When is the Structural Context Effective?

Muhammad Ali Norozi
Dept. of Computer and Information Science
Norwegian University of Science and Technology
Trondheim, Norway
mnorozi@idi.ntnu.no

Paavo Arvola
School of Information Sciences
University of Tampere
Tampere, Finland
paavo.arvola@uta.fi

ABSTRACT

Structural context surrounding the relevant information is intuitively and empirically considered important in information retrieval. Utilizing this context in scoring has improved the retrieval effectiveness. In this study we will objectively look into the significance of the *structural context* in contextualization process, and try to answer the core question of under which circumstances do we need to deal with the such types of context?

Categories and Subject Descriptors

H.3.3 [**Info. Search and Retrieval**]: Search process

1. INTRODUCTION

Document parts, referred to as elements, have both a hierarchical and a sequential relationship with each other. The hierarchical relationship is a partial order of the elements, which can be represented with a directed acyclic graph, or more precisely, a tree. In the hierarchy of a document, the upper elements form the context of the lower ones. In addition to the hierarchical order, the sequential relationship corresponds to the order of the running text. From this perspective, the context covers the surroundings of an element. An implicit chronological order of a document's text is formed, when the document is read by a user.

In focused retrieval, the use of context is a driving force to alleviate or "un-bias" the retrieval of items with varying length. Namely, information retrieval is based on evidence of the retrievable units at hand, and longer text units have indeed more textual evidence. This has led to a play-safe strategy where the larger elements are favoured by retrieval systems. How effective the context is to neutralize the side-effects or bias because of size or length (smaller elements with less textual evidence gets same opportunity to satisfy the users need), has been reported experimentally in many studies [1–3, 6, 9, 10, 8, 7]. The question asked here is: why the structural context is important in the retrieval of focused items? In addition, we also ask if the use of context, under certain circumstances (worst-case), would harm the retrieval. This means if the context is poor or even misleading.

2. CONTEXT

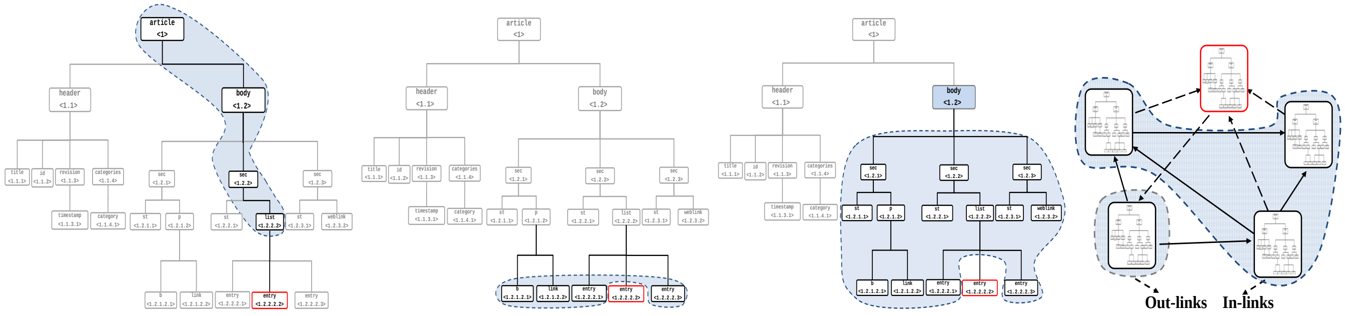
In semi-structured documents, context of an element covers everything in the document excluding the element itself. The surrounding items or elements of the relevant information is the *context*. The representation of the semi-structured documents aims to follow the established structure of documents, i.e., an academic book is typically composed of ⟨`chapters`⟩, ⟨`sections`⟩, ⟨`subsections`⟩ etc., structures. ⟨`chapter1`⟩ is followed by ⟨`chapter2`⟩ and within ⟨`chapter1`⟩, ⟨`section1`⟩ is followed by ⟨`section2`⟩. Elements ⟨`section1`⟩ and ⟨`section2`⟩ are siblings, and hence most likely, semantically related. The following element takes the concepts further from the preceding elements, and the preceding elements provide the basics or foundation for the following elements. Therefore, together in the document order, the *preceding* and *following* elements form a strong and connected perspective (the kinship structural context), surrounding the relevant information. Two general types of context can be distinguished based on the standard relationships. Hierarchical context, for one, refers to the ancestors, whereas horizontal refers to the preceding and following elements [3]. In existing studies, context has been referred to *externally* as the hyperlink structure of the elements as well. The context is *internal* when it is considered from within the document, and it is external when it is considered outside the document(s).

Contextualization [3] is a re-scoring model, where the basic score, usually obtained from a full-text retrieval model, of a contextualized document or element is re-enforced by the weighted scores of the contextualizing documents or elements (elements in the sub-tree of interest or structural context). In this section, we will formalize the context from in and outside the document using contextualization model.

2.1 Structural Context

Structural context is the *sub-tree of interest* from the hierarchical tree structure of the semi-structured document. *Internally*, in *hierarchical contextualization* [3], the intrinsic tree structure within the XML document is employed. Structural context in hierarchical or vertical contextualization is the context based on parent-child relationship in document's hierarchical structure. An element's parent or ancestors are accounted to be the structural context, while contextualizing the element itself. The sub-tree of interest is shown in Figure 1(a). *Horizontal contextualization* [3] takes into account the sibling elements in the document's hierarchical structure as the structural context. If we visualize the document's hierarchically tree structure, horizontal structural context is horizontal, as it is based on one level (the same level as the element to be contextualized) of the tree at a time (see Figure 1(b)). The most recent form of contextualization, the *Kinship contextualization* [7], is both horizontal (siblings) and vertical (ancestors & descendants

DIR 2013, April 26, 2013, Delft, The Netherlands.



(a) Hierarchical (b) Horizontal (c) Kinship (d) Citation

Figure 1: Structural context, the sub-tree of interest, example taken from [7]

elements) but intrinsically non-hierarchical perspective of the hierarchical information. Structural context is hence both vertical and horizontal in the document's hierarchical form, Figure 1(c).

And *externally*, in *citation contextualization* [8], the document's hyperlink structure is taken in to account. The structural context here is based on the hyperlinks' graph of documents hyper-linking (connecting) one another in form of inlinks (indegree) and outlinks (outdegree). In this case, the sub-graphs instead of tree of interest are the out-links graph and the in-links graphs (see Figure 1(d)).

2.2 Why Structural Context?

Structural context is the essential component of the Contextualization model [1]. With contextualization model, using the structural context, the aim is to rank higher an element in a good context (strong evidence in the structural context) than an identical element in a not so good context (less or no evidence in the structural context) within the document. And therefore, retrieve elements independent of their sizes. A small element, in term of size, can be viewed and hence scored in relation to its structural context, and its smaller size (which means having less evidence in total) doesn't stop it from being selected as one of the best results.

In order to cope up with the "biasness" issue (described earlier), in contextualization model, the weight of a relevant element is adjusted by the basic weights of the elements in the structural context (its contextualizing elements). In addition to basic weights, each element in the structural context of the contextualized element, should possess an *impact* factor. An higher impact factor shows the importance of the contextualizing element and vice versa. The role and relation of elements in the structural context are operationalized by giving the element a contextualizing weight. A contextualization vector is defined to capture the impact factor of each contextualizing element, and this contextualization vector is represented by a g function in Equation 1.

2.3 Contextualization and Random Walks

Random walk principle is employed, for contextualization, to induce a similarity structure over the documents based on the containment and reverse-containment relationships (element, sub-element and vice versa). Hence, these relationships affect the weight each element, in the structural context, has in contextualization.

The premise is that *good structural context* (identified by random walk and the contextualization model [7]) provides evidence that an element in focused retrieval is a good candidate to satisfy the user's need and therefore, the elements should be contextualized by the elements in the sub-tree of interest. Hence, the good structural context contains a strong likelihood factor that should be used to deduce that the contextualized element is a good candidate for the posed query.

The tree-structure of the XML document (Figure 1) is assumed to be a graph. In order for the structural context to take part in the contextualization process, each of the nodes in the sub-tree of interest should possess an impact factor. Conceptually, the impact factor is produced in the following manner: Myriad of random surfers traverse the XML graphs. In particular, at any time step a random surfer is found at an element and either (a) makes a next move to the sub-element of the existing element by traversing the containment edge, or (b) makes a move to the parent-element of the existing element, or (c) jumps randomly to another element in the XML graph. As the time goes on (the number iterations), the expected percentage of surfer at each node converges to a limit, the dominant eigenvector of the XML graph. This limit provides the impact or strength of each element in the structural context of the element to be contextualized, in the form of g function. All the elements, in the structural context of the contextualized element, are considered for contextualization; where the contextualization vector g identifies the importance of each of the unit of the structural context (Equation 1).

2.4 Generalized Combination Functions

The generalized re-ranking combination function based on the random walk principle, which also captures the structural context, can be formally defined as follows:

$$
\begin{aligned}
CR(x, f, C_x, g^k) \quad &= \quad (1-f)\cdot BS(x) + \\
& \qquad f \cdot \frac{\displaystyle\sum_{y\in C_x} BS(y)\cdot g^k(y)}{\displaystyle\sum_{y\in C_x} g^k(y)}
\end{aligned}
\qquad (1)
$$

where

- $BS(x)$ is the basic score of contextualized element x (text-based score, e.g., $tf \cdot ief$)
- f is a parameter which determines the weight of the context in the overall scoring.
- C_x is the kinship context surrounding the contextualizing element x, i.e., $C_x \subseteq structural\_context(x)$, $\subseteq$, because only the structural context containing the query terms are considered.

- $g^k(y)$ is the generalized contextualization vector based on random walk, which gives the authority weight (the impact) of y, the contextualizing elements (elements in structural context) of x in the sub-tree of interest.

3. EFFECTS OF CONTEXTUALIZATION ON DIFFERENT TEST COLLECTIONS

Structural context in the contextualization framework, is independent of the basic weighting scheme of the elements and it could be applied on the top of any query language, retrieval systems and test collections. The effects of contextualization on different test collections have been observed in the existing studies. Contextualization model has been applied on the top of different and competitive baseline systems using a diverse set of test collections, e.g., semantically annotated Wikipedia collection from INEX 2009[1], IEEE collection, and iSearch scientific collection [3, 7, 8]. In order to get the best possible baseline system, a data fusion was performed based on sum of normalized scores (CombSUM) [11] and Reciprocal Ranking [4] of INEX 2009 submitted runs.

In the experimental evaluation the retrieval effectiveness at different granularity levels were observed. Mainly, retrieval effectiveness at paragraph, article and INEX's focused retrieval level selection has been observed. The approaches were evaluated using the evaluation framework provided by TREC and INEX evaluation initiatives. The reported results were shown to be promising using both TREC and INEX evaluation framework [3, 7].

The focused task in INEX ad-hoc track is to retrieve most focused elements satisfying an information need without overlapping elements. An overlapping result list means that the elements in the result list may have a descendant relationship with each other and share the same text content. For instance, in Figure 1 the ⟨`entry`⟩ element ⟨`1.2.2.2.1`⟩ and the ⟨`sec`⟩ element ⟨`1.2.2`⟩ are overlapping. In the existing studies, in the focused retrieval task, the INEXs' focused approach is followed, considering a result list where only one of the overlapping elements from each branch is selected. This means that including the ⟨`sec`⟩ element in the results would mean excluding the entry element in the results or vice versa.

Contextualization and the fusion approach as scoring methods, however, do not take any stand on which elements should be selected from each branch. Thus a structural fusion has been performed, where the element level selection is taken from the baseline run and subsequently re-rank the elements of the baseline run.

3.1 Test Settings

The hierarchical structure of XML documents in the Wikipedia 2009 collection, are captured using the dewey encoding scheme (as shown in Figure 1). This way each element in the document possess a unique index within the document, and together with document's unique id, this becomes unique for the entire collection. The tree structure of XML documents are converted into a matrix, and random walk is performed on this matrix at indexing time, as it is described in detail, in our earlier work [7]. The contextualization vector g^k from Equation 1 is computed off-line for each and every XML document in the Wikipedia collection. This suggests that computing g^k vector is feasible for a reasonably large XML document collections. At the query time, the scores from g^k vector and the basic scores are combined to produce an overall ranking score, using Equation 1.

In the generalized combination function given (Equation 1), the contextualization force has to be parametrized. In our earlier work [7], the contextualization force was tuned and reported the values leading to best overall performance. In the parametrization process it was found that the optimal values of contextualization force f (from Equation 1) lies in the range, ($f \in \{.25,..., 2.50\}$). These optimal values for f are obtained by using cross-validation technique. A 68-fold[2] cross-validation (or complete cross-validation) technique has been performed - by randomly partitioning the collection into 68 training and test samples based on the number of assessed topics. Of the 68 samples, a single sample is retained as the validation set for testing, and remaining 67 samples are used as training set. The cross-validation process is repeated 68 times (for each fold), with each of 68 samples used exactly only once as validation set. These 68 independent or unseen samples are then combined to produce a single or a set of estimations for parameter f.

3.2 Query Term Probabilities

If a relevant element does not contain any of the query term(s), it does not match to the query. Hence, in order to retrieve such elements, some expansive methods, such as contextualization, ought to be used. It seems obvious that, in a relevant small element, the probability of occurrence of a query term is smaller than in a larger element. In order to demonstrate this lack of evidence on small elements, we calculated some posteriori probabilities for query term occurrences in a relevant document (R_d) and in a relevant paragraph (R_p, i.e., the relevant ⟨`p`⟩ elements from the XML graph), based on INEX 2009, 68 topics (title field) and their relevance assessments. The probabilities are calculated as the fraction of relevant elements containing any query term, or all query terms over all relevant elements of same kind. The probability of occurrence of any query term (from the query Q) in a R_p and in a R_d respectively are:

$$P\left(\bigcup_{q\in Q} q \middle| R_p\right) = 0.847, \quad P\left(\bigcup_{q\in Q} q \middle| R_d\right) = 0.995$$

This means that the probability of occurrence of none of the query terms in R_p and a R_d is 0.153 and 0.005 respectively[3]. Accordingly, the probabilities of occurrence of all the query terms in R_p and R_d, respectively are:

$$\prod_{q\in Q} P\left(q|R_p\right) = 0.127, \quad \prod_{q\in Q} P\left(q|R_d\right) = 0.469$$

The difference in the amount of evidence at different granularity levels become even more obvious, when we draw the frequencies of the query terms in this picture. A query term occurs on average 3.4 times in a R_p and 45.4 times in a R_d.

4. WORST CASE ANALYSIS

Worst-case for a document d, in contextualization models, means when structural context of element x is chosen such that:

$$structural\_context(x) \notin elements_y(d) \qquad (2)$$

$$(\forall \text{ elements } y \text{ in document } d \quad \text{where } x \text{ and } y \in d)$$

[1] Wikipedia collection containing 2.66 million semantically annotated XML documents (50, 7 Gb) and 68 related topics provided by the INEX 2009 ad-hoc track [5].

[2] 68, because of the 68 topics from INEX 2009.

[3] Test is performed without stemming or stop-word removal

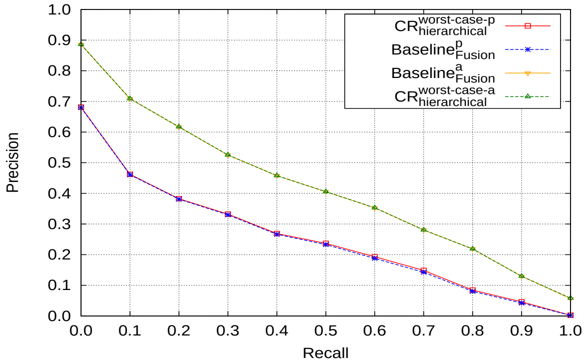


Figure 2: Precision - recall, worst-case scenario at article (a) and paragraph (p) granulation and the fusion baseline systems.

The *non-structural context* (Equation 2), should theoretically expose the worst-case effects of the contextualization model. Non-structural context is structural by definition, but physically not in the structural context of element x. How should we interpret the non-structural context, in order to experimentally visualize the worst-case scenario? Instead of taking the actual and true structural context, we randomly select the structural context from another non-relevant but retrieved document. Such a document (retrieved but not relevant) would have misleading evidence (false positive) and hence best suited for the worst-case evaluation. Randomly selecting a document with zero basic score would be trivial and not suitable for our purposes.

By applying this simplistic approach on every element to be contextualized, we can formulate the worst-case scenario. We have used the reciprocal rank fusion approach (fusing 98 INEX 2009 runs) as the baseline system, for worst-case analysis, which has been used before in our earlier work, find further details from [8]:

$$RRScore(e, q) = \sum_{r \in R} \frac{1}{k + rank(r, e, q)} \tag{3}$$

where

- R is the set of runs (rankings)
- and $rank(r, e, q)$ returns the rank of element e as a result of query q in run r.
- If e is not in the ranking, $rank(r, e, q)$ is not defined and the outcome of $\frac{1}{k+rank(r,e,q)}$ is 0.
- The parameter k is for tuning.

Figure 2 reveals the worst-case depiction of the contextualization model. Not unexpectedly, the worst-case scenario is as good as the baseline system, slightly better but not significant enough to be visible statistically. We can claim here that, when the structural context is chosen randomly (haphazardly), in the worst-case, the contextualization method will not be worse than the basic scoring method.

5. CONCLUSIONS AND FUTURE WORK

Structural context is the sub-tree of interest, utilized in conjunction with contextualization model, improves the retrieval effectiveness. We have presented an exploratory and theoretical study into the use of structural context from elements in the hierarchical structure of information, to improve retrieval performance. We looked into the structural context from document's hierarchical structure internally, and hyperlinks structure externally. We looked theoretically into the hypothesis that structural context gathered from within the document, "horizontally" and "vertically" using the hierarchical tree structure of document, and from outside, using the hyperlinks graph structure of documents referencing each other, influences the retrieval effectiveness. Worst-case experiments also support the theoretical soundness of contextualization, i.e., if we apply contextualization blindly, in the worst case, we would have as good result as the basic scoring method. The results obtained in this study are in-line with the earlier work on contextualization [1, 3, 6, 7, 9, 10]. In this study we have experimented with semi-artificial data, in the sense that we muddled the context for the worst-case analysis. However, in real data the quality of context varies as well. For example in Wikipedia there are different kinds of pages ranging from listings to topically very coherent documents. In order to get the best results in retrieval, analysing the quality and topical coherency of context would be of great benefit. The analysis of context may be topic dependent, since some queries may have contextual parts. For instance a query: "Losses Belgium in WW2", crave for answers about *Belgium* in the context of *WW2*.

6. REFERENCES

[1] P. Arvola, M. Junkkari, and J. Kekäläinen. Generalized Contextualization Method for XML Information Retrieval. In *Proc. of the 14th ACM CIKM*, pages 20–27. ACM, 2005.

[2] P. Arvola, J. Kekäläinen, and M. Junkkari. The Effect of Contextualization at Different Granularity Levels in Content-oriented XML Retrieval. In *Proc. of the 17th ACM CIKM*, pages 1491–1492. ACM, 2008.

[3] P. Arvola, J. Kekäläinen, and M. Junkkari. Contextualization Models for XML Retrieval. *Info. Processing & Management*, pages 1–15, 2011.

[4] G. Cormack, C. Clarke, and S. Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proc. of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759. ACM, 2009.

[5] S. Geva, J. Kamps, M. Lethonen, R. Schenkel, J. Thom, and A. Trotman. Overview of the INEX 2009 ad-hoc track. *Focused Ret. and Evaluation*, pages 4–25, 2010.

[6] Y. Mass and M. Mandelbrod. Component Ranking and Automatic Query Refinement for XML Retrieval. *Advances in XML IR*, pages 1–18, 2005.

[7] M. A. Norozi, P. Arvola, and A. P. de Vries. Contextualization using hyperlinks and internal hierarchical structure of wikipedia documents. In *Proc. of the 21st ACM CIKM*, pages 734–743. ACM, 2012.

[8] M. A. Norozi, A. P. de Vries, and P. Arvola. Contextualization from the Bibliographic Structure. In *Proc. of the ECIR 2012 Workshop on Task-Based and Aggregated Search (TBAS2012)*, page 9, 2012.

[9] P. Ogilvie and J. Callan. Hierarchical Language Models for XML Component Retrieval. *Advances in XML IR*, pages 269–285, 2005.

[10] G. Ramirez Camps. *Structural Features in XML Retrieval.* PhD thesis, SIKS, the Dutch Research School for Information and Knowledge Systems., 2007.

[11] J. A. Shaw and E. A. Fox. Combination of multiple searches. In *The 2nd TREC.* Citeseer, 1994.