A probabilistic model to exploit user expectations in XML information retrieval

Fouad Dahak, Mohand Boughanem, Amar Balla

A probabilistic model to exploit user expectations in XML information retrieval

Fouad Dahak[a,*], Mohand Boughanem[b], Amar Balla[a]

[a] National Computer Sciences Engineering School (ESI), BP 68M Oued Smar, 16270, Algiers Algeria
[b] IRIT, University of Paul Sabatier, 118 Route de Narbonne, 31062 Toulouse, France

A B S T R A C T

The main objective of this paper is to exploit a new source of evidence derived from the document hierarchical structure for XML information retrieval. We consider that the structure of XML document is an important source of prior knowledge, and the structural features of an element may influence the user to consider that element as relevant. We build a probabilistic model to estimate the probability that the structural characteristics of an element attract user to explore the content of this element and consider it as relevant. This probability reflects the context importance. We propose a simple, well-motivated probabilistic model to estimate the context importance. Finally, we demonstrate the effectiveness of the context importance through comprehensive experimental studies carried out on IEEE XML document collection. Experimental results show that the proposed approach outperforms models exploiting other sources of evidence.

Keywords:
Priors
Element importance
User browsing map
Language model

1. Introduction

XML (eXtensible Markup Language) is a well-known standard for data representation and exchange on the Internet. XML document contains textual information as well as logical structures that highlight the underlying semantic. The main challenge for content-oriented XML retrieval is to select highly relevant elements that would satisfy the user information needs (Lalmas, 2009). These elements do, not only must be relevant, but they must also be at the right level of granularity. To address this issue, information retrieval models leverage query-dependent features related to element characteristics, such as query-term frequency within the element content. They also exploit query-independent features, called priors, such as the Page rank (Page, Brin, Motwani, & Winograd, 1999), document length (Kraaij, Westerveld, & Hiemstra, 2002; Miller, Leek, & Schwartz, 1998), and clickthrough data (Bao et al., 2007; Joachims, 2002; Kirsch, Gnasa, & Cremers, 2006) to enhance the retrieval performance. In XML content-oriented information retrieval, element characteristics including element length (number of terms in the element content) (Kamps, Rijke, & Sigurbjörnsson, 2004), element label frequency in the collection (Ashoori, Lalmas, & Tsikrika, 2007; Ogilvie and Callan, n.d.), element path length (Huang, Watt, Harper, & Clark, 2006) and node position in the document (Huang, 2007) are also exploited.

In this paper, we present a probabilistic model that exploit user expectations to enhance XML retrieval performance. Our objective is to estimate the context importance. To fulfill this objective, we first define the notion of element type and its structural context. Then, we propose a probabilistic model to estimate the structural context importance of an element that we use as prior in a language model approach for XML retrieval.

* Corresponding author.
E-mail addresses: f_dahak@esi.dz (F. Dahak), bougha@irit.fr (M. Boughanem), a_balla@esi.dz (A. Balla).

To achieve our objectives, we make the following assumptions:

- As for web information retrieval where exploiting hyperlinks structure of the web significantly improves retrieval effectiveness (Kamps, Kaptein, & Koolen, 2010; Page et al., 1999; Westerveld, Kraaij, & Hiemstra, 2001), the hierarchical structure of XML document would be an important source of prior knowledge in XML information retrieval. We believe that structural characteristics of an element, such as its position in the document and its surrounding context, influence the user during browsing a document. Therefore, we are interested in estimating the probability that a user focuses on a particular element during browsing a document, by exploiting the document hierarchical structure. This probability, named element importance, may reflect the user expectations about where to find relevant information.
- The document elements are classified into several types according to their labels (tags) and their hierarchical level in the document tree. Each element type belongs to a structural context defined by its surrounding element types. This structural context may influence the user to focus on a given element at a given level in the document tree.

We experiment context importance prior model (CPrior) on IEEE XML document collection of INEX and compare our results with length prior model and some models exploiting other sources of evidence.

The remainder of this paper is organized as follows: in Section 2, we review some related work focusing on the different sources of evidence used as priors in XML information retrieval. Section 3 describes our probabilistic model with an approach for estimating the element context importance. Finally, we present the results of the experiments in Section 4, and conclude this work and list some perspectives in Section 5.

2. Related work

Several query-independent features have been successfully used into information retrieval models in order to enhance the retrieval effectiveness. More particularly, features such as the number of incoming links to a document (Kamps et al., 2010; Kraaij et al., 2002; Westerveld et al., 2001), the page-rank (Page et al., 1999), the type of documents associated URL (Kraaij et al., 2002; Westerveld et al., 2001), the document publication time (Peetz & Rijke, 2013) and, the anchor text of outlinks (Kamps et al., 2010) are extensively exploited in web information retrieval. Earlier works such as Hiemstra et al. (2002) have shown that de document length significantly improves the retrieval effectiveness in document information retrieval. In the same line, Miller et al. (1998) have combined the document length with the average word length.

Huurdeman, Kamps, Koolen, and Wees (2012) have exploited the number of reviews, the rating average and the user tag frequencies in the social information retrieval. Damak, Pinel-Sauvagnat, Boughanem, and Cabanac (2013) have introduced the language quality and Badache, & Boughanem, (2015) have exploited social signals such as the number of user likes and shares.

These last years, several studies such as in Beckers and Korbar, (2010); Jay, Stevens, Glencross, Chalmers, and Yang, (2007); Tran and Fuhr, (2012); Velásquez, (2013) have exploited the eye-tracking technique in order to determine the user reading process on the Internet by understanding how people look at web pages. The user reading process may reveal something about the salience and the importance of elements in the document. Since it consolidates our intuition, the study carried out in Buscher, Cutrell, and Morris, (2009) seems to be the most interesting for our work. The authors have mapped the gaze data, which reflects the user attention, to DOM (Document Object Model) elements to build up a salient map of important elements. Their objective is to create a model based on the DOM of Web pages that can predict the user attention on single elements on a page. According to this study, we can deduce that a user explores a document by following a top-down method going from the general aspects towards the detail. Initially, the user is attracted by generic elements, which are at the top of the tree structure. Then he goes in-depth of more specific elements. Buscher et al. (2009) clearly demonstrate that the first look to a document expresses user expectations about where to find relevant information.

The structural information of XML elements was earlier integrated by Guo, Shao, Botev, and Shanmugasundaram, (2003) in the XRANK retrieval process by considering two-dimensional proximity metric involving both the keyword distance and the element ancestor distance. The element length (number of tokens in the element textual content) seems to be the most used as source of evidence (Banerjee & Han, 2009; Blanco & Barreiro, 2008; Ganguly et al., 2010; Kamps et al., 2004; Lalmas, 2009; Ogilvie & Callan, 2004, 2005; Pehcevski, Thom, & Tahaghoghi, 2005; Sigurbjörnsson, 2006; Sigurbjörnsson, Kamps, & Rijke, 2004). Element length prior influences the relative ranking by favoring longest elements. However, it was clearly showed by Kamps et al., (2004) that the effectiveness of the length prior should not be interpreted as a general claim that long elements are inherently more relevant than short ones. Ashoori et al., (2007) explore another source of evidence, namely the number of topic shifts in the node content. The idea is, since the exhaustivity[1] and the specificity[2] are both expressed in terms of the "quantity" of topics discussed within each element, the number of topic shifts within an element reflects its relevance. This approach tends to favor elements, which cover several topics. However, two nodes having different structural characteristics and different content can easily have the same number of topic shifts. Mihajlovic et al., (2005) exploit information about relevant elements issued from user's relevance judgments to update the priors, in order to discover the characteristics of relevant elements and update the priors in such a way that elements with similar

[1] The exhaustivity measures how exhaustively an element discusses the topic of the user query. (Lalmas, 2009).
[2] The specificity measures the extent to which an element focuses on the topic of query. (Lalmas, 2009).

characteristics are favored. However, this kind of information depends on the query and cannot be actually considered as priors. Two other properties are used by Huang et al., (2006); Huang, (2007), namely element position in the document and its path length. The idea behind is that relevant elements tend to appear in the beginning of the document and are not likely to be nested in depth. This approach favors the nodes with shortest path and closest to the document root. As topic shifts, elements can be found with the same location at the same level in the document tree, but completely different in structure and content. The element types are used in Termehchy and Winslett, (2011) to measure the strength of the relationships in a candidate answer and rank the candidate answers according to their strengths. The intuition is that the closer the association between types is in a subtree, the more the subtree represents a meaningful and coherent object.

Among the works exploiting priors, two works have particularly attracted our attention: Those presented by Beigbeder, Géry, Largeron, and Seck, (2010); Géry and Largeron, (2012) and Arvola, Kekäläinen, and Junkkari, (2011) The work described in Beigbeder et al., (2010) is the closest one to our work. Indeed, authors propose an extension of BM25 (Robertson, Zaragoza, & Taylor, 2004), by introducing structural features in the relevance estimation formula. The authors assume that the capacity of a tag to highlight relevant terms is intrinsic to the tag itself and is therefore not dependent to the content terms. The objective then is to evaluate whether a word featuring in a title is more important than a word taken from a section/paragraph, regardless of the word itself. A weight is computed for each tag and estimates the probability that the tag marks a relevant term. We share almost the same intuition; however, there are two major differences with our approach: first, the way Géry and Largeron (2012) estimate the tag weights depends on the element content, while in our approach we do it independently of the content, we assume that the structural characteristics of an element can give better results. Second, the tag is considered once for the entire collection, while we distinguish the different levels in the document structure. For us, a paragraph in the document summary should not have the same influence that a paragraph at a section or a conclusion levels.

The second work (Arvola et al., 2011) develops contextualization models. The main idea is to explore the effect of different contextualization models on different hierarchical levels. The authors hypothesize that the retrieval of short and focused elements would benefit from structural contextualization more than the retrieval of broader ones. We are particularly interested by the different ways of considering the structural context of an element and its influence on the retrieval.

Summary, we reviewed in this section various sources of evidence. We note that they are often used to favor specific type of elements such as long elements or elements with more topic shifts. An important source of prior knowledge in XML information retrieval is the hierarchical document structure. The document structure does not only organize the document content but also plays a considerable role during the browsing of the document by user. The structure characteristics of an element such as its position in the document and its surrounding context draw the attention of the users. By understanding the probability that a user chooses such element at such a level in the document structure, we can find a new source of evidence that allows benefiting from the document hierarchical structure in order to determine, which type of elements would be useful to favor during the retrieval.

3. Context importance

We propose a novel query-independent feature for XML information retrieval. We use these characteristics as source of evidence in order to quantify the element importance that reflects user expectations about where to find relevant information. First, we present the document model, then we define the structural context and finally, we present our approach of estimating the context importance.

3.1. Structural context modeling

In this section, we present our contextualization approach, which consists of a document representation and a context model.

3.1.1. Document model

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format, which is both human-readable and machine-readable. It is defined by the W3C's XML 1.0 specification. XML documents have a hierarchical structure and can conceptually be interpreted as a tree structure, called an XML tree. XML documents must contain a root element (one that is the parent of all other elements). All elements in an XML document can contain sub elements, text and attributes. The tree represented by an XML document starts at the root element and branches to the lowest level of elements.

An example of an XML document is given in Fig. 1 below:

An XML document model can represent both tree-shaped and graph structured data. In this paper, we consider only tree-shaped XML files. We model the document structure as a tree, where each node represents an XML element identified by its XPath (path-based language for finding information in an XML document). Fig. 2 presents the document tree of the XML document shown in Fig. 1 without text nodes.

We use the following functional notations for relationships among elements:

Let be e and f two XML elements.

- *level(e)*: Level of the element e in the document tree. The document root level is 1.

```
<article>
  <title> IEEE ANNALS OF THE HISTORY OF COMPUTING</title>
  <author> David Alan</author>
  <abstract>
    <p> A history, however, looks at the deep trends of modern life and asks where they have been, where they are now, and where
    they are going</p>
    <section>
      <p>It is a discipline that looks to the future as much as it retells the story of the past. Those of us involved with the Annals believe
      that the stored program electronic computer helps us understand almost every major trend of civilization.</p>
      <p>The rise of corporations? Supported and driven by computers. The abstraction of knowledge? A key element of computer
      science.</p>
    </section>
  </abstract>
  <body>
    <section>
      <p>Some 25 years ago, 26 if we are to be precise, a small group of computer scientists decided that their discipline not only had a
      past, it had a history. A history is a very different thing from a past.</p>
      <p>In this issue you will find an essay by Bernie Galler, the founding editor of the Annals and a former president of the Association
      for Computing Machinery. Galler recounts how this journal appeared shortly after the completion of a key event in computer
      history, a trial that invalidated the patent on the ENIAC computer.</p>
      <p>Since the founding of the Annals, the physical nature of the stored program computer has changed considerably. We've gone
      from mainframes and minicomputers to desktop computers, personal digital assistants, handheld gaming computers, cell phones,
      embedded computers, and a plethora of other devices that use a processor, memory, and program.</p>
    </section>
    <section>
      <title> Annals</title>
      <p>When Galler and his colleagues founded the Annals, they were about 30 years distant from the original emergence of the
      stored-program computers. A few of the original pioneers were gone but many remained to contribute to this publication or be
      interviewed.</p>
      <subsec>
        <p>The second issue the development of computer networks is represented by Phil Frana's article "Before the Web There Was
        Gopher," which discusses the program Gopher. This article underscores an important lesson of modern technology and a
        problem faced by this journal.</p>
      </subsec>
      <subsec>
        <p>Like many who have served with the Annals, I have a personal connection to the history of computation. My father joined the
        UNIVAC Corporation in the mid-1950s and worked in the fabled "glider factory" in St. Paul, Minnesota. He later moved to
        Burroughs before finishing his career at Unisys.</p>
        <p>One of those board members, long-time contributor James Cortada, has completed this issue with an article that suggests
        where we should be going an essay that suggests some of the questions that the contributors to this journal might want to
        consider.</p>
      </subsec>
    </section>
  </body>
</article>
```

Fig. 1. Example of an XML document.

- *label(e):* Label (tag) of the element *e*.
- *parent(e):* Parent node of *e* in the document tree.
- *ancestors(e):* An ancestor of an element ***e*** is an element in the same document as *e* belonging to the hierarchical path of *e* in the document tree (its parent, grandparent, great grandparent etc.). The function ancestors(e) give all the ancestors of *e*.
- *siblings(e):* List of elements at the same level than *e*.
- *distance(e,f):* Represents the difference between the level of the element *e* and that of the element *f*. It is estimated as follows:

$$distance(e, f) = \begin{cases} level(f) - level(e) \ \ if \ \ e \in ancestors(f) \\ 0 \ \ Otherwise \end{cases} \quad (1)$$

3.1.2. Context model

The context of an element consists of elements, which have a relationship with the contextualized element at a given distance. To define a context of an element in the XML hierarchy we need to define two concepts: The structural relationship between elements and the distance, which refers to a structural remoteness between elements.

We first consider that elements sharing same characteristics such as label and hierarchical level belong to the same context. Therefore, we have to define class of elements and construct a context around it. Secondly, instead of using the

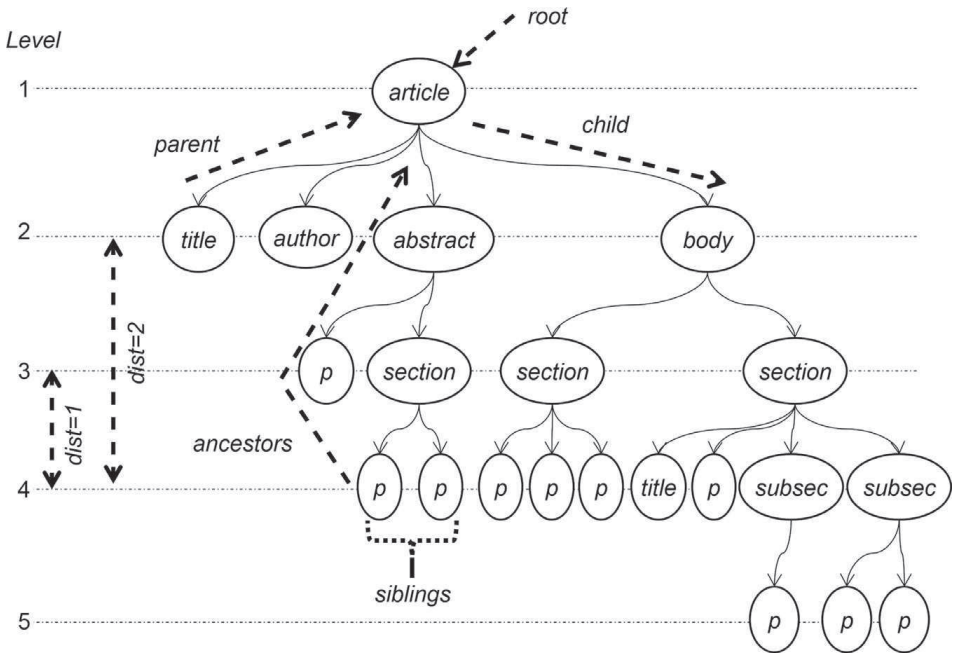

Fig. 2. XML document Tree.

distance between elements, we consider a probability between element classes. In the following, we define the element type, the relationship between the element types and then the context of an element type.

Definition 1 (Element type)**.** Element type ***T*** is a class of elements in a document having the same label ***l*** at a given level ***v*** and denoted as: $T\langle l, v\rangle = \{e | label(e) = l \ \wedge level(e) = v\}$.

The elements satisfying the element type definition are called instances of that element type. For example, the instances of the element type ***T*** **= <*p*,4>** in the XML document of Fig. 1 are those accessible by the following paths:

`/article[1]/abstract[1]/section[1]/p[1]; /article[1]/abstract[1]/section[1]/p[2]; /article[1]/body[1]/section[1]/p[1]; /article[1]/body[1]/section[1]/p[2]; /article[1]/body[1]/section[1]/p[3]; /article[1]/body[1]/section[2]/p[1].`

The element types represent classes of elements defined by a label appearing in a given level of document tree. As the elements of an XML document are linked by hierarchical relationship their respective element types should also be linked by an aggregation of hierarchical relationships. We thus define the relationship between two element types as follows.

Definition 2 (Hierarchical relationship between element types)**.** Given two element types *T* and *U*. Let *dist* be a positive integer. The hierarchical relationship between *T* and *U* at distance *dist* noted H_{dist}(T,U), determines the existence of elements ***e*** and ***f*** (respectively instances of *T* and *U*) which are at distance ***dist*** in the same document of the collection. Formally, H_{dist}(T,U) is defined as follows:

$$H_{dist}(T,U) = \begin{cases} 1 \ if \ \exists e \in T \ \wedge \exists f \in U \ where \, distance(e, f) = dist \\ 0 \ otherwise \end{cases} \quad (2)$$

The hierarchical relationship between element types represents the parent/child relationship when $dist = 1$, the grandparent/grandson relationship when $dist = 2$ and ancestor/descendant relationship when $dist > 2$.

The element types of a document can be represented as a directed graph where nodes represent the element types and oriented edges express the hierarchical relationships between these element types. Such a graph illustrates the possible relations between the different element types in an XML document.

Definition 3 (Element type graph)**.** Element type graph $G_d = (\mathbf{T}, E, dist)$ of document *d* is the directed graph. With $\mathbf{T} = \{T | T$ is an element type$\}$ is a set of nodes where each node represents an element type. E is a set of edges, which are ordered pairs of elements of T and represents the hierarchical relationship between element types. The edges are constructed as follow:

$$E = \{(T, U) | T \text{ and } U \text{ are Element Types } \wedge H_{dist}(T, U) = 1\}$$

Example. Let *d* be the XML document represented in Fig. 1. Fig. 3.a, b and c represents respectively the element type graph with $dist = 1$ (parent/child relationship), $dist = 2$ (grandfather/grandson relationship) and root/leaf relationship when *dist* is the leaf level−1.

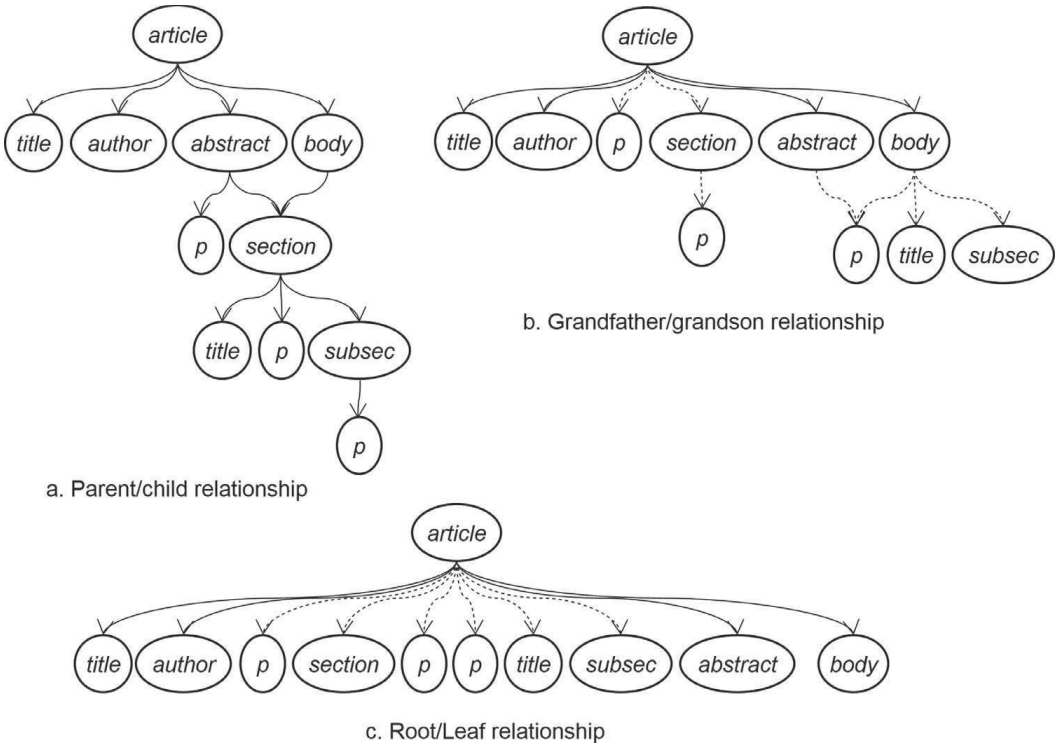

Fig. 3. Element type graph.

We note that the graph differs according to the considered distance. Some hierarchical relationships between element types disappear according to the considered distance.

We can now define the structural context of an element type (Definition 4) according to its position in the element type graph and the other nodes that are connected to it.

Definition 4 (Intrinsic structural context)**.** An intrinsic structural context of an element type ***T*** is defined by a set composed of ***T*** and its predecessors. The intrinsic structural context is denoted $C_T = \{U | U \text{ is an Element Type } \wedge (U \in R_T \vee U = T)\}$.

Where R_T is the set of all the predecessors of the element type ***T*** in the element type graph $G_d = (\mathbf{T}, E, dist)$ knowing that a predecessor of an element type in the element type graph is an element type having a hierarchical relationship with the considered element type: $R_T = \{U | U \text{ is an element Type } \wedge H_{dist}(U,T) = 1\}$.

The instances of the element type ***T*** are elements belonging to its context C_T.

The intrinsic structural context is defined over the element type graph. Thus, it depends on the distance considered in the hierarchical relationship between element types. Consequently, an element type may have different contexts according to the given hierarchical relationship.

For example, element type T = <p,4> in the element type graph of Fig. 3 has three structural contexts, shown in Fig. 4, according to the given hierarchical relationship.

The intrinsic structural context of the element type T = <p,4> is then for each considered distance as follows:

- Parent/child relationship: $C_T = \{<p,4>, <article,1>, <body,2>, <abstract,2>, <section,3>\}$
- Grandfather/Grandson relationship: $C_T = \{<p,4>, <article,1>, <body,2>, <abstract,2>\}$
- Root/Leaf relationship: $C_T = \{<p,4>, <article,1>\}$

3.1.3. Context importance

As it is mentioned, the user choice is guided by its expectations about elements which may contain relevant information. Assuming that user explores the document tree level by level and, at each level, he chooses between elements with different contexts. The choice between elements with the same contexts is done arbitrarily. We may quantify these expectations and consider it as context importance.

Definition 5 (Context importance)**.** Given two element types *T* and *U*. Let $H_{dist}(T,U)$ be the hierarchical relationship between *T* and *U* at a given distance *dist*. The context importance of the element type *U* compared to the element type *T* according to the hierarchical relationship H_{dist} denoted as $CI_{dist}(T,U)$, is the probability that a user explores an element *f* belonging to context C_U of element type U just after exploring an element *e* belonging to context C_T of element type T.

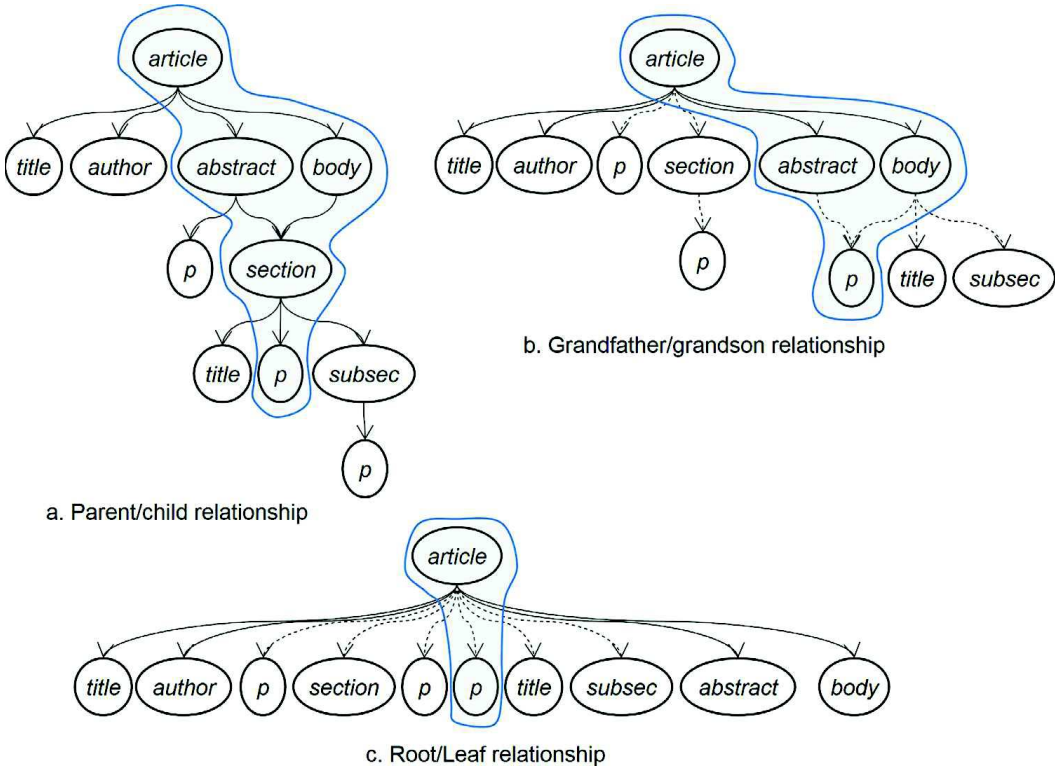

Fig. 4. Intrinsic structural context of an element type.

The context importance is defined as a transition weight from one element type to another. It can therefore, be used as a weighting of the element type graph edges and thus give a weighted graph. The resulted graph shows a map of the various possible paths that user can follow to get the desired information.

Definition 6 (User browsing map)**.** Given a document d, a user browsing map BMap(d) is the element type graph of d, where edges are weighted according to the context importance. $G_d^w = (E, V, d, w)$ where w: E$\rightarrow$[0,1] is a function that makes a mapping from directed edges to their context importance value.

Fig. 5 represents the user browsing map of the document tree shown in Fig. 2 with $dist = 1$.

The user-browsing map allows predicting the user browsing process of a given document. The edges represent the possible paths that user may follow to access an element type from another one. The context importance weighting each edge reflects the probability that a user chooses that target edge.

3.2. Context importance estimation

When user explores an element type in the user browsing map of a given document, the context importance represents the probability that he expects to find a given element type among the successors of the current element. This can be interpreted as a conditional probability and depends on the hierarchical relationships between element types in the user-browsing map. First, we define context importance by considering the parent/child relationship between element types $(dist = 1)$, this means that the user selects an element from the children of the element that he is exploring, and then we define the generalized formula.

3.2.1. Parent/child context importance estimation

Given two element types T, U and the parent/child hierarchical relationship $H_1(T,U)$. The context importance $CI_1(T,U)$ of U knowing T according to H_1 is estimated by a conditional probability as follows:

$$CI_1(T,U) = P(U|T) \tag{3}$$

CI_1(T,U) indicates the probability that a user selects an element f in the document tree belonging to the context C_U knowing that he is exploring an element e belonging to the context C_T.

According to Bayes formula, the probability of Formula (3) is calculated by:

$$P(U|T) = P(U).P(T|U)\ /P(T) \tag{4}$$

The precedence likelihood $P(T|U)$ expresses the probability that when user is exploring an element type U, he has just explored an element type T in the predecessors of U. Since an element type in a user browsing map may have several

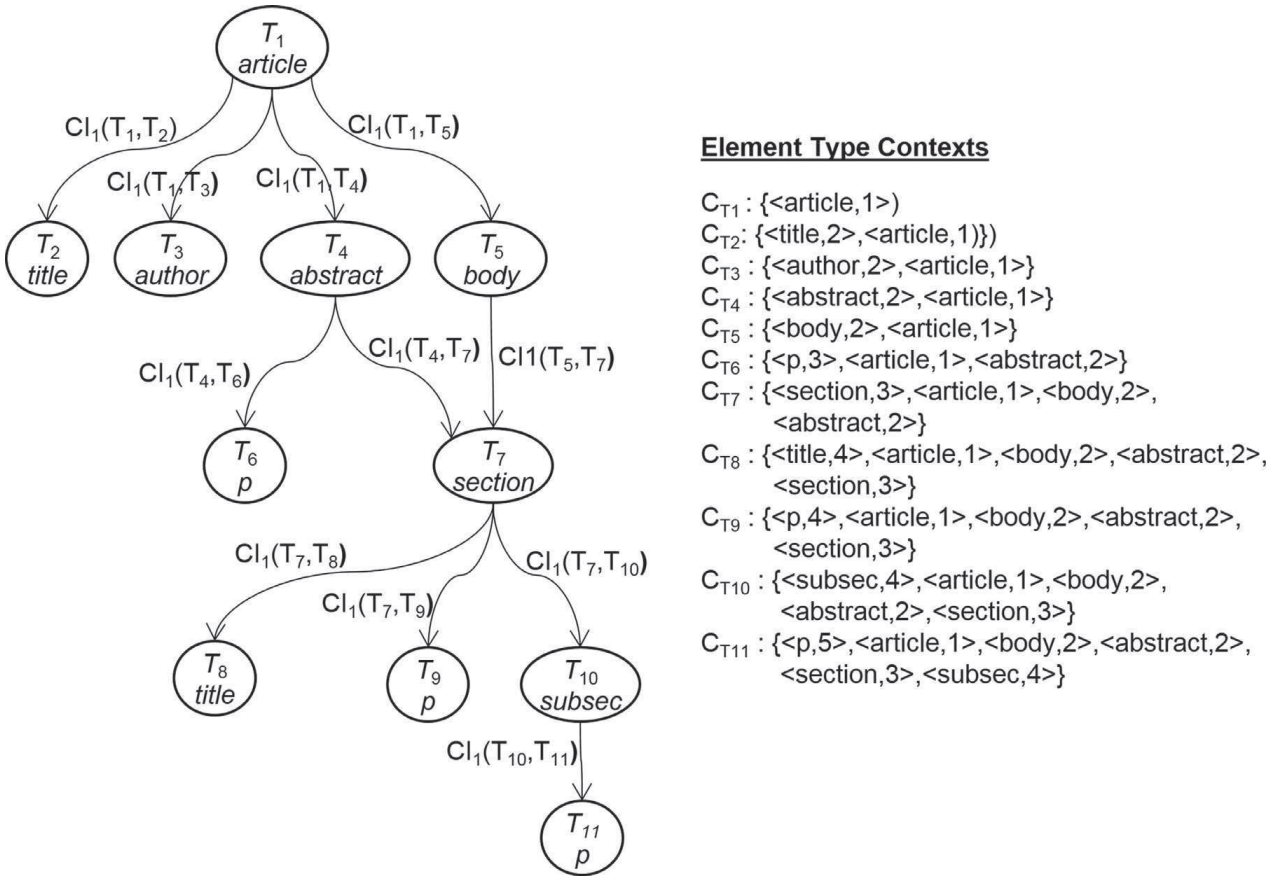

Fig. 5. User browsing map.

predecessors and each element type is represented just once. The precedence likelihood translates the probability that one of the predecessors of U is T. Thus, it is estimated as follows:

$$P(T|U) = \frac{1}{|R_U|} \tag{5}$$

Consequently, the context importance $CI_1(T, U)$ is calculated as follows:

$$CI_1(T,U) = P(U)/|R_U| * P(T) \tag{6}$$

3.2.2. Estimation of the context probability P(T)

Given an element type *T, P(T)* is the probability that an element e belonging to C_T is explored by the user. Since a user explores the document level by level, and chooses the element type to explore at each level, we estimate the probability *P(T)* by using the maximum likelihood at each level of the user browsing map.

To know the number of elements in a document belonging to a given context, we define the context cardinality as follows.

Definition 8 (Context cardinality)**.** Let C_T be the intrinsic structural context of element type T. The cardinality of C_T denoted $|C_T|_d$ is the number of elements in document d belonging to context C.

Probability *P(T)* is estimated by the following formula:

$$P(T) = \frac{|C_T|_d}{\sum_U |C_U \text{ with } U.level = T.level|_d} \tag{7}$$

where $|C_T|_d$ is the cardinality *of the context* C_T *and* $\sum_U |C_U \text{ with } U.level = T.level|$ *the sum of cardinalities of all the contexts at the same level as* C_T.

Two exceptional cases can arise in this estimation of the context importance. First, maximum likelihood will assign zero probability to contexts not occurring in a document. For example, let be $C_T = \{<section,3>,<article,1>,<body,2>\}$ an intrinsic structural context in the user browsing map of a document collection. If a context of a given element type $T = <section,3>$ does not have any occurrence in a given document d; its cardinality $|C_T|_d = 0$. Thus, the probability *P(T)* in the document is

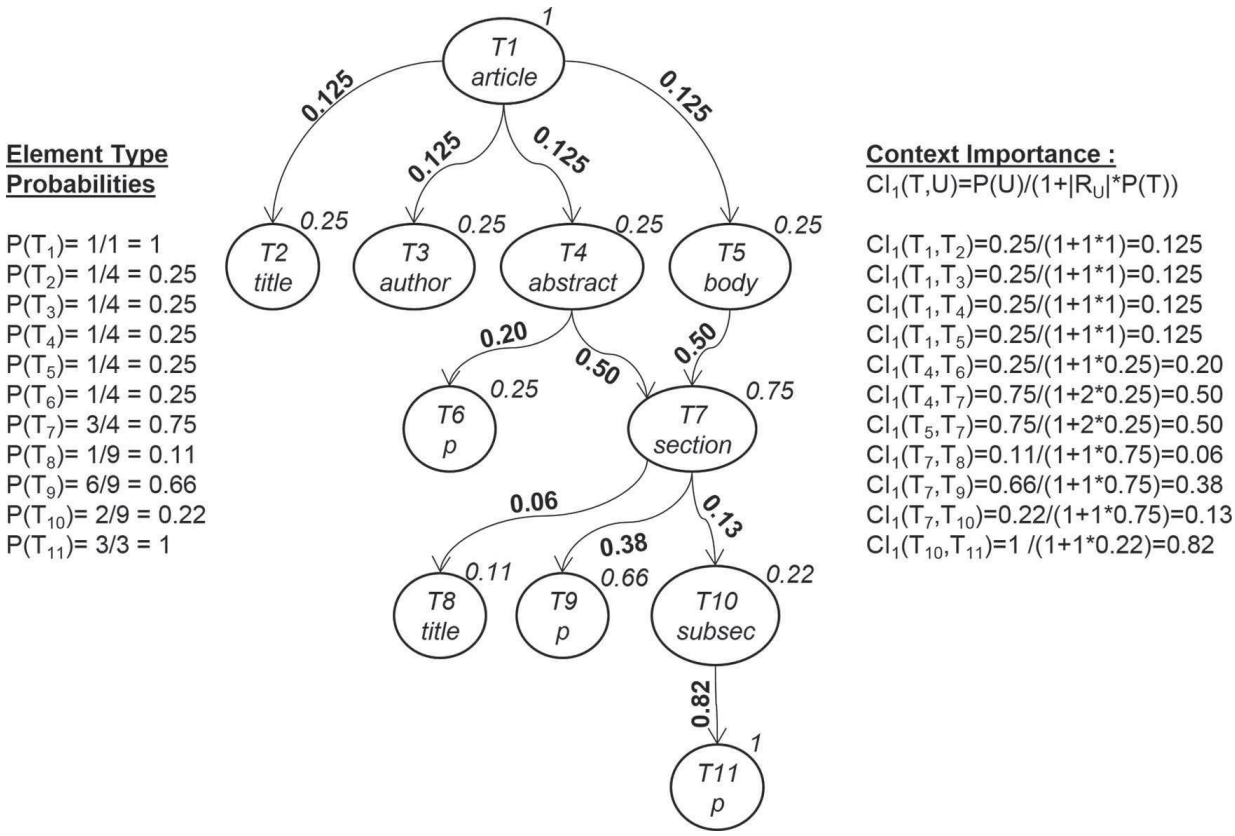

Fig. 6. Estimation of the context importance.

null. We use a smoothing method to avoid the null probability. The context importance of an element type is thus not only the fruit of the document structure but also of occurrences of that element type in all the collection.

Several classes of smoothing strategies have been proposed. (Zhai & Lafferty, 2004) studied three approaches to smoothing: Jelinek-Mercer smoothing, Dirichlet priors and absolute discounting, as well as the backoff versions of these methods. The effects of each of these smoothing mechanisms was examined on five different test collections. There was a clear ordering among the methods in terms of precision results; Dirichlet priors performed better than absolute discounting, which performed better than Jelinek-Mercer. According the (Zhai & Lafferty, 2004), we use Dirichlet smoothing technique to calculate *P(T)* as follows:

$$P(T) = |C_T|_d + \mu_s * \frac{|C_T|_c}{\sum_U |C_U \text{ with } U.level = T.level|_c} / \mu_s + \sum_U |C_U \text{ with } U.level = T.level|_d \quad (8)$$

Where c denotes the whole collection.

On the other hand, as we estimate the probability of an element type by using the maximum likelihood at every level of the document tree (Formula (7)), it is possible, when estimating the context importance in Formula (6) that P(U) is upper than P(T) because of the nature of the document structure. That can make CI_1(T,U) upper to 1. Therefore, to avoid a similar case, the Formula (6) becomes as follows:

$$CI_1(T,U) = P(U)/(1 + |R_U| * P(T)) \quad (9)$$

By applying the Formula (9) to the user-browsing map of Fig. 5 corresponding to the document of Fig. 1, we obtain the context importance illustrated in the Fig. 6 below. We annotate nodes with corresponding element type probability and edges with structural context importance.

3.2.3. Generalized context importance estimation

The generalization of context importance estimation allows considering any distance *dist*. It can be considered as a propagation of the context importance throughout the user-browsing map. This leads to consider two cases. First of all, we estimate the context importance propagation along a simple path then throughout the whole user browsing map.

3.2.3.1. Propagation of CI_{dist} along a simple path. Fig. 7 shows the importance of the element type <*p*,5> at different distances compared to its respective predecessors along a simple path.

According to distance *dist*, a user may access an element belonging to <*p*,5> from <*subsec*,3> if *dist* = 2, from <*section*,2> if *dist* = 3 *and* from <*article*,1> if *dist* = 4. Thus, the context importance of the element type <*p*,5> differs according to

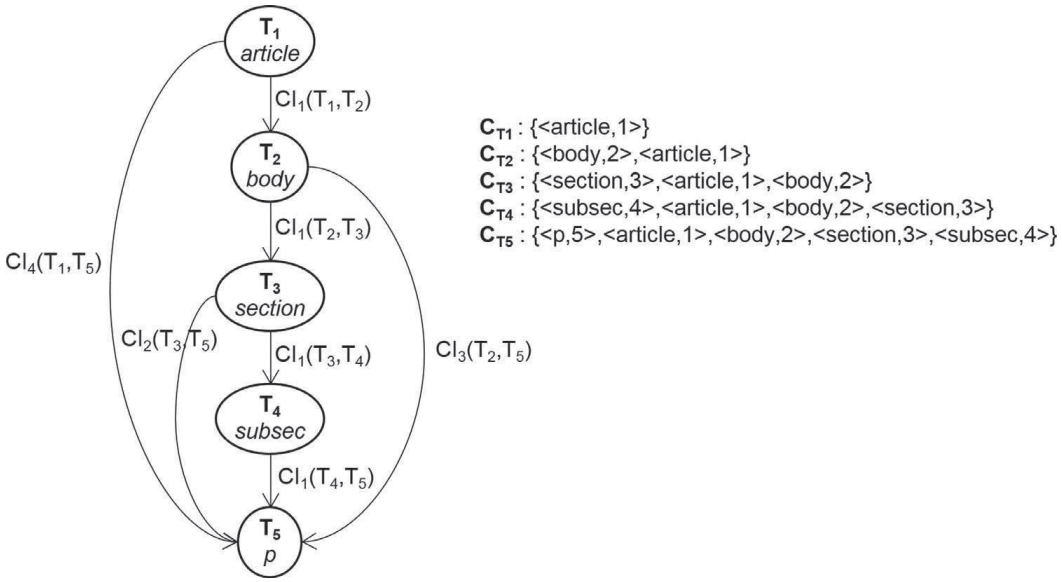

Fig. 7. Context importance along a simple path.

the given distance. In Formula (6), we defined it for distance $dist = 1$ (parent/child relationship). We assume a conditional independence over a different level of the simple path and see now how to estimate the context importance nor matter what the value of the distance *dist* is.

With distance $dist = 2$, the hierarchical relationship $H_2(T,U)$ between the element types T and U means that it exists two elements e and f in the document tree belonging respectively to the contexts C_T and C_U, where e is the grandfather of f. Thus, there exists an element g such as g is the child of e and the parent of f. According to the definition of the hierarchical relationship in Formula (2), and knowing that an element in the document tree has only one parent, this means that e cannot be the grandfather of f without the existence of an intermediate element g. Consequently, the relationship $H_2(T,U)$ cannot exists without the existence of the two relationships $H_1(T,V)$ and $H_1(V,U)$ at the same time where V is the element type of g.

On another side, the user choice of the next node to explore at any fixed time during the browsing process over the browsing map of a given XML document does not depend on the history of all the visited nodes but only on the current node. Therefore, we can consider the user browsing process as Markovian with respect to a filtration $\{F\}_t$. Thus, with the Markov property of the browsing process, the context importance at distance $dist = 2$ is estimated with the product of the two context importance at distance $dist = 1$ as follows:

$$CI_2(T,U) = CI_1(T,V) * CI_1(V,U) \tag{10}$$

By replacing the context importance in Formula (10) with their estimation in Formula (6), we obtain:

$$CI_2(T,U) = \frac{P(V)}{|R_V| * P(T)} * \frac{P(U)}{|R_U| * P(V)} = \frac{P(U)}{P(T) * |R_V| * |R_U|} \tag{11}$$

We note that, what is propagating along a simple path is just the precedence likelihood. This means that the intermediate element types between two element types along a simple path does not matter whatever the distance between T and U. In a general way, the context importance CI_{dist} along a simple path is calculated as follows:

$$CI_{dist}(T,U) = \frac{P(U)}{P(T)} * \prod_{V \in path(T,U)} \frac{1}{|R_V|} \tag{12}$$

Where path(T,U) is a function given a set of element types belonging to the path going from T to U in the user browsing map.

3.2.3.2. Estimating CI_{dist} in a user browsing map. Fig. 8 shows the context importance of the element type $T = <p,6>$ at different distances compared to its predecessors in a user browsing map. When distance *dist* is superior to 1, the same element type can be accessed from different paths.

With distance $dist = 2$, a user may access the element type $T_6 = <p,6>$ from two possible paths. He may go from $T_4 = <subsec,4>$ or from $T_5 = <citation,5>$. Thus, the probability of exploring an element of type U just after exploring an element of type $T_3 = <section,3>$ can be estimated along two simple paths. Consequently, user may choose the first path or the second one. Thus the probability of exploring an element of type U is the sum of the two probabilities, that of the path passing by V and that of the path passing by W. However, according to Formula (9), the intermediate element types

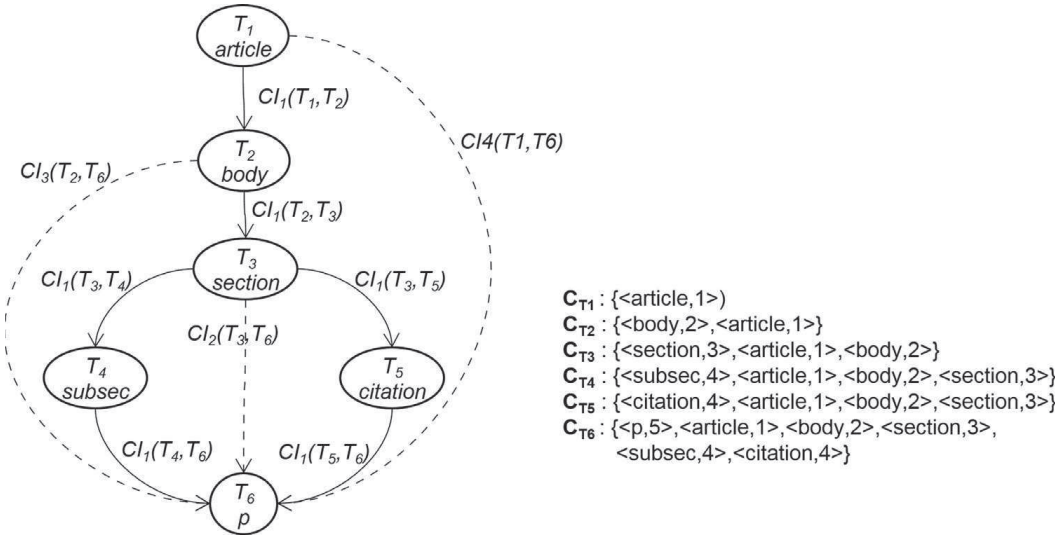

Fig. 8. Context importance on all the user browsing map.

between *U and T* does not matter:

$$CI_2(T,U) = \frac{P(U)}{2 * P(T)} + \frac{P(U)}{2 * P(T)} = \frac{P(U)}{P(T)} \tag{13}$$

In a general way, given two element types *T* and *U* the context importance of *T* compared to *U* at a given distance *dist* is estimated following the generalized formula as follows:

$$CI_{dist}(T,U) = \frac{P(U)}{P(T)} * \sum_{P \in paths(T,U)} \prod_{V \in P} \frac{1}{|R_V|} \tag{14}$$

where paths(T,U) is a set of all possible paths from T to U in the user browsing map.

To avoid the same exceptional case mentioned for the context importance estimation (see Formula (6)), Formula (14) is then estimated as follows:

$$CI_{dist}(T,U) = \frac{P(U)}{1 + P(T)} * \sum_{P \in paths(T,U)} \prod_{V \in P} \frac{1}{|R_V|} \tag{15}$$

Formula (15) represents the generalized formula of estimating the context importance of any element type in the user-browsing map whatever the distance *dist*.

4. Experiments and results

4.1. Methodology

In this section, we describe our methodology to evaluate intrinsic structural context importance in XML retrieval. Our aims are:

(a) Examining the characteristics of XML elements reflected by their context importance (Section 4.2);
(b) Comparing context importance with length prior which is the most used source of evidence in information retrieval (Section 4.3.3);
(c) Comparing our results with four different works using different sources of evidence (Section 4.4).

We conducted extensive experiments on INEX IEEE collection to investigate our three aims. Section 4.1.1 describes the IEEE collection used in our experiments, Section 4.1.2 presents the baseline retrieval model that we use, and Section 4.1.3 discusses our experimental settings.

4.1.1. Collection and metrics

INEX (The Initiative for the Evaluation of XML retrieval) provides a benchmark for the evaluation of XML information retrieval. This includes a document collection, topics, relevance assessments and metrics. There have been a number of changes in the document collection used over the years in the INEX experiments. Before 2006, the collection used was an IEEE XML document. Which consisted of 16,819 articles, marked-up in XML, from 24 magazines of the IEEE Computer Society's publications, covering the period of 1995–2004, and totaling 764 MB in size, and over 11 million in number of elements. On average, an article contains 1532 XML nodes, where the average depth of the node is 6.9. In 2007, INEX

introduced the Wikipedia XML document collection. The 2007 document collection was approximately 5.6 GB in size. On average, an article contains 161.35 XML nodes, where the average depth of an element is 6.72. In 2009, INEX provided a new Wikipedia collection, which is approximately 60 GB in size and contains 2.7 million articles with over 30,000 unique tags in it.

In order to measure the effectiveness of the element context importance we need deep documents and a great number of elements per document and per level so that our maximum likelihood based probability take its full meaning. In our experiments, we use the IEEE XML document collections version 1.8 and related topics with relevance assessments. In the INEX (IEEE) collection, the granularity levels are relatively easy to distinguish by providing reasonably clear and standard division of article-section-subsection-paragraph levels, similar to many other XML standards for structured text. On the other hand, the two smoothing parameters of our formula that we must set compel us to perform multiple tests and adjustment hence the need for a relative small collection.

Two types of queries are used in INEX: content only (CO), and content and structure (CAS). Queries of the first type are formed by simple terms without any information on the node structure. The second type of query specifies the desired content and the desired structure. In our studies, we focus on CO queries because we want to show the difference between the node types thorough their importance.

Until 2005, the relevance assessments were collected along two dimensions, specificity and exhaustivity. Since 2006, only specificity is considered. Exhaustivity, which is defined as the extent to which the document component (XML element) discusses the topic of request, is assumed a constant factor bearing no effect on the relevance score of an XML element. Specificity, which is defined as the extent to which a document component focuses on the topic of request, is calculated automatically as the ratio of the number of highlighted characters contained within the XML element and the length of the element (rsize/size). Specificity hence can take any value in [0,1]. Since exhaustivity is a constant, the relevance score of an XML element is a function of the specificity score only (Fuhr, Lalmas, & Trotman, 2007).

The INEX Benchmark propose 29 Content Only queries for IEEE collection. We used all of these queries in our tests.

The used metrics are XCG (eXtended Cumulated Gain) metrics, which are an extension of Cumulative Gain (CG) (Järvelin & Kekäläinen, 2002) which takes into account the dependencies between XML elements. Two metrics are included in the XCG metrics: nxCG (normalized extended Cumulated Gain) and ep/gr (effort-precision/gain-recall). In our experiments, we use nxCG at the cutoffs 10, 25 and 50. The value nxCG[i] represents the ratio between the gain cumulated by the user at rank i and the one he could cumulate if the system was optimal. We will also use the MAep measure, which is the average of the effort /precision obtained for each rank where a relevant element is returned.

4.1.2. Baseline retrieval model

In order to show the contribution of our approach we need to use a model that integrates element priors in the relevance estimation. We can then use different sources of evidence as priors and compare the results on the same platform in order to makes comparison more significant. The question that arises at this level is that, compared with the same user query, what are the query-independent characteristics of the element can improve the retrieval? Therefore, like in (Kaptein & Kamps, 2013; Sigurbjörnsson, 2006) our baseline retrieval model is a standard language model because it allows us to use query-independent characteristics like prior probability. The relevance of an element in a language model is computed as follow:

$$P(e|q) = P(e).P(q|e) \tag{16}$$

Where ***e*** is an element, ***q*** is a query considered as a sequence of terms t1, t2… tn. ***P(e)*** is the prior probability of element ***e*** and ***P(q|e)*** is the probability of generating query ***q*** from element ***e***. We consider a unigram language model where the terms composing the element content are produced randomly and independently from each other. It is therefore a multinomial distribution over terms $\boldsymbol{t_i}$ of the indexing vocabulary ***V*** with ***freq (t,q)*** the frequency of term ***t*** in the query ***q***. We assume that the content of an element ***e*** is obtained by gathering its own content with the content of all its descendants. The query generating likelihood by element ***e*** is obtained by the following formula:

$$P(q|e) = \prod_{t \in q} P(t|e)^{freq(t,q)} \tag{17}$$

where, the conditional probability P(t|e) represents the probability that term ***t*** occurs knowing that element's language model ***e*** occurred. It is calculated using the maximum likelihood with a Dirichlet smoothing following the formula:

$$P(t|e) = \frac{freq(t,e) + \mu_m * \frac{freq(t,C)}{|C|}}{\mu_m + |e|} \tag{18}$$

*where **freq (t,C)** is the frequency of term **t** in collection **C** and **|C|** is the sum of terms frequencies in the collection.*

4.1.3. Parameter settings

We used a Dirichlet smoothing at two levels. The first parameter μ_m in the query likelihood estimation (Formula (18)). The second parameter μ_s in the context probability estimation (Formula (8)). To determine the parameter values that give the optimal results we used cross-validation in a series of experiments. We have divided the 29 queries into two groups: Group A and Group B. A grid search (from 50 to 3000 by step of 50) is used to find the optimal parameter values for Group

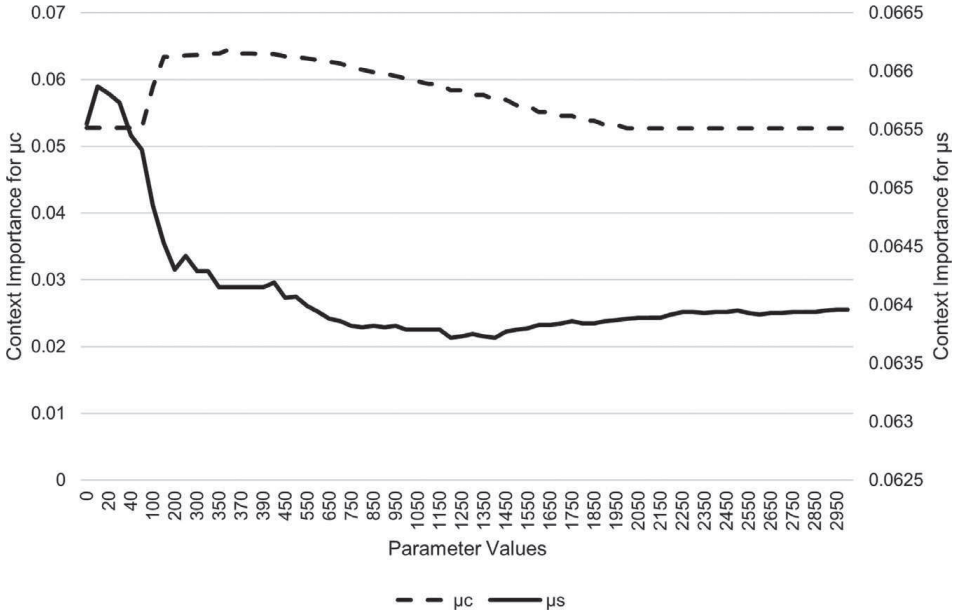

Fig. 9. Smoothing parameters estimation for MAep metric.

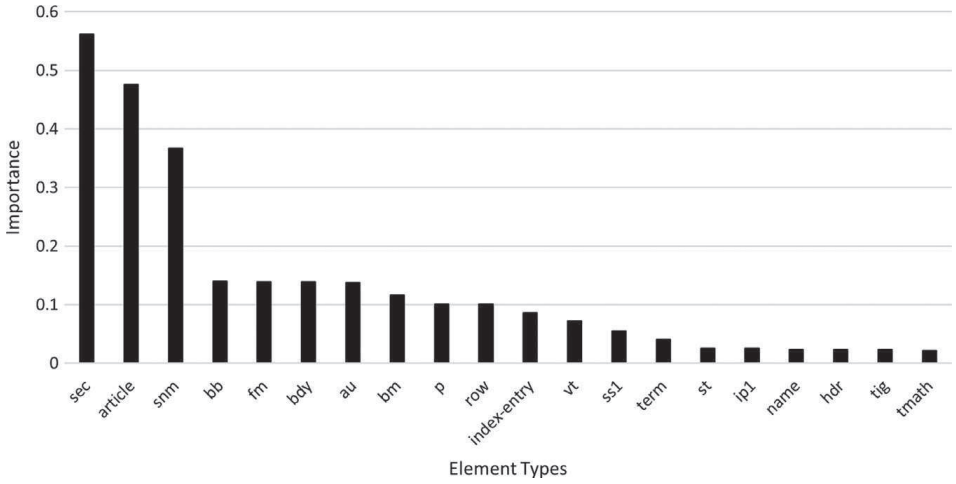

Fig. 10. Most important element labels in the collection.

A, and test on Group B, and vice versa. The results are shown in Fig. 9. The optimal values for the smoothing parameters are $\mu_m = 360$ and $\mu_s = 10$. We also noticed that the two smoothing parameters are completely independent.

4.2. Characteristics of context importance

This section discusses the results of the experiments we conducted to investigate the characteristics of XML elements reflected by their context importance. First, we examine the relation between the element type and their intrinsic structural context importance (Section 4.2.1). We discuss in Section 4.2.2 the relation between the element context importance and the elements length. We then examine whether the elements level in the document tree influence their context importance (Section 4.2.3).

4.2.1. Element type vs. context importance

This section discusses the relation between element types and their context importance. First, we present the most important element types in the collection according to their context importance. We then study the correlation between the element type frequency and the context importance.

Fig. 10 shows the list of the top 20 most important element types in the collection according to the average of their context importance considering the parent/child relationship whatever the context level.

We note that the element type ***sec*** (section) is the most important in this collection, followed by the documents root element ***article*** comes then ***snm*** (second name). Whereas element type ***ti*** (title) is placed in 28th position. For a collection

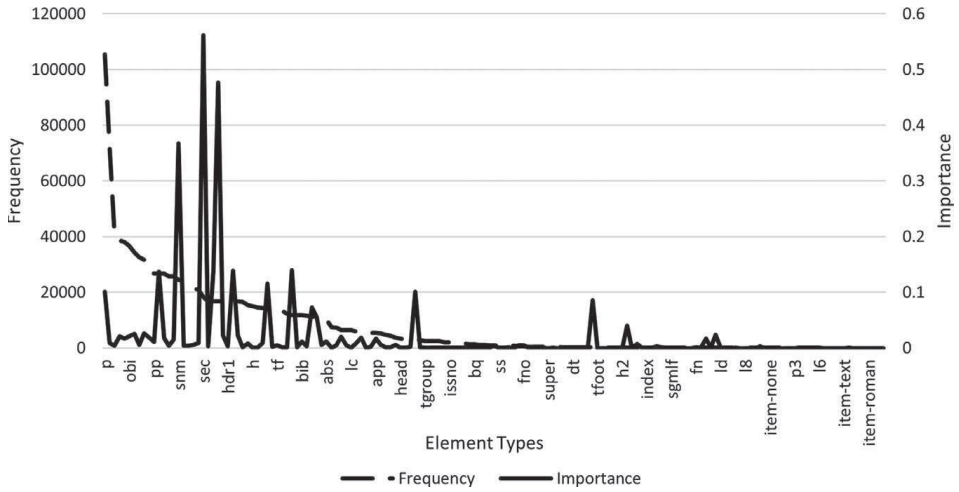

Fig. 11. Element type frequency vs. context importance.

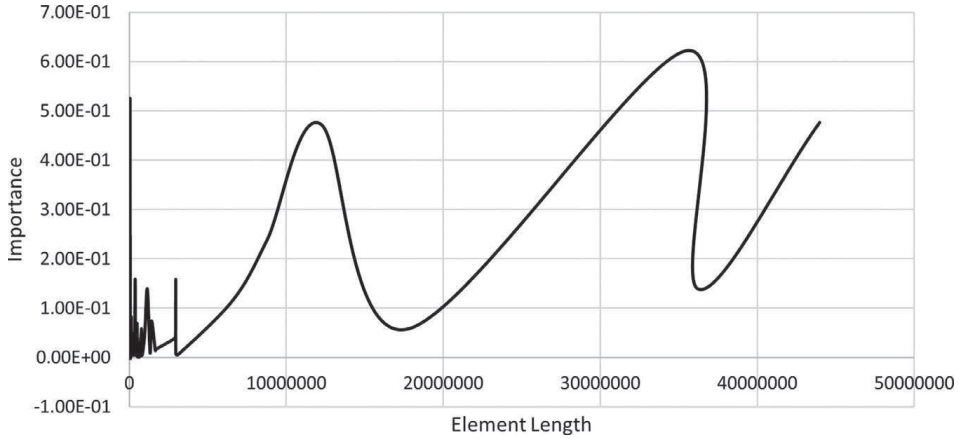

Fig. 12. Elements length vs. context importance.

containing scientific articles such as IEEE, sections are most carrying information. They are thus more frequent in the documents and especially compared to their hierarchical parent in the document tree structure. We also, notice that the element type paragraph (***p***) is at the 9th position when its parents (section, bb, bdy) are at the top of the list. That means that the context importance can make distinction between an element and its descendants or ancestors.

Fig. 11 presents context importance compared to element type frequency in the collection. We note that there is not a direct relation between the two features. However, it seems that the higher values of context importance are observed for the most frequent element types in the collection. However, there is not a proportional relationship between them. This can be explained by the distribution of the elements type over their hierarchical contexts. Thus, what makes an element type more important than another one is not only its frequency in the collection but also its frequency compared to its siblings.

Fig. 11 clearly shows that it is not the most frequent element type, which is the most important. Element type ***p*** (paragraph) for example is most frequent in the collection but it is not more important than ***section*** whose frequency is much smaller. The elements containing elements of type ***p*** also contain other elements of different type what is not the case for section. The structural context of element type *section* is thus more important than that of ***p*** even if this latter is most frequent.

4.2.2. Element length vs. context importance

Fig. 12 shows the relation between element length and context importance. We note that the context importance does not depend directly on element length. However, the length and the context importance evolve almost proportionally. The more the element is long the more it seems to be important. Indirectly, the context importance confirms that the longest elements are ready to be relevant and smallest one can be neglected. This also shows that the structure of a document is not present randomly but it brings a semantic to the textual content. Because the same elements considered as relevant according to their length are also relevant according to their context importance, which is obtained only according to the structural characteristics and without considering the element content.

This result can be used to improve retrieval effectiveness by removing for instance smaller elements according to their importance. It seems that greater is the element length more its context is important.

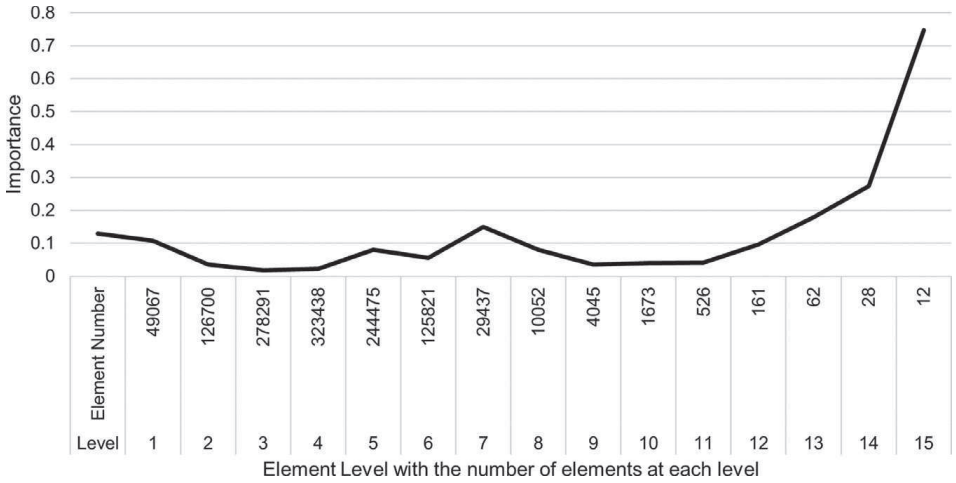

Fig. 13. Elements level vs. context importance.

Table 1
PCPrior vs. RCPrior results according to nxCG and ep-gr metrics.

	nxCG			MAep
	@10	@25	@50	
PCPrior	0.2678	**0.2437**	**0.2192**	0.06868
RCPrior	**0.2710**	0.2433	0.2164	**0.08040**

4.2.3. Element level vs. context importance

Fig. 13 shows the relation between the element types level and the average of context importance at each level of the collection tree structure.

Fig. 13 shows that the context importance is proportional to element level and inversely proportional to the number of elements per level. This means that the more the element is in-depth, more it is important, and the greater is the number of its siblings, less is its importance. The first observation can be explained by the fact that the more we go in-depth, the more we meet specific elements and more the number of the element types decrease. Moreover, less is the diversity in the element types on a given level greater is the importance. The second observation is directly related to the nature of *P(T)*, greater is the number of siblings less is the element context importance. We deduce that the most important elements in structured documents are those, which are at the top of the tree structure because they are more general, and those, which are most in-depth because they hold required information. The intermediate elements are thus less important.

4.3. Context importance as prior probability

In this section, we present the results obtained with context importance model CPrior used as prior probability in the baseline model by setting the smoothing parameters to optimal values that our experiments have shown. The results are presented according to the nxCG and MAep metrics at cutoffs 10, 25 and 50. First, we carried out our experiments on all the elements of the collection without any distinction. In second time, according to the characteristics studied in Section 4.2 about the relationship between context importance and element length, we improved our retrieval effectiveness by removing small elements.

4.3.1. Variants of the context importance prior model

According to the considered distance from the user-browsing map, the context importance can be estimated by comparison with the direct predecessors of the considered element type or with the root node. The first distance *(dist* = 1*)* gives a local estimation of the context importance when the second (*dist* = *T.level*−1 where *T.level* is the level of the element type *T*) gives a global estimation. Thus, according to these values we distinguish two models: The parent-child context prior (PCPrior) where the importance is estimated according to Formula (9) and the root context prior (RCPrior) where the importance is estimated according to Formula (15).

Table 1 shows the results obtained with these two models according to nxCG and MAep metrics at cutoffs 10, 25 and 50.

RCPrior model achieved significantly better results compared to PCPrior model (0.2710 vs. 0.2678) on the first 10% returned elements. What means that the relevant elements are returned earlier by RCPrior model. While these elements are returned only at the 25% of the retrieval result by PCPrior model. The MAep metric clearly shows that RCPrior gives better results than PCPrior (0.08040 vs. 0.06868) by improving the retrieval effectiveness of 17.06%.

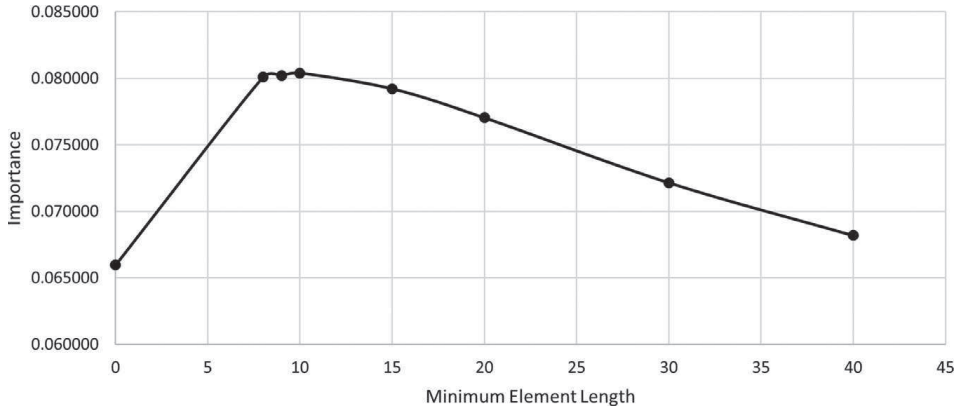

Fig. 14. CPrior improvement by removing small elements.

Table 2
CPrior vs. BaseLM and BaseLP according to nxCG and ep-gr metrics.

	nxCG[10]	nxCG[25]	MAep
CPrior	**0.2710**	**0.2433**	**0.08040**
Base $_{LM}$	0.1832	0.1921	0.06280
Diff(%)	***47.92***	***26.65***	***28.03***
Base$_{LP}$	0.2261	0.2199	0.06590
Diff(%)	***19.86***	***10.64***	***22***

The PCPrior model consider elements appearing in-depth with a reduced number of siblings as important even if they are not relevant. The local estimation of element importance can make so that an element is classified as more important than another one while both were estimated compared with their direct parents. Therefore, not estimating the importance of two elements according to the same context can biases the ranking.

On the other hand, the RCPrior model estimates the importance of all the elements according to the document root. Therefore, elements that are considered as important are important for the entire document not only for their local contexts. This means that relevant elements are more important according to the document root context than in their local context.

For the remains experiments, we retain the RCPrior model that we simply call CPrior.

4.3.2. Length improvement of CPrior

As shown in Section 4.2.2, the relationship between context importance and element length can be used to improve the retrieval effectiveness of CPrior model by removing small elements. We conducted some experiments to evaluate the MAep evolution according to the minimal element length to be considered. Fig. 14 shows that by removing elements under 10 keywords (element length is calculated after stemming and removal of stop words) the retrieval effectiveness is significantly improved by 21.87%.

4.3.3. Context prior vs. length prior

In this series of experiments, the results obtained by CPrior model are compared with the basic language model BaseLM (the baseline model without the prior probability) and length prior language model BaseLP presented in Ramírez, Westerveld, and Vries, 2005 where the prior probability is considered as an element length function.

Table 2 shows the results obtained with context importance model CPrior on IEEE document collection according to the nxCG and MAep metrics at cutoffs 10 and 25 compared with the baseline model BaseLM and length prior model BaseLP.

Compared with the Base$_{LM}$ model using no information as priors, CPrior presents a clear improvement of the retrieval effectiveness from the first returned elements: 47.92% at the first 10% returned elements and 26.65% at the first 25% returned elements. The MAep measure also shows that the structural importance used as priors improves the results by 28.03% what is a considerable improvement. This first comparison shows clearly that the structural importance can be used as priors to improve retrieval effectiveness and that elements considered as important seem to be the most relevant.

The CPrior model presents also a considerable improvement compared with the Base$_{LP}$ model (which uses the element length as priors) from the first returned elements. One notes an improvement of 19.86% at the first 10% returned elements and 10.64% at the first 25% returned elements. The improvement according to the MAep measure, which is 22%, is also considerable and this improvement is statistically significant according to *t*-test. We can deduce that the use of element content characteristics as priors improves the retrieval effectiveness but our structural context importance (which is exclusively estimated with element structural characteristics) improves it better. In addition, it shows that an element with a big importance has of big probability to be relevant whatever its content.

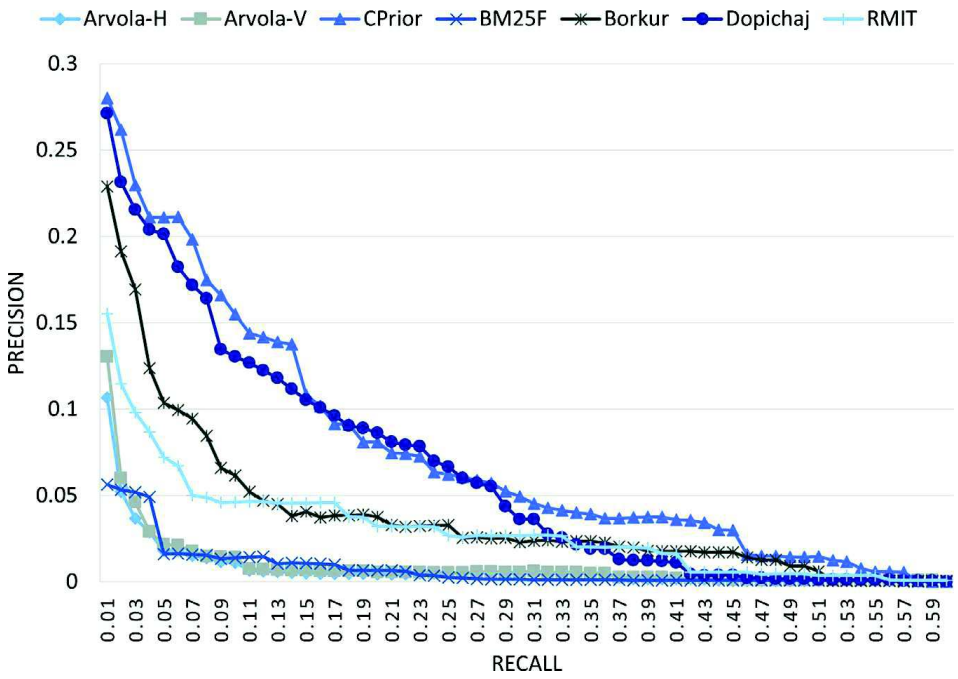

Fig. 15. Comparison with other source of evidence according to the ep-gr metric.

The experiments carried out in this section clearly show that the use of the structural properties of the elements as a new source of evidence proved to be effective. The context importance improves the retrieval effectiveness better than the length prior model.

4.4. Context prior vs. other sources of evidence

In this section, we compare our approach with works using different sources of evidence as priors and different contextualization models, namely:

1. Börkur et al. (Sigurbjörnsson, 2006) : The principal retrieval model is a language model in which the prior probability is estimated by a ratio of the elements length over the length of the collection.
2. BM25t (Géry & Largeron, 2012) : This model is an extension of the famous BM25 model (Jones, Walker, & Robertson, 2000) adapted to XML retrieval. Information about elements length are integrated in the relevance estimation with a tag weighting function which permit to estimate the capacity of a tag to reinforce a relevant term.
3. RMIT (Pehcevski et al., 2005) : In addition to the element length which is a content characteristic, this probabilistic model includes a structural feature which consists of the element absolute path length.
4. Dopichaj et al. (Dopichaj, 2006) : In addition to the element length, this model exploit the element position in the document tree.
5. Paavo Arvola et al. (Arvola et al., 2011) : The model takes into account the hierarchical distance between element and the element position in the hierarchical structure of an XML document. Two models of contextualization have been considered:
   A. Arvola-V: A vertical contextualization taking into account the ancestors of an element to estimate its relevance.
   B. Arvola H: A horizontal contextualization model that considers elements at the same level as the considered element as context.

Fig. 15 shows the comparison of CPrior model with the mentioned models according to the ep-gr metric.

All of the models shown in this comparison use a combination of element content and structural characteristics as priors. Let us note that our model gives better results compared with all the others and, this from the first returned elements. Notice that models integrating directly structural characteristics such as tags (BM25t), element absolute path length (RMIT) and element position (Dopichaj) present better results than those considering content features such as Bokür. This strengthens our intuition and demonstrates that the document structure is an important information, which allows improving the retrieval effectiveness.

On the other hand, the curves of CPrior and Dopichaj evolve almost in the same way. We recall that Dopichaj model uses the elements position in the document tree as source of evidence. It exploits patterns allowing strengthening the score of certain element types as those containing titles elements. Besides the element position, information concerning the structural context of the element is also exploited. The fact that CPrior gives better results proves that our approach of considering priors integrates intuitively the element location in the document during the estimation of its importance.

We do not need to specify which element type to boost but the element intrinsic structural characteristics are sufficient to determine its importance. The BM25t model (which takes into account the impact of tags by estimating the probability of a tag to distinguish the relevant terms of others) and the RMIT model (which exploits the element absolute path length) realize less good results than CPrior. The fact that these models exploit only specific structural characteristics and only one at the same time (tags for BM25t, element position for Dopichaj and the element absolute path length for RMIT) made that they do not benefit from all the power of the document structure. Thing, which CPrior knew how to integrate through the structural context importance concept allowing estimating the probability that an element contains relevant information.

We conclude that our model allows a good combination and a better exploitation of various element structural characteristics through the element type concept. The context importance reflects the impact of the element structure on the capacity of an element to contain relevant content

5. Conclusion and perspectives

Content-oriented XML retrieval identify highly relevant XML elements that would satisfy user information needs. To this end, several sources of evidence are exploited while the most known seems to be the element length. In this article, we hypothesize that the location of an element in the document structure has a considerable impact on the user exploring process.

What makes user considering an element as relevant is the ability of that element to reflect user expectations about where to find relevant information. We therefore exploited a new source of evidence, the structural context importance, in order to quantify the user expectations. This new measure is content-independent, as it only requires the structural information for a given element. We then define a theoretically-driven probabilistic model to estimate the structural context importance.

Using this probabilistic model, we first studied the characteristics of XML elements as reflected by their structural context importance. We then compared context importance to length prior by incorporating each of them as features in a retrieval setting in order to compare their effect on XML retrieval effectiveness. Finally, we proposed a context importance smoothing process within the language modeling framework and investigate whether using context importance like prior probability is effective for XML information retrieval. Our research objectives were investigated by carrying out extensive experiments on IEEE XML document collection.

Regarding the comparison between length prior and context importance prior, the results indicate that even if the latter is completely independent to the former, it however seems that the most important elements are not small (content length is more than ten keywords). Our analysis further indicates that, in contrary to length prior, context importance does not exclude smallest elements when these are relevant.

The comparison with other models using different sources of evidence showed that our model exploits a better combination of element structural characteristics. Our approach to estimate the importance of an element integrates at the same time the element position, the level of the element in the document tree and tags. The concepts of element type and structural context allow benefiting from the document structure at the most and, makes our probabilistic model strong by allowing reflecting the importance of elements such as probably intended by the document writers. We can also deduce that, the hierarchical structure of a document where textual content is intentionally placed at precise places is a deliberate construct of the author in order to convey information and attract readers.

Several perspectives remain open. First of all, since our model gives a non-content and a query independent element importance estimation, it can be generalized by incorporating links between XML elements in the same document and between elements in different documents. This may give a weighting approach of links in a specific collection such as the Wikipedia document collection. On another direction, the intrinsic structural context importance may be improved by integrating element content features. We could for example, integrate the element length or the term weight in order to benefit at the same time from the structure and from the content.

Our model has been evaluated using the IEEE XML document collection, and a posterior analysis is presented but with an overfitting risk. The documents of this collection are all of the same structure; it would be interesting to study the behavior of our model with a collection of heterogeneous document structures.

References

Arvola, P., Kekäläinen, J., & Junkkari, M. (2011). Contextualization models for XML retrieval. *Information Processing & Management, RI XML, 47*, 762–776.

Ashoori, E., Lalmas, M., & Tsikrika, T. (2007). Examining topic shifts in content-oriented XML retrieval. *International Journal of Digital Libraries, RI XML, 8*, 39–60.

Badache, I., & Boughanem, M. (2015). Document priors based On time-sensitive social signals. In *Advances in information retrieval - 37th European conference on IR research, ECIR 2015, Vienna, Austria* (pp. 617–622). March 29 - April 2, 2015. Proceedings. doi:10.1007/978-3-319-16354-3_68.

Banerjee, P., & Han, H. (2009). Language modeling approaches to information retrieval. *JCSE, RI XML, 3*, 143–164.

Bao, S., Xue, G.-R., Wu, X., Yu, Y., Fei, B., & Su, Z. (2007). Optimizing web search using social annotations. In *WWW* (pp. 501–510).

Beckers, T., & Korbar, D. (2010). Using eye-tracking for the evaluation of interactive information retrieval. In *INEX* (pp. 236–240).

Beigbeder, M., Géry, M., Largeron, C., & Seck, H. (2010). ENSM-SE and UJM at INEX 2010: scoring with proximity and tag weights. In *Comparative evaluation of focused retrieval - 9th international workshop of the initiative for the evaluation of XML retrieval, INEX 2010, Vugh, The Netherlands* (pp. 44–53). December 13-15, 2010, revised selected papers. doi:10.1007/978-3-642-23577-1_3.

Blanco, R., & Barreiro, A. (2008). Probabilistic document length priors for language models. In *ECIR, RI XML* (pp. 394–405).

Buscher, G., Cutrell, E., & Morris, M. R. (2009). What do you see when you're surfing?: using eye tracking to predict salient regions of web pages. In *Proceedings of the 27th international conference on human factors in computing systems, CHI 2009, Boston, MA, USA* (pp. 21–30). April 4-9, 2009. doi:10.1145/1518701.1518705.

Damak, F., Pinel-Sauvagnat, K., Boughanem, M., & Cabanac, G. (2013). Effectiveness of state-of-the-art features for microblog search. In *SAC* (pp. 914–919).

Dopichaj, P. (2006). The University of Kaiserslautern at INEX 2006. In *Comparative evaluation of XML information retrieval systems, 5th international workshop of the initiative for the evaluation of XML retrieval, INEX 2006, Dagstuhl Castle, Germany* (pp. 223–232). December 17-20, 2006, revised and selected papers. doi:10.1007/978-3-540-73888-6_22.

Fuhr, N., Lalmas, M., & Trotman, A. (2007). Comparative evaluation of XML information retrieval systems. *5th international workshop of the initiative for the evaluation of XML retrieval, INEX 2006, Dagstuhl Castle, Germany* December 17-20, 2006, revised and selected papers, lecture notes in computer science. Springer.

Ganguly, D., Leveling, J., Jones, G. J. F., Palchowdhury, S., Pal, S., & Mitra, M. (2010). DCU and ISI@INEX 2010: adhoc and data-centric tracks. In *INEX, RI XML* (pp. 182–193).

Géry, M., & Largeron, C. (2012). BM25t: a BM25 extension for focused information retrieval. *Knowledge and Information Systems, 32*, 217–241. doi:10.1007/s10115-011-0426-0.

Guo, L., Shao, F., Botev, C., & Shanmugasundaram, J. (2003). XRANK: ranked keyword search over XML documents. In *Proceedings of the 2003 ACM SIGMOD international conference on management of data* (pp. 16–27). ACM.

Huang, F. (2007). Using language models and topic models for XML retrieval. In *INEX* (pp. 94–102).

Huang, F., Watt, S. N. K., Harper, D. J., & Clark, M. (2006). Compact representations in XML retrieval. In *INEX, RI XML* (pp. 64–72).

Huurdeman, H. C., Kamps, J., Koolen, M., & Wees, J. van (2012). Using collaborative filtering in social book search. *CLEF (online working notes/labs/workshop)*.

Järvelin, K., & Kekäläinen, J. (2002). Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems, 20*, 422–446.

Jay, C., Stevens, R., Glencross, M., Chalmers, A., & Yang, C. (2007). How people use presentation to search for a link: expanding the understanding of accessibility on the web. *Universal Access in the Information Society, 6*, 307–320.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *KDD* (pp. 133–142).

Jones, K. S., Walker, S., & Robertson, S. E. (2000). A probabilistic model of information retrieval: development and comparative experiments - part 1. *Information Processing & Management, 36*, 779–808. doi:10.1016/S0306-4573(00)00015-7.

Kamps, J., Kaptein, R., & Koolen, M. (2010). Using anchor text, spam filtering and Wikipedia for web search and entity ranking. *TREC*.

Kamps, J., Rijke, M. de, & Sigurbjörnsson, B. (2004). Length normalization in XML retrieval. In *SIGIR, RI XML* (pp. 80–87).

Kaptein, R., & Kamps, J. (2013). Exploiting the category structure of Wikipedia for entity ranking. *Artificial Intelligence, 194*, 111–129.

Kirsch, S. M., Gnasa, M., & Cremers, A. B. (2006). Beyond the web: retrieval in social information spaces. In *ECIR* (pp. 84–95).

Kraaij, W., Westerveld, T., & Hiemstra, D. (2002). The importance of prior probabilities for entry page search. In *SIGIR 2002: proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval, August 11-15, 2002, Tampere, Finland* (pp. 27–34). doi:10.1145/564376.564383.

Lalmas, M. (2009). *XML retrieval, RI XML*. Morgan & Claypool Publishers.

Mihajlovic, V., Ramírez, G., Westerveld, T., Hiemstra, D., Blok, H. E., & Vries, A. P. de (2005). TIJAH scratches INEX 2005: vague element selection, image search, overlap, and relevance feedback. In *INEX, RI XML* (pp. 72–87).

Miller, D. R. H., Leek, T., & Schwartz, R. M. (1998). BBN at TREC7: using hidden Markov models for information retrieval. In *Proceedings of the seventh text rEtrieval conference, TREC 1998, Gaithersburg, Maryland, USA* (pp. 80–89). November 9-11, 1998.

Ogilvie, P., & Callan, J. (2005). Parameter estimation for a simple hierarchical generative model for XML retrieval. In *INEX, RI XML* (pp. 211–224).

Ogilvie, P., & Callan, J. (2004). Hierarchical language models for XML component retrieval. In *INEX, RI XML* (pp. 224–237).

Ogilvie, P., & Callan, J. (2007).n.d. Using Language Models for Flat Text Queries in XML Retrieval.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The PageRank citation ranking: bringing order to the web. (Technical Report No. 1999–66). Stanford InfoLab.

Peetz, M.-H., & Rijke, M. de (2013). Cognitive temporal document priors. In *ECIR* (pp. 318–330).

Pehcevski, J., Thom, J. A., & Tahaghoghi, S. M. M. (2005). RMIT University at INEX 2005: Ad Hoc track. In *Advances in XML information retrieval and evaluation, 4th international workshop of the initiative for the evaluation of XML retrieval, INEX 2005, Dagstuhl Castle, Germany* (pp. 306–320). November 28-30, 2005, revised selected papers. doi:10.1007/11766278_23.

Ramírez, G., Westerveld, T., & Vries, A. P. de (2005). Structural features in content oriented XML retrieval. In *CIKM, RI XML* (pp. 291–292).

Robertson, S. E., Zaragoza, H., & Taylor, M. J. (2004). Simple BM25 extension to multiple weighted fields. In *Proceedings of the 2004 ACM CIKM international conference on information and knowledge management, Washington, DC, USA* (pp. 42–49). November 8-13, 2004. doi:10.1145/1031171.1031181.

Sigurbjörnsson, B. (2006). *Focused information access using XML element retrieval*. Universiteit Amsterdam.

Sigurbjörnsson, B., Kamps, J., & Rijke, M. de (2004). Processing content-and-structure queries for XML retrieval. In *TDM* (pp. 35–41).

Termehchy, A., & Winslett, M. (2011). Using structural information in XML keyword search effectively. *ACM Transactions on Database Systems TODS, 36*, 4.

Tran, V. T., & Fuhr, N. (2012). Using eye-tracking with dynamic areas of interest for analyzing interactive information retrieval. In *SIGIR* (pp. 1165–1166).

Velásquez, J. D. (2013). Combining eye-tracking technologies with web usage mining for identifying website keyobjects. *Engineering Applications of AI, 26*, 1469–1478.

Westerveld, T., Kraaij, W., & Hiemstra, D. (2001). Retrieving web pages using content, links, URLs and anchors. *TREC*.

Zhai, C., & Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems TOIS, 22*, 179–214.