Exploiting Links to Improve Search in XML Documents

Samia Berchiche-Fellag
University Mouloud Mammeri of Tizi-Ouzou
Algeria
samfellag@yahoo.fr

Mohamed Mezghiche
University M'Hamed Bougara Boumerdes
Algeria
mohamed.mezghiche@yahoo.fr

Journal of Digital Information Management

ABSTRACT: *This paper describes an approach that exploits links in XML retrieval. The proposed approach consists of reranking the set of documents returned for a given query by considering three sources of evidence namely, the relevance scores w.r.t query of a document neighbours, the text of the anchor links, and the document title tag. Our approach is evaluated on INEX 2006 collection. The results showed significant improvements of the retrieval performances.*

Subject Categories and Descriptors
I.2.7 Natural Language Processing] Language Generation [I.7.2 Document Preparation] Markup languages

General Terms:
Document Processing; XML; Query Processing

Keywords: Information Retrieval, Relevance Propagation, XML, Anchor Text Link, Title Tag, INEX

DOI: 10.6025/jdim/2018/16/4/169-179

Received: 18 March 2018, Revised 24 April 2018, Accepted 3 May 2018

1. Introduction

The particular nature of XML documents which combine content and structure constraints, leads to specific user needs. Indeed users might be interested only on a given part (or elements) of an XML document instead of the whole document. This part might be explicitly specified in the query, this type of queries are named Content And Structure query (CAS), or not, in what is named Content Only query (CO), which contain only simple keywords, in that case the relevant part is identified automatically. The challenging question is how to return the most relevant part of the document related to the query. Several issues arise when we handle these documents, one of the important is how to exploit the document structure and its content to better select and rank relevant elements for CO queries.

In this paper, we are particularly interested in links. It is well known that links between documents express a certain relatedness and semantic proximity between them, that might be exploited to enhance relevance of the results. Exploiting links as retrieval feature has been widely studied particularly in web search. Most of the proposed methods are based on the notion of popularity, considering that «*a page referenced by many good pages is a good page"*. Two popular algorithms exploit this notion, namely PageRank [4] that made the reputation of Google search engine, and HITS [20]. Several derivatives have also emerged from these two algorithms, such as Trafficrank [31], TrustRank [15], Pathrank [24] and Salsa [22].

In addition to hyperlinks, anchor text has been also exploited in commercial search engines to improve web search. The anchor text of the link, is text which consists to a succinct description of the target page so that the reader of the current page can decide whether or not to follow the hyperlink.

The use of anchor text has consistently been shown to improve effectiveness for web retrieval [7] [12] [33].

Since anchor texts are typically short and descriptive, they are potentially similar to queries [11] and can reflect user's information needs. Brin and Page [4] stressed the

importance of anchor text to be associated with the page pointed by the link. Clever system [5, 6] uses the content of the anchors with surrounding text to give more weight to linked pages that use terms appearing in a query. Bharat and Henzinger [3] and Li et al. [23] incorporate weighting, based on the content of the web pages.

TOP HITS model [21] provides an extension of HITS algorithm by incorporating anchor text. Eiron and McCurley [12] investigated properties of anchor text in a large intranet and showed the similarity with real user queries and consensus titles. It shed light that using anchor text can help to obtain better search results from user queries.

Another class of approaches for link analysis, developed in web Information search is the relevance propagation. In such approaches the relevance of documents is computed according to the relevance propagated by their neighbours. Several algorithms have been proposed for this purpose [28][8][29].

Savoy and Rasolofo [28] propose to propagate a fraction of relevance score of each retrieved page to its best neighbours, namely the top-ranked page neighbours returned by a search engine of the considered query.

The cited approaches consider the document level in the processing. However, XML retrieval differs from web search in at least three points:

• The retrieval units in XML retrieval are often elements not the whole document,

• The retrieval unit is not predefined but automatically identified according to the query,

• Links can be either between elements or elements and documents.

These differences lead to several issues regarding the impact of link-based approaches in XML retrieval.

Our objective is to revisit some techniques and issues related to link analysis in the context of XML retrieval.

Our goal in this paper is to exploit relevance propagation [28] combined with anchor text links and the title tag in XML documents. In particular, we propose a propagation based approach that spreads the element score through weighted links computed according to the topical relevance of their anchor text. We hypothesize that the documents pointed in by links containing query terms in their anchor text, must be promoted compared to documents that are not.

This paper is organized as follows, we first overview related work exploiting links on XML documents. We present our approach in section *3*. Section *4* is devoted to the experiments. We conclude on interests of our contribution and introduce our future work.

2. Related Work

Most of works that have exploited links in XML retrieval are based on HITS [20] or PageRank [4] algorithms. XRank [25] is one of the first algorithms, inspired by PageRank, that was adapted to deal with XML retrieval. It assigns a relevance score to an element according to scores of hierarchical and Xlink links between nodes. Mataoui et al [26] propose an approach named *Docrank* based on an adaptation of PageRank to XML document collections.

In the same spirit, Kamps et al [17, 18] propose to estimate a link to rerank the retrieval results, based on a score of a document by combining two parameters namely, *"local indegree"* and *"global indegree"*. "*Global indegree"* represents the links number of the incoming links from collection to an XML document, and *"local indegree"* is the number of incoming links to an XML document from returned results for a given topic. Kimelfeld et al [19] proposed an approach based on filtering and ranking. On filtering the top N (N = 500) relevant documents to the user's query are selected, and on ranking each element of the filtered documents is ranks. Two ranking types are used, one based on the language model and the second combined language model and Hits algorithm.

Pehcevski [27] also exploits links for reranking entities. The entity score is a linear combination of incoming links score to that entity and its initial score.

Since 2007, anchor text was used in Link-the-Wiki (LTW) track, which is link-detection task at INEX. The aim of the LTW track is to automatically identify hyperlinks between documents. Itakura et al [16], Geva [13] and Weerkamp et al [32], this not related to search.

The approach we aim to introduce in this paper also consists of reranking the results obtained during an initial search. However, it is not based on PageRank or Hits algorithms, but it exploits anchor texts, the document title in additional to the propagated relevance score as presented by Savoy [28].

3. The Proposed Approach

3.1 Motivation and Context

First of all, we consider three fundamental hypotheses. The first one expresses that "*a document referenced by several important documents, is important"*. We materialize this importance by spreading scores of source documents to the referenced ones. The second one considers that *"if any query terms appears in the* anchor text link, the referenced document is more likely *to be related to the query"*. Indeed, a document may reference several other documents through links that are not all related to the query. To get relevant documents related to the query, we need to differentiate the target documents related to the query from those that are not. In this purpose, we assign to each document an anchor text links score which reflects its similarity with the query. In the third hypoth

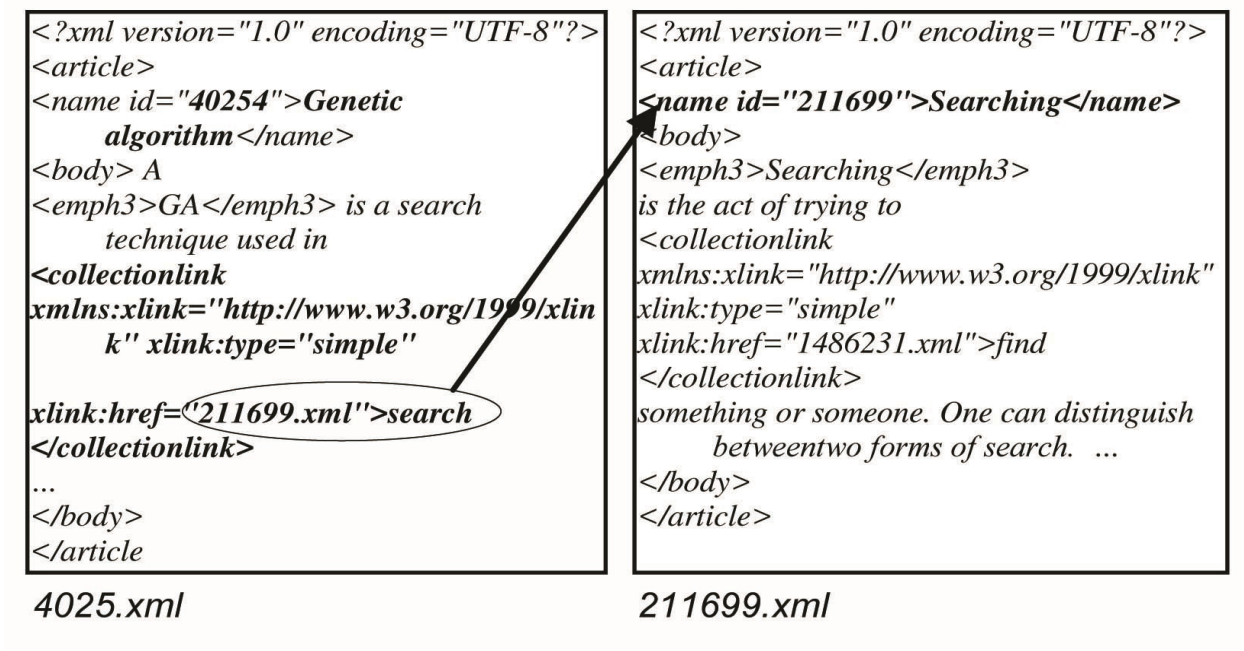

Figure 1. Using the term "*search*" as anchor text from *4025.xml document* to *211699.xml document*

esis, we assume that "*If query terms can be* found in a document title, then this could be relevant to *the query and therefore its elements so are*". This leads us to assign to each target document a title score that measure its similarity with the query. It seems natural to consider that if the query terms are reflected in the title tag of a document, this one will be necessarily returned on the first search phase since it is relevant, knowing that the objective here is to rerank documents returned by the initial search phase by exploiting existing links.

Two kinds of links are considered in XML documents, XLink [10] links (inter documents) and XPointer [14] links (intra documents). We intend to consider in this paper Xlink links.

We consider the carried information by the anchor Xlink links, can impact the relevant documents selection. Indeed, anchor text link may contain query terms, allows assuming that referenced documents through such links, necessarily related to the query.

The figure 1 shows an example of anchor text Xlink link containing the term "*search"* from the document *4025.xml* to *211699.xml* from Wikipedia collection 2006.

3.2 Exploiting Anchor text, Title Tag and Relevance Propagation

3.2.1 General Approach

The general approach we propose consists of reranking a set of potential relevant elements returned by an initial XML retrieval search. The initial set of elements is selected by considering only the relevance between a query and elements. Starting from the set of returned elements, we first select all the referenced documents.

Each element propagates its score to documents it refer ences. We propose to do this propagation guardedly. Indeed, we assume that links are not of the same importance. A document referenced by link having a query terms on its anchor text is more likely to be relevant than the one that has not. For this purpose, the portion of element score we propose to propagate must be proportional to the relevance of the link related to the query. For this, we intend to propagate to referenced document, portion of element score weighted by anchor text links score. We recall that XML links reference documents not elements. Thus, only the first level of propagation is elements documents, the following are documents-documents. Propagation process continues on many levels. Its termination level is generally defined by experiment, but we hypothesize that beyond the third level, one move away from the query and the returned results become uninteresting.

In addition to the propagation scores from incoming links, we propose to consider the title score of target document. Indeed, the link anchor may not contain information, then in order to differentiate target documents related to the query from those are not, we propose, the use of title score of target document.

The final score of referenced document is evaluated based on its title score, document source score propagated weighted with links anchor text scores.

At the end of the propagation process, the new element score is the sum of its initial search score and the score of the document that contains it.

3.2.2 Notation

In order to formally describe our approach, we use the following notation:

C: Corpus of XML documents d. $C = \{d_i\}$, $i = 1, r$

d: Tree of elements, represented as a set of elements e. $d = \{e_i^d\}\ i = 1, p$

e: Set of n terms t_i weighted with w_i. $e = \{(t_1,w_1),\ldots,(t_n,w_n)\}$

q: Query, set of m terms t_i weighted with w_{qi}. $q = \{(t_1,w_{q1}), \ldots,(t_m,w_{qm})\}$

R: Set of elements e occurring in d. $R = \{e_j^d\}\ j = 1, k$

D: Set of documents d returned after links exploitation.

$score\ (q,e)$: Element score *e* related to the query q.

$scorelink\ (d_s,d_t)$: Link anchor text score between source document *ds* and target document d_t.

$scoretitle\ (d)$: Title score of the referenced document *d*.

We precise that:

$score\ (q,e)$ is computed using any XML retrieval model.

R is obtained ***without taking into account*** links.

Documents in C are linked through Xlink links

3.2.3 Propagation Process

The links we exploit are the incoming ones to a document. We exploit the classical assumption that "*the author of a document dealing with a given topic* refers only to documents he considers as relevant to *this topic*". Therefore, we propose that source document d_s will propagate its relevance score to target document d_t[1] [2]. We consider here that the score propagated from document d_s to document d_t is proportional to the importance of the link between them. Indeed, more the link anchor text relates to the query, higher the propagating score is. Hence, the document score propagated d_s to the document d_t is weighted by score link as:

$$scoreprop\ (q,\ d_t) = LinkWeight\ (d_s,\ d_t) \times score\ (q,\ d_s) \quad (1)$$

$score\ (q,\ d_s)$: Source document score. Note that d_s can be the whole document, or a part of it.

$LinkWeight\ (d_s,\ d_t)$: Weighting factor of propagated relevance score from document source to target document and measures the link anchor text. We define It's values in the interval [0, 1].

3.2.3.1 Link Weight Evaluation

Instead of only considering links between documents as linked or not, this is what it is often used in link based approach. In our case, we propose to weight these links according to their anchor text. In this purpose, we introduce $scorelink\ (d_s,\ d_t)$ which measures the link anchor text between source document d_s and target document d_t.

Incoming links of document d_t are all from the documents neighbours that link d_t. The link anchor text may or not contain query terms, therefore may or not be relevant to the query. It is clear that the link anchor text is even more important if it contains the query terms. To quantify this importance, we define its relevance score based on the query recovery rates.

$$scorelink\ (d_s,\ d_t) = \frac{\sum_{i=1}^{|q|} tf\ (t_{qi},\ link)}{\sum_{j=1}^{|link|} tf\ (t_j,\ link)} \quad (2)$$

Where:

$tf\ (t_{qi},\ link)$: Frequency of query term t_{qi} in the link anchor text between d_s and d_t.

$tf\ (t_j,\ link)$: Frequency of term t_j in the link anchor text between d_s and d_t.

$|q|$: Query term number in the link anchor text between d_s and d_t

$|link|$: Total terms number in the link anchor text between d_s and d_t and $|q| <= |link|$

We use the term frequency in the formula because the term can be repeated several times in the anchor link as is shown in this example where document *10699.xml* from Wikipedia collection references document *717222.xml*

```
<collectionlink xmlns:xlink="http://www.w3.org/1999/xlink"
xlink:type="simple"xlink:href="717222.xml">
List of islands of the Faroe Islands</collectionlink>
```

As we explain it above, we introduce *scorelink* as weighting link score, so:

$$LinkWeight\ (d_s,\ d_t) = scorelink\ (d_s,\ d_t) \quad (3)$$

The link anchor text between source and target document may not contain the query terms or not contain any terms. As is the case in this link from *10008.xml* to *35180.xml* document in Wikipedia collection

```
<collectionlink xmlns:xlink=http://www.w3.org/1999/xlink
xlink:type="simple"xlink:href="35180.xml">
1 </collectionlink>
```

In this case, the link weight is low but not zero. We propose, to compute it based on the number of outgoing links from the source document.

$$LinkWeight\ (d_s,\ d_t) = \frac{1}{lkout + \omega} \quad (4)$$

Where:

$lkout$: Number of outgoing links from source document d_s,

ω: Smoothing parameter, in case there is no outgoing links in the source document.

To summarize:

$$LinkWeight(d_s, d_t) = \begin{cases} scorelink(d_s, d_t)\ if\ scorelink(d_s, d_t) <> 0 \\ \dfrac{1}{lkout + \omega}\ otherwise \end{cases} \quad (5)$$

3.2.3.2 Score Title Evaluation

The document title is obviously a distinctive element of the document content. Thus, a document containing query terms in its title is likely more relevant than document which title does not. This makes us defining the title document score by the rates of query terms it contains.

$$scoretitle\ (d) = \frac{\sum_{i=1}^{|q|} tf\ (t_{qi}, title)}{\sum_{j=1}^{|title|} tf\ (t_j, title)} \quad (6)$$

Where:

$tf\ (t_{qi}, title)$: Frequency of query term t_{qi} in the title of document d.

$tf\ (t_j, title)$: frequency of term t_j in the title of document d.

$|q|$: Query term number in the title of document d.

$|title|$: Total terms number in the title of document d and $|q| <= |title|$

3.2.3.3 Final Element Score

In addition to the propagation scores from the incoming links, we add to the target document score, its score title.The use of this score allows us to assess the relevance of the target document's title with the query mostly when the links anchor text does not contain information.

Ultimately, a target document will have the following score

$$score\ (q, d_t) = \propto scoretitle\ (d_t) + \sum_{i=1}^{k} scoreprop_i\ (q, d_t) \quad (7)$$

Where

k: Number of incoming links to the document d_t.

α: Weighting factor to title score of document d_t, we define its values in the interval [0,1].

At the end of exploiting outgoing links from source documents(the above described phase), we obtain as result a set R of documents, ordered by decreasing order of their scores.

We have now to identify relevant elements from R, knowing that every document in the corpus is represented by a set of elements. This will be achieved using the following formula:

$$score\ (q, e) = \beta \times scoreinit\ (q, e) + (1-\beta)\ score\ (q, d_t) \quad (8)$$

With:

$scoreinit\ (q,e)$: Element's initial score from the first search phase,

β: Weighting factor of element and document respective scores.

Final values of α and β parameters will be fixed by experiments, as frequently in Information retrieval domain.

The general algorithm that summarise the different steps is the following:

Search algorithm

***Data*:**

$C = \{d / d = \text{Set of elements } e\}$,

$e = \{(t_1,w_1), \ldots,(t_n,w_n)/\ n$: Number of terms representing the element $e\}$,

$q = \{(t_1,w_{q1}), \ldots\ldots ,(t_n,w_{qm})\}/\ m$: Query terms number $\}$

***Results*:** Set R of answers elements.

- For each element e of a document d of the corpus C
  - Compute $score\ (q, e)$ of e related to the query q
  - Saved in R by decreasing order of $score\ (q, e)$

- For each e of R
  - Select outgoing links to the documents of *C*
  - Save the number of these links in *lkout* see for mula *(4)*

- For each target document d_t
  - Compute $scoretitle\ (d_t)$ with formula (6)
  - Compute $linkweight\ (d_s, d_t)$ of all the incoming links from ds to d_t with formula (5)
  - Compute $scoreprop\ (q, d_t)$ of all the incoming links to d_t with formula (1)
  - Compute $score\ (q, d_t)$ with formula (7)

  Save each target document d_t in D by decreas ing order of scores.

  - For each d_t of D repeat propagation process until reach the required propagation level

- For each e of R
  - Identify its document parents d_t in D
  - Compute $score\ (q, e)$ with formula (8)
  - Save in R by decreasing order of scores.

As mentioned above *scores propagation* will stop following levels number defined by experiment.

3.3 Illustration

We describe below an illustrative example showing the different steps of the approach. We use an example taken from INEX 2006 collection, we consider query 290 and its

top 8 results returned by TopX[30].

The query *290* from INEX has the following format:

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPEinex_topic SYSTEM "topic.dtd">

<inex_topic topic_id="290" ct_no="9">

<title>"genetic algorithm"</title>

<castitle>//article[about(.,
"geneticalgorithm")]</castitle>

<description>Find information about the
history, algorithm,function, data struc-
tures and implementation of genetic algo-
rithms.</description>

<narrative>I am doing an experiment which
needs to tune more than 4 parameters. I
was told that the genetic algorithm is a
suitable method for this. I want to have
an overview of this type of algorithms,
and especially I am interested in the al-
gorithms, functions, data structures and
implementations of genetic algorithm. Rel-
evant elements should mention any of the
above information of genetic algorithms.
</narrative>

<ontopic_keywords>genetic algorithm; GA;
algorithm</ontopic_keywords>

</inex_topic>
```

The *Top 8* results returned by TOPX to *query 290* are listed in *table1*, some references of these results and their anchor texts are presented in *table 2*.

We note that returned elements can be the whole document, as document *554480.xml*, with */article [1]* as result, or only a part of it, as for *554546/article [1]/* body[1]/section [3].

At this stage, we have to describe how to compute referenced documents scores. Let's take for example the *555480.xml* document whose title is *"chromosome genetic algorithm"*, referenced by documents *554546.xml* and *555213.xml* with *"chromosome genetic algorithm"* as anchor text. Remind that query terms are *"genetic algorithm"*.

We obtain using formula *(6)*:

$$Scoretitle\,(555480.xml) = 2/3 = 0{,}6666$$

$$Scorelink\,(554546.xml, 555480.xml) = 2/3 = 0{,}6666$$

$$Scorelink\,(555213.xml, 555480.xml) = 2/3 = 0{,}6666$$

The score of $555480.xml$ is computed with formula *(7)*

$$Score\,(q, 555480.xml) = \alpha * scoretitle\,(555480.xml) + Scoreprop\,1\,(q, 555480.xml) + Scoreprop2\,(q, 555480.xml)$$

We replace $Scoreprop_i(q, 555480.xml)$ using formula *(1)*

$$Score\,(q, 555480.xml) = \alpha * scoretitle\,(555480.xml) + Linkweight\,(554546.xml, 555480.xml) \times score\,(q, 554546.xml) + Linkweight\,(555213.xml, 555480.xml) \times score\,(q, 555213.xml)$$

$Score\,(q, 554546.xml)$ and $Score\,(q, 555213.xml)$ are respectively initial scores of $554546.xml$ and $555213.xml$.

We compute *Linkweight* with formula *(5)*

$$Linkweight\,(554546.xml, 555480.xml) = Scorelink\,(554546.xml, 555480.xml) = 0{,}6661$$

$$Linkweight\,(555213.xml, 555480.xml) = \text{Scorelink}(555213.xml, 555480.xml) = 0{,}6653$$

With $\alpha = 0,6$, the score value of $555480.xml$ after the first

Rank	Document	Document Title	Path	Rsv
1	554546	Crossover genetic algorithm	/article[1]/body[1]/section[3]	0.80836
2	555480	Chromosome genetic algorithm	/article[1]	0.73780
3	901056	Interactive genetic algorithm	/article[1]	0.73133
4	901162	Interactive evolutionary computation	/article[1]/body[1]/section[2]	0.73028
5	901149	Human based genetic algorithm	/article[1]	0.72794
6	2519383	Merchant of Venice	/article[1]/body[1]/p[1]	0.72498
7	555213	Mutation genetic algorithm	/article[1]	0.72285
8	**40254**	**Genetic algorithm**	**/article[1]/body[1]/section[9]**	**0.72235**

Table 1. TOPX results to the query 290 in INEX 2006

Rank	Source Document	Anchor Text	Target Document
1	554546	Genetic algorithm	40254.xml
		Chromosome genetic algorithm	555480.xml
		Mutation genetic algorithm	555213.xml
2	555480	Chromosome genetic algorithm	6438.xml
		Genetic algorithm	40254.xml
		data structure	8519.xml
		Crossover	554546.xml
3	901056	Genetic algorithm	40254.xml
		Interactive evolutionary computation	901162.xml
		Evolutionary art	213371.xml
		Karl Sims	767665.xml
		Human-based genetic algorithm	901149.xml
4	901162	Evolution strategy	940033.xml
		Interactive genetic algorithm	901056.xml
		Genetic programming	12424.xml
		Human-based genetic algorithm	901149.xml
		Evolutionary art	213371.xml
5	901149	Genetic algorithm	40254.xml
		Interactive genetic algorithm	901056.xml
		Distributed artificial intelligence	237629.xml
		public distributed artificial	467465.xml
		Interactive evolutionary computation	901162.xml
6	2519383	**No references**	
7	555213	Genetic algorithm	**40254.xml**
		Genetic operator	371709.xml
		Chromosome genetic algorithm	555480.xml
		Bit	3364.xml
		random variable	25685.xml
8	40254	Genetic programming	12424.xml
		Interactive genetic algorithms	901056.xml
		Simulated Annealing	172244.xml
		Recombined	554546.xml
		Genetic algorithm	**40254.xml**

Table 2. TOP8 documents results references

level of exploiting links is then

$score\,(q, 555480.xml) = 1{,}4204,$

Remind that 555480.*xml* had an initial score resulting from the first search. Which value is 0,73780.

The final score of 555480.*xml* in the first level using the formula *(8)* is:

$score\,(q, 555480.xml) = \beta * scoreinit\,(q, 555480.xml) + (1-\beta) * score\,(q, 555480.xml)$

$score(q,555480.xml) = \beta * 0{,}73780 + (1-\beta) * 1{,}529$ with $\beta = 0{,}3$

$score(q, 555480.xml) = 1{,}2153.$

This represents how is performed in a first level of links

exploitation.

The final given score in *table 2* of 555480*.xml* which value is 0,7650 is calculated at the third level.

The same computation was applied for all referenced documents.

We notice that the result element of document 40254*.xml* which title is *"genetic algorithm"* is entirely pursuant to the query, but not well ranked *(8th)* at the end of the first stage, even if it is referenced by most of the results.

After considering links across three levels of reference, the document score becomes:

$Score\,(q, 40254.xml) = 7,27285$

With this value, element's scores of the document 40254*.xml* will be increased and their ranking changed.

The final score of the element 40254 */article[1]/body[1]/section[9],* noted 40254_{el} *is* $score(q,40254el) = \beta*0.72235+(1-\beta)*6,9107$ with $\beta = 0,3$ $score(q, 40254el)=5,3077$.

The achieved results on the sample evaluation INEX campaign is shown in *table 3.*

Rank	New Rank	Document	Document Title	Path	New Rsv
8	**1**	**40254**	**Genetic algorithm**	**/article[1]/body[1]/section[9]**	5,3077
3	**2**	901056	Interactive genetic algorithm	/article[1]	3,0217
5	**3**	901149	Human based genetic algorithm	/article[1]	0,9393
1	**4**	554546	Crossover genetic algorithm	/article[1]/body[1]/section[3]	0,7820
7	**5**	555213	Mutation genetic algorithm	/article[1]	0,7673
2	**6**	555480	Chromosome genetic algorithm	/article[1]	0,7650
4	**7**	901162	Interactive evolutionary computation	/article[1]/body[1]/section[2]	0,5340

Table 3. TOP 8 Results to the query 290 on a sample INEX collection 2006 after exploiting links with our approach

4. Experimentations

The aim of this experiment is to compare our model to a well ranked system in INEX 2006. TOP-X [30] is one of the highest ranked systems in INEX 2006, which consider simplified XML data model, where Xpointer and Xlink links are disregarded.

4.1. INEX: Initiative for the Evaluation of XML Retrieval

We used for our experiments the INEX 2006 collection [9]. INEX is the only campaign assessment of XML information retrieval systems. The main INEX evaluation purpose is to promote their search in XML documents by providing a test collection, and assessment procedures to allow participants to benchmark their results. The test collection consists of a set XML documents, queries and relevance judgments, and uses a collection made from English documents from Wikipedia.

INEX consists of several tasks such as *"focused"* task,*"thorough"* task, *"Best in context"* task. We based our evaluations on the *"focused"* task.

4.2. Data Collection

The collection contains about *659 388* documents extracted from Wikipedia and provides a set of *126* queries for evaluation. Note that in Wikipedia, the<*collectionlink*> that generates anchor text is pointing to a full document and not to its elements. Table 4 lists the characteristics of this collection.

Collection size	**4.6 GO**
Documents Number	**659388**
Links number	**16737300**
Topics number	**126**

Table 4. Features of INEX 2006 collection

4.3. Evaluation Protocol

Our experiments are performed based on search results returned by TOPX system on INEX 2006 *"Focused"* task to CO (Content Only) queries.

We have experimented 25 queries over the whole INEX 2006 collection. For each query, we rerank the top 50 results returned by TopX.

We used the normalized cumulated gain $nxCG[t]$ measure

which was used in the evaluation of the *"focused"* task in INEX 2006.

With $nxCG[i]$ measure, system performance was reported at several rank cutoff values (i).

For a given topic, the normalized cumulated gain measure is obtained by dividing a retrieval run's xCG vector by the corresponding ideal xCI vector.

$$nxCG[i] = \frac{xCG[i]}{xCI[i]} \qquad (9)$$

$xCG[i]$ takes its values from the full recall-base of the given topic.

$xCI[i]$ takes its values from the ideal recall-base and i ranges from 0 and the number of relevant elements for the given topic in the ideal recall base.

For a given rank i, the value of $nxCG[i]$ reflects the relative gain the user accumulated up to that rank, compared to the gain that could have attained if the system would have produced the optimum best ranking.

First of all, in order to assess the different parameters, we conducted preliminary experiments on small sample of documents and queries (20 queries, 100 documents) from INEX 2006 collection. We set the parameters used in our formulas, to the following values: $\alpha = 0{,}6$; $\beta = 0{,}3$; $\omega = 10$ and the propagation process level is fixed to three.

The obtained results by the experiments we conducted over the whole INEX 2006 collection are presented below on table 5.

We performed the Student test and attached + and ++ to the performance number in the table when the test passes at 95% and 99% confidence level, respectively.

Figure 2 compares our approach with TOPX system. This figure indicates that our approach based on exploiting links is better than TOPX model which do not.

Improvement in results is observed on different gain values at 10, 20, 30 and 50 documents. Although the highest performances, are observed in the 10 and 20 first documents. This is because our approach could provide large scores to the relevant XML elements satisfied with user's query using relevance propagation, anchor text links score and title document score. We can say that our approach can improve the effectiveness of XML search.

	nxCG @10	**nxCG @20**	**nxCG @30**	**nxCG @50**
TOPX	0,754	0,709	0,681	0,633
Own system	0,843	0,764	0,707	0,647
%improvement	**10,493**$^{++}$	**7,234**$^{+}$	**3,679**	**2,209**

Table 5. Results for the « Focused » Task with the nxCG metric at different cutoffs

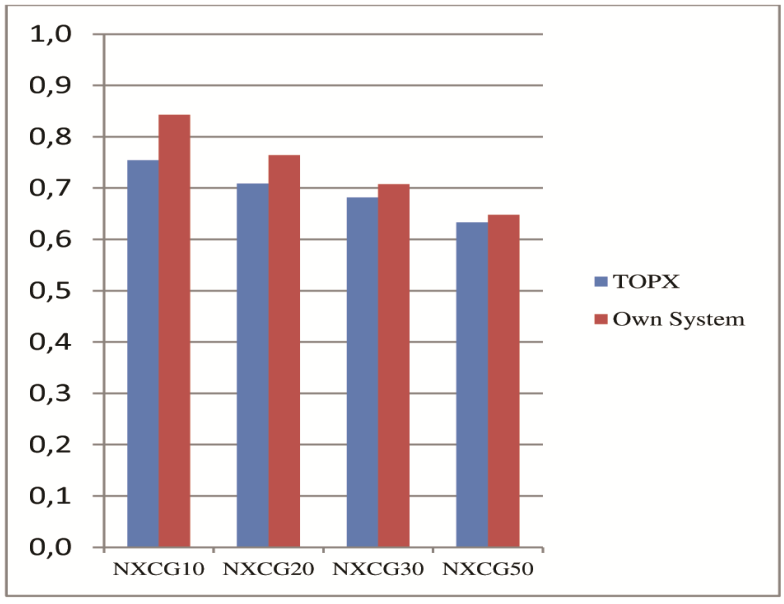

Figure 2. Comparative graph the achieved results of our approach to that of TopX

4.5. Discussion

As noticed in introduction, all methods we've mentioned [25,17,18,19] in this work, dealing with links exploitation in XML documents are based on reranking results of an initial search. However, and despite of this apparent similarity, our approach has three major differences compared to these methods.

First, most of these methods, have adapted Pagerank [25] or Hits [19] for XML retrieval, and have not exploited the relevance propagation although it achieved good results [28], [8], [29]. It is precisely because of the good results obtained in links exploitation on the web, we thought adapt it to XML. This is confirmed by our experimental results.

The second aspect that distinguishes our approach over the others is the use of anchor text links that are not used at all in SIR to exploit links. Especially the use of this anchor as a weighting parameter to the propagation score which has been used neither in structured information retrieval nor on the Web. Indeed, considering anchor text links allows targeting documents related to the query and then raise their scores while decreasing the score from documents that are not. *How to evaluate this score?*

"When a relevant source document references a target *document, it attests that, it is relevant"*. We have represented this intuition by propagating a part of source document's score to the target document.

How much score the source document should it give *the target document?* To this end we propagate a score proportional to the anchor link between the two documents; as mentioned before if anchor links score is high, it means that the target document is relevant.

The third point is the use of title tag. Even if this parameter has been widely used in IR, it was not included in the SIR during the links exploitation, by none of mentioned methods. Indeed, this parameter is very important when the links anchor text does not contain information. It thus allows us to distinguish the target documents related to the query from those that are not. We noticed that its inclusion introduces higher performance, as proved by our experimental results.

5. Conclusion

We presented in this paper a new way to exploit links in XML documents based on relevance score propagation. Especially the main contribution consists of propagating score proportionally to the significance of a link between a source document and its neighbors. The significance of a link is computed by the relevance of the anchor text of the link and the query.

We used Top X search model [30] to get a sorted initial lements list. By using this list, the source documents scores obtained are propagated according to incoming Xlink links to the target documents. In addition to the relevance propagation, we also evaluated the links anchor text and textual information carried by the title tag of the referenced documents by assigning a score to each of these representations. At the end, each document score is calculated according to the propagated score, the link score and the title score. The experiments we conducted on 2006 Wikipedia collection showed that our approach improve the results of the first XML retrieval system.

In the future we plan to exploit semantics links anchor text and a calculation method to define the most appropriate value of the propagation level.

Acknowledgments

We would like to express our warm thanks to Pr Mohand Boughanem, full professor at Paul Sabatier university of Toulouse (France) for his invaluable help and guidance in writing this paper.

References

[1] Berchiche-Fellag, S., Boughanem, M. (2011). Exploitation des liens dans la recherche d'information dans les documents XML, *In*: Colloque sur l'Optimisation et les Systèmes d'Information.

[2] Berchiche-Fellag, S. (2012). Propagation de pertinence et exploitation du texte ancre des liens et de la balise titre pour améliorer la recherche dans les documents XML, In Inforsid.

[3] Bharat, K., Henzinger, M. R. (1998). Improved algorithms for topic distillation in a hyperlinked environment, *In*: SIGIR '98, p. 104–111. ACM Press.

[4] Brin, S., Page, L. (1998). The anatomy of a large-scale hypertextual web search engine, *In*: Proc. of the 7th international conference on *World Wide Web* (*WWW*), p. 107–117, Brisbane, Australia.

[5] Chakrabarti, S., Dom, B., Raghavan, P., Rajagopalan, S., Gibson, D., Kleinberg, J. (1998). Automatic resource compilation by analyzing hyperlink structure and associated text, In WWW7, p. 65–74. Elsevier.

[6] Chakrabarti, S., Dom, B. E., Kumar, S. R., Raghavan, P., Rajagopalan, S., Tomkins, A., Gibson, D., Kleinberg, J. (1999). Mining the Web's link structure, *In*: Computer, 32 (8)60–67.

[7] Craswell, N., Hawking, Robertson, S. (2001). Effective site finding using link anchor information. *In*: Proc. of SIGIR, p. 250–257.

[8] Crestani, F., Lee, P. L. (2000). Searching the Web by constrained spreading activation, In Information Processing and Management, 36 (4) 585-605.

[9] Denoyer, L., Gallinari, P. (2006). The Wikipedia XML corpus, In SIGIR Forum 40 (1) 64–69.

[10] Deroze, S., Maler, E., Orchard, D. (2001). Xml link-

ing language (xlink), Technical Report 1.0, In World Wide Web Consortium (W3C), W3C Recommendation, juin.

[11] Dou, Z., Song, R., Nie, J.-Y., Wen, J.-R. (2009). Using anchor texts with their hyperlink structure for web search, In Proc. 32nd Annual Intl ACM SIGIR Conf. on Research and Dev. In Information Retrieval, (July).

[12] Eiron, N., McCurley, K. S. (2003). Analysis of anchor text for web search, In Proc. 26th Annual Intl ACM SIGIR Conf. on Research and Dev. in Information Retrieval, (July).

[13] Geva, S. (2008). Gpx: Ad-hoc queries and automated link discovery in the wikipedia Focused Access to XML Documents (p. 404-416): Springer.

[14] Grosso, P., Maler, E., Marsh, J., Walsh, N. (2003). Xml pointer language (xpointer) , Technical repor*t,* World Wide Web Consortium (W3C), W3C Recommendation.

[15] Gyongyi, Z., Garcia-Molina, H., Pedersen, J. (2004). Combating web spam with trustrank, *In*: Proc. of the 30th International Conference on Ver Large Databases, pages 576–587.

[16] Itakura, Y. K., Charles Clarke, L. A. (2008). Adhoc, Book, and Link-the-Wiki Tracks, In INEX 2008, p. 132-139.

[17] Kamps, J., Fachry, K. N., Koolen, M., Zhang, J. (2008). Using and Detecting Links in Wikipedia, In Focused Access to XML Documents, p. 388-403.

[18] Kamps, J., Koolen, M. (2008). The importance of link evidence in wikipedia, In Lecture Notes in Computer Science, p. 270–282, Heidelberg.

[19] Kimelfeld, B., Kovacs, E., Sagiv, Y., Yahav, D. (2007). Using language models and the hits algorithm for XML retrieval, In INEX 2006, p. 253–260.

[20] Kleinberg, J. (1998). Authoritative sources in a hyperlinked environment, *In*: Proc. 9th Annual ACM-SIAM Symposium Discrete Algorithms, pages 668–677.

[21] Kolda, T., Bader, B. (2006). The tophits model for higherorder web link analysis, *In*: Workshop on Link Analysis, Counterterrorism and Security.

[22] Lempel, R., Moran, S. (2000). The stochastic approach for link-structure analysis (SALSA) and the TKC effect. *Computer Networks*, 33 (1) 387-401.

[23] Li, L., Shang, Y., Zhang, W. (2002). Improvement of HITS based algorithms on web documents, *In*: WWW '02, p. 527–535. ACM Press.

[24] Li, J., Zhao, Y. (2009). Pathrank: Web page retrieval with navigation path, In ECIR '09, p. 350–361, Berlin, Heidelberg, Springer-Verlag.

[25] Lin, G., Feng, S., Chavdar, B., Jayavel, S., Xrank. (2003). Ranked search over xml documents, In SIGMOD'2003, San diego, CA.

[26] Mataoui, M., Mezghiche, M. (2009). Prise en compte des liens pour améliorer la recherche d'information structurée, CORIA 2009, France, p. 363-372.

[27] Pehcevski, J., Vercoustre, A. M., Thom, J. M., (2008). Exploiting Locality of Wikipedia Links in Entity Ranking, In ECIR 2008, p. 258–269, Heidelberg.

[28] Savoy, J., Rasolofo, Y. (2004). Hyperliens et recherche d'information sur le web, *In*: 7eme journée internationales d'Analyse statistique des Données Textuelles.

[29] Shakery, A., Zhai, C. X. (2003). Relevance propagation for topic distillation, UIUC trec 2003 web track experiments, *In*: TREC, p. 673–677.

[30] Theobald, M., Broschart, A., Schenkel, R.,Solomon, S., Weikum, G. (2006). TopX - AdHoc Track and Feed back Task. *In*: INEX 2006, p. 233-242

[31] Tomlin, J. A. (2003). A new paradigm for ranking pages on the world wide web, *In*: Proc. of the 12th International WWW Conference, p. 50–355.

[32] Weerkamp, W., Krisztian Balog., Edgar Meij. (2008). A Generative Language Modeling Approach for Ranking Entities, *In*: INEX, p. 292-299.

[33] Westerveld, W. K. T., Hiemstra, D. (2002). Retrieving web pages using content, links, urls and anchors. *In*: Tenth Text Retrieval Conference, TREC '02, p 663–672.