XML Information Retrieval using ELECTRE III method

Bal Kamal[1], Nouali Omar[2]
Research Center on Scientific and Technical Information, CERIST
Algiers, Algeria
{k_bal@esi.dz, onouali@cerist.dz}

ABSTRACT: *XML information retrieval aims to retrieve relevant document fragments (XML elements) instead of retrieving whole documents. The retrieved XML elements must not only contain relevant information. They must be of good level of granularity. They must be the most specific and the most exhaustive at once. The coexistence of content and structural information in XML documents makes that multiple heterogynous relevance factors (sources) can be used in retrieval process. As in traditional information retrieval, the most majority of XML information retrieval approaches try to model the notion of relevance and use score function to attribute a relevance score to each XML element. The number of relevance factors used simultaneously is then limited due to their heterogeneity. Our goal is to develop an XML information retrieval approach in which several relevance factors can be used in retrieval process. We propose an approach based on relational aggregation paradigm. The relevance process is not guided by the estimation of relevance degree, but by the comparisons of each pair of XML elements in order to determine a final order of all XML elements, from the most relevant to the less relevant.*

Keywords: XML Retrieval, Relevance factors, Relational aggregation, information retrieval, Outranking

Received: 10 May 2010, Revised 14 June 2010, Accepted 20 June 2010

1. Introduction

In traditional information retrieval, a list of relevant documents is returned to the user. The user has to browse the content of returned documents to find fragments containing the desired information. This is because the traditional information retrieval models were developed for searching in flat documents. The documents were considered as bags of words (without structure). In Content-Oriented XML information retrieval, a collection of XML documents (document-Centric view) is confronted to an information need expressed classically by a list of keywords. The aim is to exploit logical structure of XML documents (coded by XML tags) in order to retrieve not a list of documents, but only relevant fragments of documents. Documents fragments correspond to XML elements. These XML elements must not only contain relevant information but must be also of good level of granularity [2].

In this context, several XML information retrieval approaches were developed. These approaches were classified into four types [14]. Scoring approaches, contextualization approaches, propagation approaches and fusion approaches. In scoring approaches, classical information retrieval model (vector space model, probabilistic and language-modeling models) were adapted and used to estimate XML elements relevance by considering evidence from only the element content (considered as an atomic document). For example, in [10] a vector space model is used where the weight of a term in an element is evaluated according to its weights in elements of the same type. OKAPI model was adapted to XML retrieval in [16]. In [12], language-modeling approach is used to estimate the relevance degrees of XML elements. The contextualization approaches consider in addition to evidences from the element itself, other evidences from its context. For example, Bayesians networks based model, where the relevance on an XML element depends on its content and the relevance of its parent elements, is developed in [18]. The propagation approaches are used when only leaves nodes of XML documents are indexed. Works presented in [22] and [9] are examples of propagation approaches. First, the relevance of leaves nodes is estimated by using classical relevance model or relevance formula. Relevance scores of internal nodes are obtained by propagating up relevance scores of leaves nodes in the document. The last type of approaches is the fusion approaches [17], [15]. Here, several lists of results are calculated for the query. A merger mechanism is used to merge all these lists into a single list of results.

All these types of approaches try to model relevance and to estimate relevance (or similarity) degree (score) of each XML element. These scores are obtained by using score functions which reflect the interpretation of the notion of relevance in each approach. The relevance is represented by a unique value (score) estimated for each XML element. As shown in [8], representing relevance by a unique score value reflects generally a global aggregation of relevance sources. Aggregation operators such as “the weighted sum”, “the weighted product”, “similarity measures”, Boolean and fuzzy operators are used in score functions. In each approach, a set of relevance sources (relevance criteria) that have an impact on the relevance of XML element are considered, and then combined (aggregated) to assign a single relevance score to each XML element. XML elements are then returned according to the fading order of their relevance scores. The query terms frequency (*Tf*), Inverted Element Frequency (*Ief*), query terms proximity in XML element content and XML element specificity are among other some of relevance sources used in XML retrieval approaches.

We propose in this paper a different XML information retrieval approach which is not based on global aggregation of relevance sources. The relevance sources will not be combined in a unique score value to assign to each XML element. The retrieval process will not be guided by the estimation of relevance by score function but by the comparison of the all potentially relevant XML elements according to each relevance source. Each pair of XML elements will be compared according to each relevance sources. The preference relations according to each relevance source will be then combined to build a global preference relation allowing comparing each pair of XML elements according to all relevance sources. Finally the global preference relation will be used to produce a final order of XML elements, from the most relevant to the less relevant one. For this aim, we use concepts, tools and methods furnished by the multi-criteria aggregation paradigm.

The remainder of this paper is organized as follows; we will present and describe briefly in section 2 some of relevance sources which are differently used in XML Retrieval approaches. We will discuss in section 3 about motivations of our proposed approach detailed in sections 4. An illustration example is presented in section 5.

2. Relevance sources in Content-Oriented XML retrieval

Several relevance sources were differently considered and used in content-oriented XML information retrieval. Some of these relevance sources are related to the content; some others to the structure and others to the context of XML elements. Relevance sources related to linguistic aspect are also used by some systems. We think that it is necessary to not neglect any of these dimensions in the retrieval process to have good results. Let us describe some of these relevance sources:

- The weight of query terms in the content of XML elements is an important evidence source. It evaluates the quantity of relevant information in an XML element. XML elements in which query terms have important weights are preferable to those in which query terms have less important weights. To estimate this criteria this formula in [22], [23] and [25]:

$$\mathrm{Weight}(e,q) = \sum_{t \in q} \mathrm{tf}(t,e) \times \mathrm{ief}(t,e) \qquad (1)$$

Where *e* is an XML element, *q* is the query, *tf(t,e)* is the term frequency of *t* in *e* and *ief(t,e)* is the Inverted Element Frequency. It is expressed as:

$$\mathrm{ief}(t,e) = \log \frac{N}{\mathrm{ef}(t)} \qquad (2)$$

Where *N* is the total number of XML elements with the same name (tag) as *e*, *ef(t)* is the number of XML elements with the same name as *e* which contain *t*.

- Query terms proximity criterion is widely used as evidence source in information retrieval [6]. The idea of using proximity in the IR was first implemented in Boolean systems with adding ADJ and NEAR operators. ADJ is used to specify that the connected terms must be adjacent. NEAR is used to specify that the connected terms must be in an bounded interval. Other more recent approaches use direct proximity of query terms for calculating the score of relevance of documents. Clarke et al. [6] have developed method for classifying documents taking into account the proximity of terms. The idea is to select ranges of text that contain query terms, and then to give them partial scores (the shorter interval will have the highest score). Intervals scores are then added to give a score to the document. In XML context, two notions of proximity are suggested in [11]: A classical proximity between terms (horizontal proximity) and a vertical proximity in terms of proximity of XML nodes containing query terms. In [1] the proximity between two terms is equals to 1 if they are in the same XML node, else it is equals to the length of the shortest path containing these two terms.

- XML element specificity: The major problem in XML information retrieval is to identify the good granularity level of XML elements to retrieve. For example, if a paragraph element contains relevant information, the section element in which appears this paragraph and the chapter element in witch appears the section element will contain necessary the relevant information. Obviously, the section and the chapter elements can contain also other irrelevant element. The question is what is the good level of granularity of element to retrieve? Is it better to retrieve the paragraph element, the section element or the chapter element? Retrieve the chapter, the section and the paragraph is not desirable to the user. The XML element to retrieve must be then specific, is must concern only the topic of the query; it should not contain a lot of irrelevant information. In [1] and [25], the authors consider that more a relevant element is deeper more it is specific.

- Structural positions of query terms in XML elements: XML elements which contain just a small number of terms are not desirable elements to retrieve even if they contain all query terms [25]. Generally, theses short XML elements are titles elements or formatting element (*italic, bold,…*). In [25], experiments attests that these short XML element reinforce relevance of there ancestor element when they contain several query terms. For example, a *section* element with *title* element containing all query terms will talk necessary about the topic of the query. This importance diminishes while climbing up in the document tree [23]. The relevant title element in the following path */article/section/title/…* will reinforce more the *section* element than the *article* element.

- Contextual relevance: Contextual relevance of XML element refers generally to the relevance of the XML document in which appears the XML element (parent document). This relevance source is considered in [1], [23] and [25]. Experiments in [23] show that considering contextual relevance leads to good retrieval results when we are interesting to retrieve all relevant XML elements (even overlapped). This relevance source, noted here *context(e,d),* can be evaluated with classical TDIDF formula. For an XML element *e* with parent document *d:*

$$\mathrm{Context}(e,d) = \sum_{t\in q} \mathrm{tf}(t,d) \times \mathrm{idf}(t) \qquad (3)$$

Where *tf(t,d)* is frequency of the query term *t* in the document *d* and *idf(t)* is estimated by:

$$\mathrm{idf}(t) = \log \frac{N}{\mathrm{df}(t)} \qquad (4)$$

Where *N* is the total number of XML documents in the collection and *df(t)* is the number of XML documents containing the query term *t.*

We can also talk about other used relevance sources such as the one used in [2] where the content of each XML element is segmented by text segmentation algorithm [26]. A topic shift is considered between two adjacent segments when the vocabulary changes between these two segments. Relevant XML elements are those with the less number of topic shifts in their content. They are the most specific ones.

A study of all relevance sources cited above, shows that they are multiple, heterogeneous and have very variable value scales. If we want to consider many relevance sources in retrieval process regardless of their heterogeneity and their value scales, we do not use relevance score function. The use of score function means that we must aggregate globally (in a unique value) all theses heterogeneous sources. This can present some limits related especially to the diversity and to the heterogeneity of relevance sources:

- Global aggregation of heterogynous and different value scales relevance criteria in a unique value can present a compensation effect between criteria [7], [8]. Heights performance values for some relevance sources can cover (hide) very poor performance value for some others relevance sources.

- Global aggregation limits the number of relevance sources that can be considered simultaneously.

- The unique value aggregation is also very sensitive to very weak criteria values variations. A minimal variation in the performance values of XML element according to some criteria can radically change the order of returned XML element list. For example, if we have two XML element *e1* and *e2* with two relevance sources *r1* and *r2* with the following performance values :
    - For e1 : *r1(0.5), r2(0.5)*
    - For e2 : *r1(0.1), r2(0.9)*

With a score function as weighted sum, *e1* will be strictly better than *e2* if its performance for *r1* pass to 0.51, but *e2* will be better if the performance of *e1* for *r1* pass to 0.49.

Our goal is to develop an XML information retrieval approach in which several relevance factors can be used in retrieval process. We propose an approach based on relational aggregation of relevance criteria. It means that we will not try to estimate relevance score of each XML element according to the query, but we will compare each pair of XML elements according to each relevance criteria. From all these binary comparisons, we will deduce a final order of XML elements according to all relevance criteria, from the most relevant one, to the less relevant one.

3. Motivations

The start point of our approach that we will explain in section 4 is the following motivations:

- In practice, XML information retrieval problem is also a ranking problem. Relevant XML elements must be ranked from the most relevant one to the less relevant one. Users rarely examine the entire list of results, but only the elements top classed elements. The quality (relevance) of the first ranked elements must then be higher and good.

- Relevance in XML retrieval context does not refer only to the content but also to the structure and to the context of XML elements. To have a good quality of results, we should not neglect any of those aspects, more we consider relevance criteria more results will be better.

- In XML information retrieval, relevance criteria are value and kind heterogeneous. It is not evident to define a score function that can aggregate all these criteria in a unique and single value.

- The evaluation of the performance of XML element according to the relevance criteria is often tainted with arbitrariness and vagueness. A very weak difference between performances of two elements according to relevance sources must not necessary discriminate them.

- It is easier to compare each pair of XML elements according to each relevance criterion, than to decide globally which of them is the most relevant.

4. Our approach

The multi-criteria aggregation paradigm aims to furnish tools to resolve decision problems where several points of view or criteria are considered. The multi-criteria aggregation paradigm is that there are several solutions (alternatives) that must be analyzed according to several criteria (criteria can be in conflict) in order to resolve a decision problem [21], [19]. The decision problem can be:

- A choice problem : Choose the best alternative from the set of alternatives,
- A rank problem : Rank the alternative from the best one to less good one,
- A classification problem: Assign each alternative to a predefined category.

Whatever the decision problem is a rank, a choice or a classification, the central question is always a comparison problem. We have to compare the alternatives to see which is better than the other. To do this, the performance of each alternative according to each criterion must be aggregated. In this context, two competitive aggregation approaches exist: the global aggregation and the partial (or relational) aggregation. The first one use score function to aggregate all performance of alternatives in a single and unique value. The second approach consists to compare first of all each pair of alternatives according to each criterion. From all theses comparisons, a global preference relation allowing comparing each pair of alternatives according to all the set of criteria is deduced and used to solve the decision problem. Methods of this approach are called outranking method.

We use tools and concepts of multi-criteria paradigm to develop our approach. We model XML information retrieval problem as a multi-criteria aggregation problem in which:

- The set of alternatives corresponds to the set of all XML elements of the documents collection.
- The set of criteria corresponds to the set of relevance sources that will be considered.
- And the decision problem is here a ranking problem. Potentially relevant XML elements must to be ranked from the most relevant to the less relevant one.

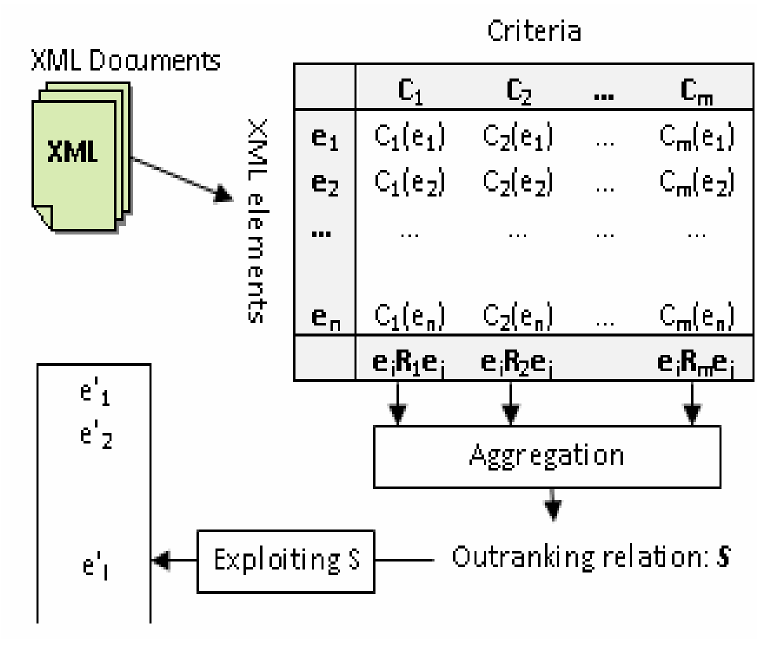
XML Documents
XML
XML elements
Criteria
C_1 C_2 ... C_m
e_1 $C_1(e_1)$ $C_2(e_1)$ ... $C_m(e_1)$
e_2 $C_1(e_2)$ $C_2(e_2)$ ... $C_m(e_2)$
... ...
e_n $C_1(e_n)$ $C_2(e_n)$ ... $C_m(e_n)$
$e_iR_1e_j$ $e_iR_2e_j$ $e_iR_me_j$
Aggregation
Outranking relation: S
Exploiting S
e'_1
e'_2
e'_l

Figure 1. Enchainment of the proposed approach

Relevance will be represented as a multidimensional entity, i.e. the relevance of an XML element will be represented by a set of values. Each value corresponds to the performance of the XML element according to one relevance criterion. For the aggregation of the performance of XML elements, we will use a partial aggregation approach. Each relevance criterion will model a binary preference relation between XML elements according to this criterion. Binary preference relations will be then aggregated to built an outranking relation '*S*' that will model binary preferences of XML elements according to all criteria. Outranking relation 'S' will be next exploited by a ranking algorithm to give in last, a ranked list of relevant XML elements. An element will be ranked higher than another if a coalition of criteria is favorable to this judgment without existing strongly unfavorable criteria to this judgment.

Let us put:

- E : the set of all XML elements.
- C : the set of considered relevance criteria.
- $C_j(e)$: the performance value of an XML element e, according to the relevance criterion C_j.
- For each XML element e, a performance vector $V(e)$ will be associated, such as: $V(e) = (C_1(e), C_2(e), \ldots, C_n(e))$.

The enchainment of the approach, as schematized in Figure 1, consists on the following steps:

- Modeling binary preference relations (R1, R2… Rn).
- Aggregation of preference relations, construction of the outranking relation (S).
- Exploitation of the outranking relation to rank XML elements.

4.1. Modeling binary preference relations

Binary preference relations allow us to compare two XML elements according to each criterion. Since evaluation of XML elements according to each criterion can be imprecise and/or uncertain (several formulas can be accepted to evaluate a criterion), a slight difference between values of two elements for a criterion does not necessarily discriminate them. To model this uncertainty, the following thresholds used in partial aggregation methods [19] will be used to construct gradual (fuzzy) preference relations:

- **The indifference Threshold, noted 'q',** allows two XML elements having very close values for a criterion to be judged equivalent even if they have not the same criterion values.
- **The preference Threshold, noted 'p',** it is introduced when we need more precision for the preference description.
- **The veto Threshold, noted 'v'**: it is the value from which the performance difference between two elements *e2* and *e1* according to a criterion with this threshold, is considered as too large to accept the judgment *"e1 preferable than e2 "*.

Let us consider two elements *e1*, *e2* and a criterion C_j with two thresholds (of indifference and preference) q_j and p_j such as $q_j < p_j$. $C_j(e)$ is the value of the element *e* for the criterion C_j. The preference relation between *e1*and *e2* can take the following states:

- *e1* and *e2* are indifferent noted (*e1 I e2*):
  $(e1\ I\ e2) \Leftrightarrow Cj(e1) - Cj(e2) \leq q_j$
- *e1* is weakly preferable than *e2 (e1 Q e2):*
  $(e1\ Q\ e2) \Leftrightarrow q_j < Cj(e1) - Cj(e2) \leq p_j$
- *e1* is strictly preferable than *e2 (e1 P e2):*
  $(e1\ P\ e2) \Leftrightarrow Cj(e1) - Cj(e2) > Pj$

In addition, for a criterion C_j, when the element *e1* is less good than the element *e2* , we looks if the difference between *e2* and *e1* does not surpass the veto threshold v_j, i.e. a threshold beyond of which it will be prudent to refuse all credibility of the outranking of *e2* by the element *e1* noted *(e1 S e2):*

$$Not\ (e_1\ S\ e_2)\ if\ C_j(e_2) - C_j(e_1) \geq v_j.$$

4.2. Aggregation of preference relations

Binary preference relations according to each criterion will be aggregated to obtain a global binary preference relation '*S*' called outranking relation. This outranking relation will be exploited to resolve the ranking problem. Outranking relation is a concept introduced as a binary relation, defined on the set of alternatives A, such as: Given two alternatives *a* and *b*, "*a outranks b"*, or "*a S b"*, if given all that it is known about the two alternatives, there are enough arguments to decide that *a* is at least as good as *b* without there be important reason to refuse this affirmation [20]. To build our outranking relation, we have to use and follow an outranking method. Outranking methods solve decision problem by constructing an outranking relation. ELECTRE III [19], [24], [4] is a well-known outranking method dedicated to decision ranking problem. It is the first method dedicated to decision ranking problems, which introduce the veto threshold. Its main feature is that it constructs a fuzzy (gradual) outranking relation. ELECTRE III method is based on two basic principles:

- **The concordance principle:** which means that an element *e1* is at least preferable to another element *e2*, if a majority of criterion is in favor of this judgment.

- **The discordance principle:** This principle means that an element *e1* is at least preferable to another element *e2* if any discordant criterion refuses strongly this judgment.

We present here the algorithm used by ELECTRE III to build the outranking relation ‘S’. Obviously, the algorithm is adapted to our case of studies. For each pair of XML elements (*e1*, *e2*), ‘S’ takes a value *S(e1, e2)* reflecting the credibility degree of the assumption “*e1 outrank e2*”. Building *S* requires calculating the following indices:

1) Before all, a criterion concordance index, noted $C_j(e_1,e_2)$ is calculated for each pair of XML elements according to each criterion. It indicates the importance of “e_1 outrank e_2” according to the criterion C_j. It is expressed by:

$$Cj(e1,e2)\begin{cases} 0 & \text{if} \quad Cj(e1) < Cj(e2) - pj \\ 1 & \text{if} \quad Cj(e1) \geq Cj(e2) - pj \\ \in \,]0,1[& else \end{cases} \tag{5}$$

2) Then, a global concordance index, noted $C(e_1,e_2)$, indicating the importance of “e_1 *outrank* e_2” according to all criteria, is calculated for each pair of XML elements. $C(e_1, e_2) \in [0, 1]$ is such as:

$$C(e1,e2) = \sum_{j=1}^{p} wj \times Cj(e1,e2) \tag{6}$$

Where w_j is the wheight of the criterion C_j.

3) In parallel, a discordance index is calculated for each pair of XML elements according to each relevance criterion with a not null veto threshold. Two XML elements are more discordant since the discordant index is strong. It is noted $D_j(e_1,e_2)$, for each criterion C_j, $D_j(e_1,e_2)$ is such as:

$$Dj(e1,e2)\begin{cases} 0 \quad \text{if} \quad Cj(e1) - Cj(e2) \leq pj \\ 1 \quad \text{if} \quad vj \leq Cj(e1) - Cj(e2) \\ \in \,]0,1[\; \text{else} \end{cases} \tag{7}$$

4) Finally, the credibility degree $\delta(e_1, e_2) \in [0,1]$ which express in which measure “e_1 *outrank* e_2” is credible considering concordance indices and discordance indices is calculated for each pair of XML elements. It is the global concordance index diminished by the discordance force.

$$dj(e1,e2)\begin{cases} C(e1,e2) & \text{if } \forall j, Dj(e1,e2) \leq C(e1,e2) \\ C(e1,e2).\prod_{j \in F} \dfrac{1 - Dj(e1,e2)}{1 - C(e1,e2)} & \text{else} \\ \text{with } F = \{ j / Dj(e1,e2) > C(e1,e2) \} \end{cases} \tag{8}$$

The outranking relation 'S' is defined by the credibility degree as:

$$S(e1,e2) = \delta(e1,e2) \tag{9}$$

4.3. Exploiting the outranking ralation

Defined outranking relation allows us given two XML elements e1 and e2, knowing which element is more relevant than the other. We should now deduct from all these binary comparisons a rank of all XML elements, from the most relevant one to the less relevant one. For this, ELECTRE III proposes a ranking algorithm that produces a partial order of all the actions. First, two total preorders are obtained by two distillation algorithms as follows:

1) *Descending distillation:* Allows having the first preorder by these steps:
   - a) Identify E^1 " the better" elements of E;
   - b) Place E^1 at the top of the ranking list;
   - c) Set $E= E/E^1$ $(E - E^1)$.
   - d) Repeat (a),(b),(c) until $E = \emptyset$.
2) *Ascending distillation*: Allows having the second preorder by these steps:
   - a) Identify E^1 " the less good" elements of E;
   - b) Place E^1 at the bottom of ranking list;
   - c) Set $E= E/E^1$ $(E - E^1)$.
   - d) Repeat (a),(b),(c) until $E = \emptyset$.

To define E^1, E^2, …,E^n, the following indices are calculated:

- $P(e) = \sum S_{e' \in E}(e, e')$ (The Power of *e*),
- $W(e) = \sum S_{e' \in E}(e',e)$ (The Weakness of *e*),
- $Q(e) = P(e) – W(e)$ (The Qualification of *e*).

At each iteration, XML elements with high qualification Q(e) are ranked up and removed from E in case of descending distillation. XML elements with low qualification are ranked down and removed from E in case of ascending distillation. The final partial preorder is given by the intersection of the two total preorders.

4.4. An illustration example

Let us consider an example with 03 XML elements (*e1,e2,e3*) and 03 relevance criteria (*C1,C2,C3*) with respectively indifference, preference and veto thresholds (*q, p, v*). w_i is the weight of the criterion C_i as shown in Table 1. and table 2.

	C1	**C2**	**C3**
e1	16	05	17
e2	20	10	07
e3	12	14	15

Table 1. Elements performance values

	C_1	C_2	C_3
q	1	2	1
p	3	3	3
v	6	6	3
w	0.4	0.3	0.3

Table 2. Criteria weights and thresholds

Before all, a concordance degree C_j (e_i,e_j), is calculated for each pair of XML elements according to each criterion C_j :

$C_j(e_i,e_j)$	C_1			C_2			C_3		
	e_1	***e_2***	***e_3***	***e_1***	***e_2***	***e_3***	***e_1***	***e_2***	***e_3***
e_1		0	1		0	0		1	0
e_2	1		1	1		0	0		0
e_3	0	0		1	1	0.5	1		

Table 3. Criteria concordance degrees

Then, a discordance degree $D_j(e_i,e_j)$ is calculated for each pair of XML elements according to each criterion C_j :

	C_1			C_2			C_3		
	e_1	***e_2***	***e_3***	***e_1***	***e_2***	***e_3***	***e_1***	***e_2***	***e_3***
e_1		0.33	0		0.66	1		0	0
e_2	0		0	0		0.33	1		1
e_3	0.33	1		0	0		0	0	

Table 4. Criteria discordance degrees

The global concordance degree $C(e_i,e_j)$, for each pair of XML elements, will be the following:

C(ei,ej)	**e1**	**e2**	**e3**
e1		0.333	0.7
e2	0.7		0.4
e3	0.45	0.6	

Table 5. Global concordance degrees

Finally, the credibility degree ä(ei,ej), for each pair of XML element will be the following:

ä(ei,ej)	**e1**	**e2**	**e3**
e1		0.136	0
e2	0		0
e3	0.45	0.6	

Table 6. Credibility degree

The two distillation procedures allow us to obtain the two pre-orders: (a) for descending distillation and (b) for the ascending distillation. The intersection of these two pre-orders will give the final order (c).

(a) : e3 →(e2,e1)

(b) : (e3,e2) --→ e1

(c) =(a)⋂(b):e3→ e2→ e1

5. An adaptation for focused retrieval

As it is presented, the proposed approach allows retrieving an ordered list of relevant XML elements. It does not take into account the overlap problem in focused XML retrieval. The list of retrieved XML elements may contain nested elements and therefore redundant information (an XML element with its descendants, for example). To address this problem and retrieve a results list without redundancy (Focused task of INEX[1]), the two distillation procedures of ELECTRE III ranking algorithm are modified as follow:

1) *Descending distillation*: To have the first preorder :
   a) Identify E^1 " the better" elements of E;
   b) Place E^1 at the top of the ranking list;
   c) Set $E^1_{overlap}$ = the set of all descendants and ascendants elements of E^1;
   d) Set E = $(E - E^1 - E^1_{overlap})$.
   e) Repeat (a),(b), (c) and (d) until E = ∅.
2) *Ascending distillation*: To have the second preorder:
   a) Identify E^1 " the less good" elements of E;
   b) Place E^1 at the bottom of ranking list;
   c) Set $E^1_{overlap}$ = the set of descendants and ascendants elements of E^1 already ranked in the list.
   d) Remove $E^1_{overlap}$, from already ranked elements
   e) Set E= $(E - E^1)$.
   f) Repeat (a),(b),(c), (d) and (e) until E = ∅.

6. Conclusion

We presented in this paper an approach for content-oriented XML retrieval. The approach is not base on relevance estimation using score function as it is the case in the majority of XML information retrieval approaches, but based on a relational aggregation of relevance sources. The aggregation of relevance criteria is done not at relevance source value performance but at relevance source preference relations level. This way to aggregate allows considering several heterogeneous relevance criteria and can minimize compensation effect between relevance criteria. Notions, methods and ideas of relational aggregation paradigm where used to elaborate the approach.

The application of the approach in practice requires the initialization of all parameters: the weights of criteria,different thresholds of preference, indifference and veto. This will require a lot of experiments using official test collections (INEX collections). To give weights to the criteria, we can for example compare the results obtained by each criterion considered separately. Currently we are working on these aspects. We will present the results of these experiments in the future. Our aim in the future is also to integrate user criteria to have a personalized version of the approach.

[1] http://inex.is.informatik.uni-duisburg.de/

References

[1] Abbaci, F., Francq, P. (2008). XML Components Ranking: Which Relevant Ranking Criteria? Which Relevant Criteria Merging?, ICADIWT'08, *In*: First IEEE International Conference on the Applications of Digital Information and Web Technologies. Ostrava, Czech Republic.

[2] Ashoori, E., Lamas, M., Tsikrika, T. (2007). Examining Topic Shifts in Content-Oriented XML Retrieval, *In*: International Journal on Digital Libraries, 8(1) 39—60.

[3] Ben, Aouicha., Tmar, M. Abid, M., Boughanem, M. (2009). XML information retrieval based on tree matching, *In*: IEEE International Conference on Engineering of Computer Based Systems(ECBS 2008),Belfast, Ireland, 31/ 03/2008-04/04/2009, Roy Sterritt, David Bustard (Eds.), IEEE Computer Society. 499-500.

[4] Charlotte, M., Michel, L. (2005). La méthode multicritère ELECTRE III : Définitions, principe et exemple d'application à la gestion des eaux pluviales en milieu urbain, Bulletin des laboratoires des ponts et chausses, 29- 46.

[5] Chiaramella, A.Y.,Mulhem, P., Fourel, F. (1996). A model for multimedia information retrieval Technical report, Technical report, FERMI ESPRIT BRA 8134, University of Glasgow.

[6] Clarke, C. L. Comarck., A G. V. Thdhope, E. A. (2000). Relevance ranking for to three term queries, Information processing and Management, 36. 291-311

[7] Farah, M., Vanderpooten, D. (2007). An Outranking Approach for Rank Aggregation in Information Retrieval, *In*: Proceedings of the 30th Annual International ACM SIGIR Conference (SIGIR 2007), Amsterdam, the Netherlands, ACM Press (2007) 591 598.

[8] Farah,] M. (2006). Approche multicritère des problèmes de recherche documentaire. Phd thesis, Université Paris- Dauphine.

[9] Geva, S. (2004). GPX-gardens point xml information retrieval at inex 2004, *In*: INEX 2004 Workshop Proceedings,. Dagsthul, Germany, December.211-223.

[10] Grabs, T., Scheck, H.-J. (2002).Flexible information retrieval from XML with PowerDB XML, *In*: Proceedings in the First Annual Workshop for the Evaluation of XML Retrieval (INEX), 26–32.

[11] Guo, L., Shao, Botev, F. C., Shanmugasundaram, J. (2003). XRANK: Ranked keyword search over XML documents, *In*: International Conference on Management of Data.

[12] Kamps, J., de Rijke, M., Sigurbjörnsson, B. (2005). The Importance of Length Normalization for XML Retrieval, *In*: *Information Retrieval journal*, 8 (40. 631-654.

[13] Kazai, G., Lalmas, M. (2005). Inex 2005 evaluation metrics, *In*: INEX 2005 Workshop Pre-Proceedings,. Germany, November 401,406.

[14]. Lalmas, M., Trotman, A. (2009). XML retrieval, Encyclopedia of Database Systems, O.M. Tamer and L. Ling (Eds.), Springer.

[15] Ray, R. , Larson (2005). A Fusion Approach to XML Structured Document Retrieval, *Journal of Information Retrieval*, 8. 601 – 629.

[16] Lu, W.,Robertson, S.E., MacFarlane, A. (2006). Field-weighted XML retrieval based on BM25, *In*: Fuhr N., Lalmas M., Malik, S., Kazai G. editors. Advances in XML Information Retrieval and Evaluation: Fourth Workshop of the INitiative for the Evaluation of XML Retrieval (INEX 2005), V 3977 of Lecture Notes in Computer Science. Springer-Verlag. 161-171.

[17] Mass, Y., Mandelbrod, M. (2003). Retrieving the most relevant XML Component, *In*: Proceedings of the Second Workshop of the Initiative for The Evaluation of XM Retrieval (INEX), Schloss Dagstuhl, Germany, December (2003) 53-58. 15-17

[18] Piwowarski, B., Gallinari, G (2005). A bayesian network for XML Information Retrieval: Searching and Learning with the INEX Collection, *Information Retrieval journal*, 8 (4). 655-681.

[19] Roy, B., Bouyssou, D. (1993). Méthodologie multicritère d'aide à la décision : méthodes et cas. Economica.

[20] Roy, B. (1985). Méthodologie multicritère d'aide à la décision. Economica,.

[21] Roy,B. Critères multiples et modélisation des préférences : l'apport des relations de surclassement. Revue d'Economie Politique. 1974

[22] Sauvagnat, K. (2005). Modèle flexible pour la Recherche d'Information dans des corpus de documents semi-structurés. PhD thesis : université Paul Sabatier Toulouse.

[23]Sauvagnat, K., Boughanem, M (2005). A la recherche de noeuds informatifs dans des corpus de documents XML: où pourquoi on a toujours besoin de plus petit que soi.... ," *In*: Conférence francophone en Recherche d'Information et Applications (CORIA 2005), Grenoble, 09/03/2005-11/03/2005, IMAG, (2005) 119-134.

[24] Shanian, A., Milani, A.S.. Carson, C and Abeyaratne, R. C. (2008). A new application of ELECTRE III and revised Simos'procedure for group material selection under weighting uncertainty, Knowledge.-Based Syst.21,7 (2008), 709-720.

[25] Sigurbj Äornsson, B.,Kamps,.J., M. de Rijke, (2003). An element-based approach to XML retrieval. *In*: Proceedings of INEX 2003 workshop, Dagstuhl, Germany.

[26] Hearst, M. A. (1994). Multi-paragraph segmentation of expository text, *In*: Proceedings of the 32nd Association for Computational Linguistics, p. 9–16.