a-wi-fi-network-in-windows-8/
- http://kb.netgear.com/app/answers/detail/a_id/22155/~/how-to-connect-to-a-wireless-network-in-windows-8
- http://windows.microsoft.com/en-in/windows-8/get-online-tutorial
- http://r1.umkc.edu/wireless/Windows8.asp

Figure 6: Final Result on system interface

Efficiency of proposed System: -

In this presented work we have focused on some shortfalls of existing systems which deal with HTML web domain and we try to confiscate, thus proposed system have been given solutions of these shortfalls which prove our system efficient and superior to existing HTML web document retrieval systems as follows:

I. In HTML documents it is possible that a query term is present in only title tag (for example it may be composed in a rhetorical way) [11], however title tag is consider as a highest prioritize tag, so document that contain query term in title tag but not in any other tag (in other part of document) is kept first by IR system, which a not a efficient way of retrieval. So in the proposed system prioritizes tags and assigns fuzzy weights which show importance of each class.

II. Treating each tag independently is not an optimized way as represented in previous work [5] [3] therefore we divide HTML tags (most of all tags, which are responsible for text retrieval) into four number of classes that cover more number of HTML tags. Proposed system not only depends on term frequency to rank the documents, it also applies HTML tag priorities.

III. It is also possible that query terms are present in more than one tag (with different priorities) in same document and in more than one documents then how to merge all document weights and present in a final ranked result list proficiently, this problem is solved by proposed operator (Extended-Induced Ordered Averaging Operator) to aggregate all the document values.

IV. It has been proved [10] [16] [11]; that in comparison to Boolean model fuzzy logic deals with vagueness/uncertainty of information (Unlike the Boolean model that is based on binary decision criterion {relevant, not relevant}), fuzzy logic expresses relevance as degrees of memberships and gives more precise results, so in our proposed model we use fuzzy weights of tag classes and fuzzy trapezoidal numbers to assign importance of documents within classes.

In this presented work a final ranked list of relevant documents has been generated by applying E-IOWA. So we can compare this ranked list to Google ranked list which has been generated without applying E-IOWA. We have used Google search engine to retrieve top ten URLs because of its better efficiency among all other search engines [25-27]. For this comparison we use Kendall's tau distance between proposed system ranked list and Google ranked list to measure efficiency of this implemented system. The Kendall tau rank distance is a metric that counts the number of pair-wise disagreements between two ranking lists [28-30]. The larger the distance, the more dissimilar the two lists are. Therefore, on the basis above theoretical points and Kendall Tau Distance we have proved efficiency of our proposed system as follows:

The Kendall tau rank distance deals with inverted pairs (i, j) of lists (l_1, l_2), in which i, j are in the opposite order in l_1, l_2. We have been used Kendall tau distance formula from (Desarkar et al. 2011) as:

$$K_{ij}(l_1,l_2) = \frac{(d_1{}^2+d_2{}^2)}{4n^2} \qquad (13)$$

d_1 = absolute rank distance between document i and j in ranked list l_1

d_2 = absolute rank distance between document i and j in ranked list l_2

n = Total number of documents in ranked lists l_1, l_2

The distance KTDispSq between two ranked lists is computed as the summation of distance of each inverted pair, calculated by eq. 13. We have two ranked list first one has been obtained by Google (without E-IOWA) and second one has been obtained from presented system (with E-IOWA):

Ranked List Of Google	d_1	d_2	d_3	d_4	d_5	d_6	d_7
Ranked List of Implemented system	d_1	d_3	d_2	d_5	d_6	d_4	d_7

Now we have to calculate the distance between these two ranked lists. Here we have three inverted pairs (2,3) (4,5) (4,6) and n = 7 (total number of documents in list l_1 and l_2). Inverted pair weights for KTDispSq can be calculated as follow:

$$(2,3) = \frac{1^2+1^2}{4\times7^2} = \frac{2}{196} = 0.01020$$

Similarly for (2,5) & (4,6) inverted pair. Hence total distance KTDispSq between two lists is:
(0.01020 + 0.02551 + 0.02551 = 0.06122)
A value of 0.06122 indicates that ≈ 6.122% of pair differ in ordering between the two list.

The result shows that our method retrieves all most similar results as Google with 6.122% dissimilarity in HTML web document domain.

VII. Conclusion & Future Work

In this paper a new approach has been proposed for information retrieval of structured documents. The main attainment of this new approach is that through our approach, we are able to consider the possible dependencies among the fields that form the document structure. Additionally our model has employed classification of HTML tags according to priorities. We have defined E-IOWA (Extended-Induced Ordered Averaging Order) which is used a new method for calculating order inducing variables in HTML web domain. In our work we find that if query terms are present in more than one class in documents, then E-IOWA proposed by us in this paper is able to aggregate the rank of all those documents in more efficient way and present a final ranked list.

In future research, the present work may be extended to categorize and sort web documents according to semantics and synonyms of query terms to facilitate semantic scope of IR system.

References

[1] Guiwa Wei & wende Yi, Induced Trapezoidal Fuzzy Ordered Weighted Harmonic Averaging Operator. Journal of information & Computational Science, vol. 7, issue no. 3, 2010, pp. 625-360.

[2] R.R. Yager, D.P. Filev, Operations for granular computing: Mixing words and numbers. In: Proceedings of the FUZZ-IEEE World Congress on Computational Intelligence, Anchorage, vol. 1, 1998, pp. 123–128.

[3] Youssef Bassil & Paul Semaan , Semantic-Sensitive Web Information Retrieval Model for HTML Document. European Journal of Scientific Research, ISSN 1450-216X, vol. 69(4), 2012.

[4] Ronald R. Yager , On Ordered Weighted Averaging Aggregation Operators Multi-criteria Decision making. IEEE Transactions on Systems, Man, And Cybernetics, vol 18, issue no. 1, 1988, pp. 183-190.

[5] Jiawei Deng and Lihui Chen, Web Documents Categorization Using Fuzzy Representation and HAC. In: Proceedings of the IEEE First International Conference, vol. 2, 2000, pp. 24-28

[6] John Yen & Reza langari, Fuzzy Logic Intelligence, Control and Information (Pearson publication, 2011), 343-370.

[7] F. Chiclana a, E. Herrera-Viedma b, F. Herrera b, S. Alonso, Some induced ordered weighted averaging operators and their use for solving group decision-making problems based on fuzzy preference relations. European Journal of Operational Research, vol. 182, 2007, pp. 383–399.

[8] R. R. Yager, and V. Kreinovich, On how to merge sorted lists coming from different web search tools. Soft Computing Research Journal, vol. 3(1), 1999, pp. 83-88

[9] Ronald R. Yager and Dimitar P. Filev, Induced Ordered Weighted Averaging Operators. IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics, vol. 29, no. 2, 1999.

[10] Nowacka, K., S. Zadrozny, and J. Kacprzyk, A new fuzzy logic based information retrieval model. In:12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2008), Malaga, Spain, 2008.

[11] Pérez-Iglesias, Joaquín, Víctor Fresno, and J. R. Perez-Aguera, Modelling field dependencies on structured documents with fuzzy logic. In Fuzzy Systems: IEEE International Conference, 2009, pp. 496-501.

[12] Ronald R.Yager, Induced aggregation operators. Fuzzy Sets and Systems 137, issue no. 1, 2003, pp. 59–69.

[13] R.R. Yager, Quantifier guided aggregation using OWA operators. International Journal of Intelligent Systems 11, 1996, pp.49–73

[14] W3 schools, http://www.w3schools.com

[15] His-Ching Lin, Li-Hui Wang, Shyi-Ming Chen, Query expansion for document retrieval based on fuzzy rules and user relevance feedback techniques. Expert Systems with Applications 31, 2006, pp.397-405

[16] N.O. Rubens, The Application of Fuzzy Logic to the Construction of the Ranking Function of Information Retrieval Systems. Computer Modelling and New Technologies, vol.10 (1), 2006, pp. 20-27

[17] H.B. Mitchell, D.D. Estrakh, A modified OWA operator and its use in Lossless DPCM image compression. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 5, issue no. 04, 1997, pp. 429–436

[18] Molinari, Andrea and Gabriella Pasi, A fuzzy representation of HTML documents for information retrieval systems. In: Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, vol. 1, 1996, pp. 107-112.

[19] M.Q. Suo, Y.P. Li, G.H.Huang, Multicriteria decision making under uncertainty: An advanced ordered weighted averaging operator for planning electric power systems. Engineering Applications of Artificial Intelligence, vol. 25, Issue 1,2012, pp. 72-81.

[20] Yair Altman, JMI wrapper-local MatlabControl Part 1. http://www.undocumentedmatlab.com/blog/jmi-wrapper-local-matlabcontrol-part-1/, May 2010.

[21] Yair Altman, JMI wrapper-local MatlabControl Part 2. http://www.undocumentedmatlab.com/blog/jmi-wrapper-local-matlabcontrol-part-2/. May 2010.

[22] Kamin Whitehouse, Calling Matlab from Java. http://www.cs.virginia.edu/~whitehouse/matlab/JavaMatlab.html, 2010.

[23] MITOPENCourseware, Matlab tutorial. http://ocw.mit.edu/courses/mathematics/18-06-linear-algebra-spring-2010/related-resources/MIT18_06S10_matlab.pdf, 2014.

[24] Brian Vick, Matlab Commands and functions, Mechanical engineering department Virginia Tech. http://www.hkn.umn.edu/resources/files/matlab/MatlabCommands.pdf, 2011.

[25] Al-Maskari, Azzah, Mark Sanderson, and Paul Clough, The relationship between IR effectiveness measures and user satisfaction. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 2007, pp. 773-774.

[26] Kumar, BT Sampath, and S. M. Pavithra, Evaluating the searching capabilities of search engines and metasearch engines: A comparative study. Annals of Library and Information Studies, Vol. 57, Issue 2, 2010, pp. 87-97.

[27] Kumar, BT Sampath, and J. N. Prakash, Precision and relative recall of search engines: A comparative study of Google and Yahoo. Singapore Journal of Library & Information Management, Vol. 38, Issue 1, 2009, pp. 124-137.

[28] Gayatri, M., and MHM Krishna Prasad, Quantitative Measurement of Scores by Ranks. International Conference on Advancements in Information Technology with workshop of ICBMG 2011 IPCSIT Vol.20, 2011.

[29] Desarkar, Maunendra Sankar, Rahul Joshi, and Sudeshna Sarkar, Displacement based unsupervised metric for evaluating rank aggregation. In Pattern Recognition and Machine Intelligence, Springer Berlin Heidelberg, 2011, pp. 268-273.

[30] Yilmaz, Emine, Javed A. Aslam, and Stephen Robertson, A new rank correlation coefficient for information retrieval. In: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 2008, pp. 587-594.