International Journal of Web Information Systems

Finding target and constraint concepts for XML query construction

Keng Hoon Gan Keat Keong Phang

Received 18 April 2015
Revised 5 May 2015
Accepted 5 May 2015

Finding target and constraint concepts for XML query construction

Keng Hoon Gan
School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia, and
Keat Keong Phang
Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia

Abstract

Purpose – This paper aims to focus on automatic selection of two important structural concepts required in an XML query, namely, target and constraint concepts, when given a keywords query. Due to the diversities of concepts used in XML resources, it is not easy to select a correct concept when constructing an XML query.

Design/methodology/approach – In this paper, a Context-based Term Weighting model that performs term weighting based on part of documents. Each part represents a specific context, thus offering better capturing of concept and term relationship. For query time analysis, a Query Context Graph and two algorithms, namely, Select Target and Constraint (QC) and Select Target and Constraint (QCAS) are proposed to find the concepts for constructing XML query.

Findings – Evaluations were performed using structured document for conference domain. For constraint concept selection, the approach CTX+TW achieved better result than its baseline, NCTX, when search term has ambiguous meanings by using context-based scoring for the concepts. CTX+TW also shows its stability on various scoring models like BM25, TFIEF and LM. For target concept selection, CTX+TW outperforms the standard baseline, SLCA, whereas it also records higher coverage than FCA, when structural keywords are used in query.

Originality/value – The idea behind this approach is to capture the concepts required for term interpretation based on parts of the collections rather than the entire collection. This allows better selection of concepts, especially when a structured XML document consists many different types of information.

Keywords Web search and information extraction, Web structure/linkage mining, Indexing and retrieval of XML data

Paper type Research paper

1. Motivation

Up-to-date, eXtensible Markup Language (XML) (Bray *et al.*, 1997) is the most widely adopted standard used to represent structured resources or documents. With its well-defined standard, it is adopted for representing contents that requires both

International Journal of Web Information Systems
Vol. 11 No. 4, 2015
pp. 468-490

1744-0084
DOI 10.1108/IJWIS-04-2015-0017

This work was supported by the USM Short Term Grant 304/PKOMP/6312142], Universiti Sains Malaysia.

meanings and structures. As it has been well received by both research and commercial communities, development of methods like query languages (Chamberlin, 2002; Boag *et al.*, 2007; Trotman, 2009; Carmel *et al.*, 2003) query optimization (Petkova *et al.*, 2009; Li *et al.*, 2009; Gan and Phang, 2014a), retrieval models (Itakura and Clarke, 2010; Li and van der Wiede, 2009), search engines (Taha and Elmasri, 2010; Theobald *et al.*, 2008; Liu *et al.*, 2007; Graupmann *et al.*, 2004; Cohen *et al.*, 2003) and schema definitions (Fallside and Walmsley, 2004) can be seen in many recent works.

The potential of semantically rich XML resources on the Web is obvious. With contents represented in a conceptual and structural rich form, these resources have more to offer to solutions in the information-seeking domain. When resources are incorporated with concepts like role, category, topic, class, attributes, etc. (Huffman and Baudin, 1997), a query would be able to utilize it for a better definition of information needs.

However, due to the diversities of concepts used in XML resources, it is not easy to select a correct concept when constructing an XML query. For example, a popular XML query type, NEXI can be written as //movie[about(.//title, Avatar) AND about(.//director, James Francis Cameron)]. In this query, there can be many variations used for same or similar concept for title, like movie title, name, etc. To overcome this issue, automated concept finding has become a popular method in solving the problem.

Literatures that are related to concepts or structures finding for XML query either use probabilistic method or natural language method to obtain the structural parts of the query. In these literatures, keywords query or descriptive query is used without needing to specify its structural parts. In the former, probabilistic methods are used to estimate the association between a term and its structure (Petkova *et al.*, 2009; Kim *et al.*, 2009; Hsu *et al.*, 2004; Bao *et al.*, 2010). In general, this estimation applies well when a collection has either simple or little structure types, where there are little variants in term of syntax or less ambiguities in term of concept usage. In the latter, grammar templates are used to find concepts to construct XML query from a given descriptive query. For example, Tannier (2005) uses XSL Transformation to generate NEXI structured query from a natural language query. However, as this method obtains concepts/structures from the query itself, it works on query where concepts are specified in natural language form, e.g. "searching paragraphs about databases", where "paragraph" is the structure to return. Similarly, Gan and Phang (2014b) proposes an intermediate query model that is able to represent structural keywords from various inputs such as forms, advance query, etc. to construct them into structured queries like NEXI and XQuery.

In this paper, we focus on solving issues in structures finding method which is similar to the former group of literatures rather than the latter. In particular, we are interested to explore issues in structures finding when the structures contained in document are more complex. In such scenario, existing collection-based probabilistic estimation is insufficient to suggest a good concept or structure.

The rest of the paper is organized as follows. The next section of this paper presents background and issues of this research. In Sections 3, 4 and 5, we present the proposed approach for target and constraint finding. This is followed by the description of evaluation settings in Section 6. In Section 7, we present and discuss the results of evaluation, and, lastly, we conclude in Section 8.

2. Problem definitions

2.1 Background

In XML retrieval, there are several types of queries available like xQuery, NEXI, as well as other looser forms. Although they have different syntax, they share similar structural information needs.

- *Path-based*: NEXI (Trotman, 2009). //TARGET PATH [about(FILTER PATH, FILTER TERM) (e.g. //movie[about(.//title, avatar) AND about(.//director, james francis cameron)];
- *Concept-based*: COMPASS (Graupmann *et al.*, 2004), XSEarch (Cohen *et al.*, 2003). CONCEPT = VALUE (e.g. author = Tolstoy), LABEL: KEYWORD, LABEL: or : KEYWORD (e.g. authors: Kempster : Stirling); and
- *Fragment based*: XML Fragment (Carmel *et al.*, 2003). <CONTEXT>TERM </CONTEXT><TARGET>CONTEXT </TARGET> (e.g. <book><year> 1973 </year><title>Search </title></book><TARGET>book </TARGET>)

From the above examples, information needs can be specified as content needs and concept needs. The content needs are keywords indicating the information user would like to seek. Concept needs are keywords containing the content keywords to a narrower subset of results based on categories, types, kinds, roles, topics, etc. Concept needs in a query can be further classified into target concept and constraint concept.

2.1.1 Target concept. During query analysis, a concept can be used to define the overall query's scope or focus. For example, in fragment query by Carmel *et al.* (2003), a target concept signifies the component type expected as target result, e.g. <target>book</target>. In NEXI CAS query by Trotman (2009), a target concept defines what to be returned to user.

Thus, in query transformation, finding target is part of the process to form a complete structured query. In Petkova *et al.* (2009), target for a query is obtained from concepts of query terms via a set of operations (i.e. expand, aggregate and order). Li *et al.* (2009) utilizes the root node of subtree (known as master entity) associated to its query terms to obtain the target. Using a different approach, Hsu *et al.* (2004) selects concept nodes using a context analysis method to form its structured query. Nodes selection is carried out by exploring structure paths of query's terms based on semantic distance of query terms on the document structure. Bao *et al.* (2010) also proposes that an effective keyword search in XML search should be able to identify the correct type of the target node(s).

2.1.2 Constraint concept. Besides identifying target concept for a query, a concept is used to constraint the meaning of terms in a query. For example, when a term is used in various kind of elements, indicating a concept will restrict the query to a specific type of elements.

Collection-based probabilistic methods are often used in the selection of the most relevant concept for a term based on collection statistics, e.g., Petkova *et al.* (2009) and Kim *et al.* (2009) use unigram language model to determine the most relevant concept for a term. When collection-based frequency is insufficient, Bao *et al.* (2010) incorporate node-type (equivalent to our concept) frequency (Cvia(T, q)), with an additional factor known as In Query Distance (IQD), utilizing keywords distance within a query in its concept selection.

2.2 Issues

The main issue of current works is that the measure of concept and keyword does not take into consideration the weighting of keyword under a concept. For example, if a keyword is contained in a long text, hence, it is associated with the concept will be less relevant. Another limitation of current works is to identify ancestor structural nodes (other than a direct associated parent structural node) but could potentially be a better constraint concept. However, including distanced nodes will eventually increase the choices of concepts that makes it difficult for selection especially for collection with heterogeneous structures.

In the next section, we will describe how the concepts required for query interpretations are obtained. We propose a context-based term weighting approach to improve the current methods of finding query target and constraint concepts selection for XML query construction. The general idea behind this approach is to capture the concepts required for term interpretation based on parts of the collections rather than the entire collection. This emphasizes on a specific usage of concepts especially in collection consisting many different contexts of information. Using this approach, we focus on two improvisations as listed below.

(1) Refining scoring methods for context-based term weighting (Section 4).

(2) Improvisation of target/constraint selection based on scores from context-based term weighting (Section 5).

3. Preliminaries

Before a query can be interpreted, knowledge for interpretation is required. The source of knowledge is obtained from XML documents of a collection. We also refer an XML document as structured document in this work. This document can either be created based on a schema or none. Although most structured documents contain both logical tags and descriptive tags in its contents representation, we focus on the latter, as these tags reflect concept or meaning that can be used for query interpretation.

An excerpt of a structured document about conference workshops from a conference site is shown in Figure 1.

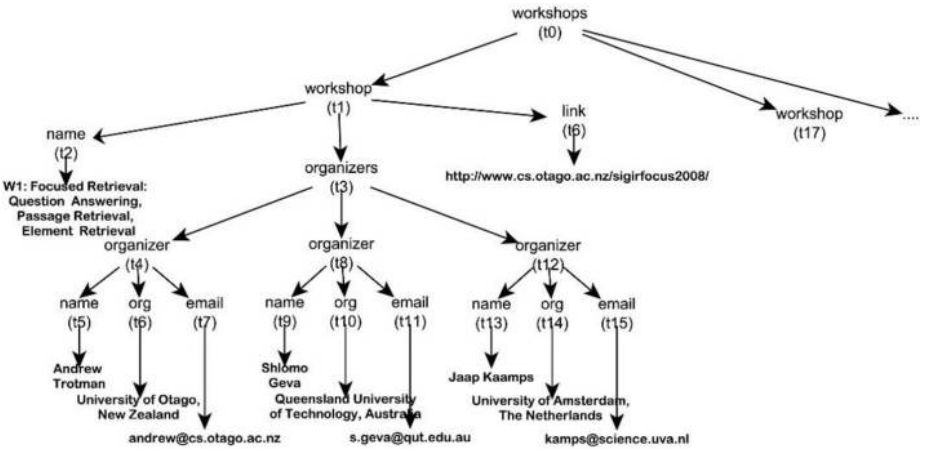

Figure 1.
Partial document structure of a structured document from conference collection

Definition 1
(Structured Document) A structured document is a rooted, acyclic graph defined as $G_{doc} = (V, ED)$, where V is a set of nodes which can be either a structure or a content, $V = \{v: v \in v_{content} \cup v \in v_{struc}\}$. In G_{doc}, its root node and all intermediate nodes are structures, v_{struc}, while its leaf nodes, $v_{content}$ are contents or data.

There are three useful items in a structured document, i.e. term, concept and context. A term is a meaningful unit of string obtained from either a content node or a structure node of a structured document, G_{doc}.

Definition 2
(Term) Given a structured document, G_{doc}, a content term, ct, is a term obtained from content node, $v_{content}$. A structure term, st, is a term obtained from structure node, v_{struc}. ct can be a single word or a phrase obtained from term parsing method, whereas st is required to refer to the exact string of v_{struc}.

In a structured document, the meaning of a term can be observed through the relation between a term (content node) and its structures (structure node). As such, we can obtain the prediction of what a term means, by capturing the relationships between the term and its structures. Different from thesaurus, the meaning of a term are reflected through the usage of structures (including tags, markups, annotations) in the tree. We call these structures as concepts. We use the word concept to refer to the type (or class) of a structure, e.g. name, hotel, article, etc. and the word structure to refer a unique physical unit of a structure term, or structure node.

Definition 3
(Concept) Given a structured document, G_{doc}, a concept, cpt, for a content term, ct, is a structure obtained from structure node, v_{struc} of G_{doc}, where v_{struc} is a parent or ancestor of content node, $v_{content}$ containing ct.

A context defines a specific condition of where a concept is used. Context may not be significant in collection where its documents have homogeneous structures, due to the simplicity and the size of the information. However, in collection where documents contain heterogeneous structures, there may be different parts in a document that presents information of different kinds. Hence, when a document contains many different parts of information, it has become not meaningful if these parts are treated as the same type under the same document. Dividing document into contexts overcomes this by classifying parts of the same type under the same context.

Definition 4
(Context) Given a structured document, G_{doc}, a context, ctx, for a content term ct and its concept cpt, is a structure obtained from structure node, vstruc of G_{doc}, where v_{struc} is an ancestor of structure node, v_{struc} containing cpt.

Now, we proceed to define two important outcomes of this work, i.e. the target concept and constraint concept. Target and constraint concepts are obtained from the result of query interpretation. For the purpose of query interpretation, the terms, concepts and contexts related to the query are modeled as a query context graph, QG_{CTX}. A query context graph captures all possible contexts and concepts for the terms given in a query.

Definition 5
(Query Context Graph) Consider all the terms, ct and st, in a keywords query, Q_K and their corresponding concepts, cpt and contexts, ctx, a query context graph, QG_{CTX} is a rooted directed acyclic graph,

$QG_{CTX}(V_{QG}, E_{QG})$, such that:

- the root node, V_{QGroot} is the ctx;
- the intermediate node(s), $V_{QGinter}$ is cpt of ct;
- the leaf node, V_{QGleaf} is either a content node, ct, or a structure node, st;
- each leaf node has none or only one intermediate node;
- if leaf node is ct, the edge between intermediate node and root node is weighted edge with $score_{CTXPROX}$ to reflect the frequency of a cpt under the ctx;
- the edge between leaf node and intermediate node is weighted edge with $score_{CW}$ to reflect the frequency of ct in cpt; and
- if leaf node is st, the edge between leaf node and root node is weighted edge with $score_{CTXPROX}$ to reflect the frequency of st under ctx.

Consider a query, "andrew trotman jaap kamps" that looks for any outcomes by these two person on a conference collection, there are multiple relevant query context graphs such as graph with root "workshop", root "paper" and root "poster committee". If we looked at another query, "andrew trotman focused retrieval", the relevant query context graph may also have similar root "workshop" but with a different set of concepts. If a structural term is used in the query, e.g. "organizer" and "workshop" in "organizer-focused retrieval workshop", they can be reflected in the query context graph as well. See Figure 2.

3.1 Constraint concept weighting

Constraint concepts for query interpretation can be obtained from the intermediate nodes of a query context graph. Here, we define the possible candidates of concepts that can be selected as constraint concepts.

Definition 6
(Constraint Concept Candidates) Consider a query context graph, QG_{CTX} of an keywords query, Q_K, a constraint concept candidate, $constraintCand_{cpt}$, for a content term in Q_K is a concept node for the content term in QG_{CTX}:

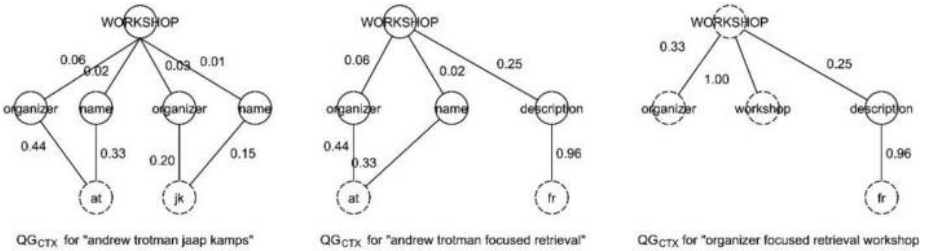

Figure 2.
Query context graph examples

$$\forall ct \in Q_K, constraintCand_{cpt} = \{V_{QG} \in QG_{CTX}\}$$

where V_{QG} is cpt and V_{QGleaf} is ct and $E(V_{QG}, V_{QGleaf})$ is the edge connecting both nodes.

The weight for a constraint concept of a term, $score_{CW}$ is a real number in the range of [0, 1] obtained from the term weighting function for concept, $score_{CW}$ (cpt, ct):

$$\forall ct, score_{CW}(cpt, ct): constraintCand_{cpt} \rightarrow score_{CW}$$

3.2 Target concept weighting

Target concepts for query interpretation can be obtained from the root of query context sub graph. Here, we define the possible candidates that can be selected as target concepts.

Definition 7

(Target Concept Candidates) Consider a query context graph, QG_{CTX}, of keywords query, Q_K, a target concept candidate, $targetCand_{cpt}$, for the query can either be the root node of QG_{CTX} (or concept leaf node for subtarget) of QG_{CTX}:

$$\forall Q_K, targetCand_{cpt} = \{V_{QGroot} \in QG_{CTX} \cup V_{QGleaf} \in QG_{CTX}\}$$

where V_{QGleaf} is st.

The weight for a target concept of a term, $score_{CTXPROX}$ is a real number in the range of [0, 1] obtained from the context proximity function for context, $score_{CTXPROX}(cpt_{ct}, ctx)$.

$$\forall Q_K, score_{CTXPROX}(cpt_{ct}, ctx):targetCand_{cpt} \rightarrow score_{CTXPROX}$$

An interpreted query contains both contents and concepts, but cannot contain concepts only. An interpreted query can have multiple target concepts as well as multiple constraint concepts. Each constraint concept needs to bind to a content. We define a query interpretation as follows.

Definition 8

(Query interpretation) A query interpretation, QI, is a tuple

QI = (TARGET, CONSTRAINT), where $TARGET = \{target_{cpti} \mid 1 \leq i \leq n\}$ and $CONSTRAINT = \{(constraint_{cptj} : constraint_{ctj} \mid 1 \leq j \leq n)\}$.

4. Context-based term weighting approach

This section explains how terms are weighted under different contexts for the purpose of query interpretation. As there are two types of terms that we are looking at in a keywords query, two different term weighting models are presented.

4.1 Content term weighting

For content term, we first measure the importance of term with respect to a concept. This weight, given as $score_{CW}$, will be used for selection of constraint concept later. Second, we measure the importance of these term and concept among various contexts. This

weight, given as $score_{CTXPROX}$, will be used for selection of target concept later. A simple term weighting representation is illustrated in Figure 3.

4.1.1 Term weighting among concepts ($score_{CW}$). Within a context, a term may be associated with multiple concepts. First, term is weighted against individual elements in the collection, followed by aggregation of weights according to the element type (concept). In our term weighting measure, we take into consideration distant concepts in the structure hierarchy.

4.1.1.1 Term-Element Weighting. Various term weighting models have been actively used in document retrieval, such as TFIDF (Salton and Buckley, 1988), OKAPI BM25 (Robertson and Zaragoza, 2009) and Language Model (Ogilvie and Callan, 2002). Because term weighting in document has been a mature field in information retrieval, its scoring models are extended to cater for term weighting in element (Wang *et al.*, 2007). In our concept weighting measure, we shall adopt these basic term weighting models. The weight of a content term, ct in an element, e, is denoted as $score_{TW}$ (e, ct) below.

TFIEF (Term Frequency Inversed Element Frequency):

$$score_{TW}(e, ct) = tf_{ct,c} * \log \frac{N_e}{ef_{ct}}$$

Where e is element, ct is content term, tf(ct, e) is frequency of tc in e, N_e is frequency of e in collection and ef_{ct} is frequency of e in collection that contains ct.

4.1.1.2 Term-Concept Weighting. Concepts are generalizations of elements based on the type of structure of the elements. The weight of a term with respect to a concept is estimated based on the relatedness between a term and a type of structure (referred as concept in this paper) instead of an element. For weighting concept on structure hierarchy, the distance of concept from term is taken as $D_{e,ct}$. Let us denote E_{cpt} as the set of elements of type cpt, where cpt is a concept. The weight of a content term, ct in a concept, cpt, is denoted as $score_{CW}$ (cpt, ct) below:

$$score_{CW}(cpt, ct) = \frac{\sum_{c \in Ecpt} \left[score_{TW}(e, ct) * \frac{1}{D(e, ct)} \right]}{|Ecpt|},$$

where cpt is concept, ct is content term, E_{cpt} is set of elements of type cpt.

This scoring factor, $score_{CW}$ (cpt, ct), captures the intuition that, when ranking a constraint concept, a direct relationship of concept and term is favored. A direct relationship means that the lesser additional terms contained by the concept is better, e.g. <author>Andrew Trotman</author>, compared to the one contained together in a paragraph with other terms, e.g. <keynote abstract>[…] Andrew Trotman began his career at […] </keynote abstract>.

$$\forall ct, score_{CW}(cpt, ct) : constraintCand_{cpt} \rightarrow score_{CW}$$

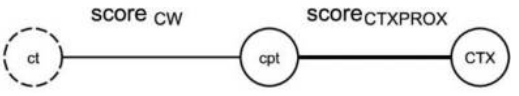

Figure 3.
Weighted edge in term weighting representation

4.1.2 Term: concept weighting among contexts ($score_{CTXPROX}$). Besides weighting a term with respect to a concept within a context, it is necessary to measure the importance of the term if it appears in multiple contexts. This is to show the importance of a term and concept under a particular context compare to another. For example, "author: an-drew troman" may be more important under the context "conference" compare to "tutorial". This is measures using the proximity between "concept:term" and its context.

Contextual proximity between concept:term and context is obtained by combining two proximity factors, i.e. Distance-based Closeness, and Content-based Closeness. Here we denote concept:term unit as cpt_i,ct_j, a context unit as ctx_k.

4.1.2.1 Distance-based closeness. For distance-based closeness, we measure the distance between $cpt_{i,ctj}$ and ctx_k using the approach of semantic space in a taxonomical tree. Edges are used as distance. Distance between concept nodes within the space is taken as measurement of semantic closeness. To measure the semantic closeness, we measure distance similarity, DISim, between a concept and its context for each term unit, ct_j, is given as:

$$DISim(cpt_{i,ctj} : ctx_k) = \frac{1}{edge(cpt_{i,ctj} : ctx_k)},$$

where edge is the number of nodes interval between $cpt_{i,ctj}$ and ctx_k.

4.1.2.2 Content-based closeness. For content-based closeness, we measure the occurrences of $cpt_{i,ctj}$ and ctx_k. The semantic closeness between two concepts node in a taxonomy can also be measured based on contents frequency that subsume concepts. Here, for a term unit, ct_j, the density, DEN, of a concept and context pair, is given as:

$$DEN(cpt_{i,ctj} : ctx_k) = \frac{pf(cpt_{i,ctj} : ctx_k)}{\sum_{i=1}^{N} pf(cpt_{i,ctj} : ctx_k)},$$

where pf is the pairs frequency of $cpt_{i,ctj}$ and ctx_k.

4.1.2.3 Contextual proximity score. Contextual proximity is taken as the product of both scoring of distance-based closeness and content-based closeness,

$$score_{CTXPROX}(cpt_{i,ctj} : ctx_k) = DISim(cpt_{i,ctj} : ctx_k) * DEN(cpt_{i,ctj} : ctx_k)$$

Consider a collection of conference domain, some examples of weighted terms with both $score_{CW}$ and $score_{CTXPROX}$ are shown in Figure 4. Weight on the thin edge, $score_{CW}$, of each term captures the importance of a term with respect to a concept. Weight on the thick edge, $score_{CTXPROX}$, of each term captures the importance of a concept:term pair with respect to a context. For instance, for term "andrew trotman" (see "at" in Figure 4), within the "WORKSHOP" context, this term has stronger relationship with the concept, "organizer" and "committee", based on the concept score, $score_{CW}$. For this example, term weighting model BM25 is used to calculate $score_{TW}$. Another information that is captured is the importance of con-text. For instance, when "committee: andrew trotman" appears in "WORKSHOP" context with a weight 0.06 and "FULL PAPER PC" contexts

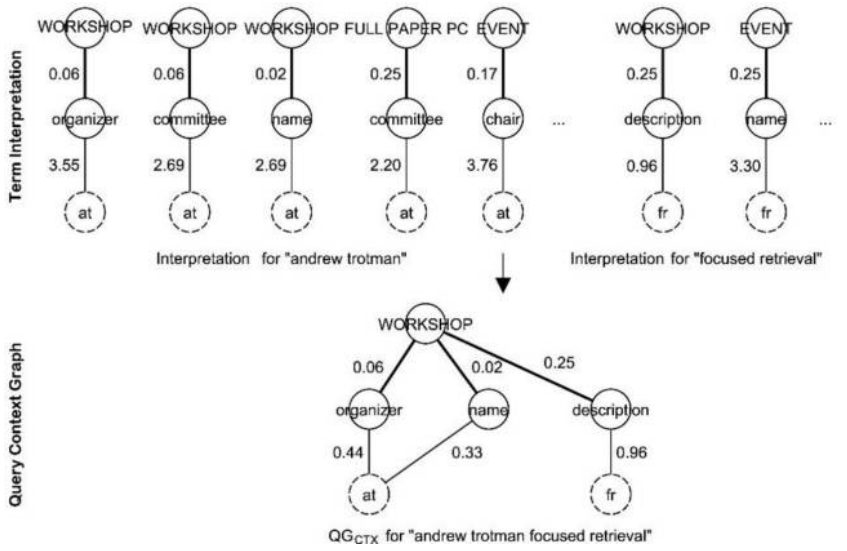

Figure 4.
Examples of query interpretation using query context graph

with a weight 0.25, this shows that committee: andrew trotman is more relevant to the context, "FULL PAPER PC".

4.2 Structure term weighting

In a query, its keyword can be a structure term. Similar to content term, these terms also helps to define the query's context. In this section, we explain how a structure term is weighted with respect to a certain context. For structure term weighting, the weight of a structure in a context is measured with respect to the taxonomical characteristic, which is based on how structures are related to each other in a collection. Repetitive structures such as a list of items of the same structure like <book>, merely reflect the usage of the same structure type; hence, it does not affect the importance of a structure. As such, its frequency will not be taken for structure term weighting.

The scoring of structure term in a context is similar to distance-based closeness measure used for measuring content term. The main difference is the source of semantic space. For content term, its semantic space is taken from the taxonomical tree of document structure, whereas, for structure term, its semantic space is taken from the taxonomical tree of concept structure. The former includes both concepts and contents in its tree structure, but the latter only includes concepts in its tree structure.

4.2.1 Distance-based closeness (for concept tree). For distance-based closeness, we measure the distance between st_i and ctx_j using the approach of semantic space in a taxonomical tree of concept structure. Edges are used as distance. Distance between concept nodes within the space is taken as a measurement of semantic closeness. To measure the semantic closeness, we measure distance similarity, DISim, between the corresponding concept for each structure term, and its context:

$$DISim(st_i : ctx_j) = \frac{1}{edge(st_i : ctx_j)}$$

Where, edge is the number of nodes interval between st_i and ctx_j.

4.2.2 Contextual proximity score (for concept tree). Contextual proximity for structure term is taken as the average distance-based closeness for all non-repetitive term interpretations for structure term, $I_{structure}$ in collection:

$$score_{CTXPROX}(st_i : ctx_j) = \frac{\sum_{i=1}^{n} DISim(st_i : ctx_j)}{n}$$

Where, n is the total of unique term interpretations for structure term, st_i.

Consider the same collection of conference domain, the weighted structure terms with $score_{CTXPROX}$ are shown in Figure 5. Different from content term, structure term only has one weighted edge. The weight on the edge of each structure term captures the importance of the structure/concept with respect to context. In this example, we show how the weighted context sub-graph for query "organizer focused retrieval workshop" from Figure 2 is obtained from the weighted term interpretations.

5. Finding query target and constraint

Given a query context graph of a query, we can proceed to find the target and constraint concepts for the query. Depending on the source query type, the process of finding the target and constraint concepts is different. If a query only contains content keywords, both concepts need to be identified from query context graph. Otherwise, if structural keywords are present in a query, they can be used as hints to selected target or constraint concepts.

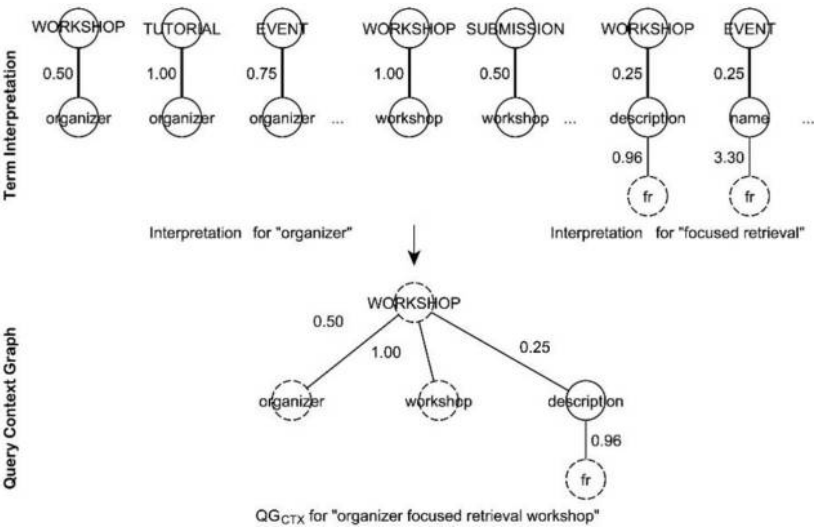

Figure 5.
Examples of query interpretation (with structure term) using query context graph

5.1 Keywords query with content keywords (QC)

In this section, we shall describe the case where keywords query contains only content keywords. The selection of the target concept and constraint concept for this query type is described in algorithm 1 *Select Target and Constraint (QC)*. This algorithm has two inputs, the query Q_K and its query context graph QG_{CTX}. The first step is to find the target of the query. As a query context graph is an abstraction of retrieval unit, taking the root of the graph (anologous to retrieval unit) as a target concept is reasonable. Hence, a target, TARGET is the root of QG_{CTX} (Case number 1 in Algorithm 1).

Algorithm 1. Select Target and Constraint (QC)

Require: keywords query Q_K, query context graph QG_{CTX}
Ensure: query interpretation, *QI*, consisting a set of target, *TARGET* a set of constraint, *CONSTRAINT*

▷ 1. Find Target Concept

$TARGET \leftarrow QG_{CTX_{root}}$

▷ 2. Select Best Concept as Contraint Concept

for all $qt_{content} \in Q_K$ **do**
  $CONSTRAINT_{qt_{content}} \leftarrow get\ BestConcept(qt_{content}, QG_{CTX})$
end for
$QI \leftarrow TARGET$
$QI \leftarrow$ CONSTRAINT

Now, for same query context graph, QG_{CTX}, we proceed to find the constraint concept for every term in the query. For each term, the best constraint concept is selected (Case number 2 in Algorithm 1). The constraint pair consisting of a concept and a content term will be inserted in the constraint list, CONSTRAINT. The target and constraint is returned as query interpretation in the form of QI = (TARGET, CONSTRAINT).

For a clearer picture, let us revisit the query from Figure 4, "andrew trotman focused retrieval". Its content term is $qt_{content}$ = {andrew trotman, focused retrieval} and its query context graph, $QG_{CTXWORKSHOP}$. We obtain the root, i.e. "workshop" as target. TARGET = {workshop}.

First, we look at the constraint for the first content term, "andrew trotman". For this term, there are two concept candidates, "organizer" and "name" with $score_{CW}$ "0.44" and "0.33". The best concept, with highest score is selected, giving us the first constraint, CONSTRAINT = {organizer:andrew trotman}.

Following this, we proceed to select the constraint concept for the second content term in a similar manner. This gives us the constraints set, CONSTRAINT = {organizer:andrew trotman, description:focused retrieval}. Both target and constraint concepts are returned as query interpretation, QI = {workshop}{organizer:andrew trotman, description:focused retrieval}.

5.2 Keywords query with content and structural keywords

In the case where a query contains both content and structural keywords, the structural keywords can serve two purposes, either as a target concept for the query or as a constraint concept describing a term. Hence, when these keywords are used, they need to be identified. Algorithm 2 *Selecting Target and Constraint* (QCAS) (an extension of Algorithm 1 Selecting Target and Constraint (QC)) is used to address this need. This algorithm has two inputs, the query, Q_K and its context sub-tree, QG_{CTX}. Similarly, we obtain a target concept based on the root of the context sub-tree. In this case, as a target

concept can be either specified by user in the query or suggested by system, its source status is noted (see case 1a in Algorithm 2) for prioritization selection later.

Next, we identify the constraint concept for content term in query (see case 2 in Algorithm 2). All possible concepts are stored for selection. The main difference between this and previous algorithm is we do not pick the best concept first, as we would like to check whether any of the possible concepts is mentioned in the query. If a concept matches any of the structural keyword specified a query, it is selected as the constraint concept for the term (see case 2a(i) in Algorithm 2). In the case where there is no match between any concept and structural keywords of query, the selection is based on the best concept (see 2a(ii). of Algorithm 2). The remaining ds from query are treated as possible targets of a query (see case 3 in Algorithm 2).

Let us illustrate this case with an example query, "organizer focused retrieval workshop" from Figure 5. Its content term is $qt_{content}$ = {focused retrieval}, structure term is $qt_{structure}$ = {organizer, workshop} and its query context graph, $QG_{CTXWORKSHOP}$. We obtain the root, i.e. "workshop" as target.

TARGET = {workshop}. Because "workshop" is specified in query, it is set as source from USER, giving us TARGET = {$worskhop_{ROOT}$: USER} (see case 1a(i) of Algorithm 2).

Similar to the previous example, we select the constraint of the content term, "focused retrieval". However, this time, we need to check the concept candidate, "description" against the structural keywords. Because the candidate is not specified in the query, it source is noted as SYS, indicating it is suggested by the system rather than user. This gives us the constraint, CONSTRAINT = {$description_{SYS}$: focused retrieval} (see case 2a(ii) of Algorithm 2).

Then, we check the remaining structural terms, i.e. "organizer" (see case 3 of Algorithm 2). The remaining terms are assigned as TARGET of the query. Because this term is specified by user, this gives us TARGET = {$worskhop_{ROOT}$: USER, $organizer_{USER}$}. Lastly, both target and constraint are returned as query interpretation, QI = {$worskhop_{ROOT}$: USER, $organizer_{USER}$} {$description_{SYS}$: focused retrieval}.

Algorithm 2. Selecting Target and Constraint (QCAS)

Require: keywords query Q_K, query context graph QG_{CTX}
Ensure: query interpretation, *QI*, consisting a set of target, *TARGET* and a set of constraint, *CONSTRAINT*

▷ 1. Find Target Concept

$TARGET_i \leftarrow QG_{CTX_{root}}$
$TARGET_{i_{status}} \leftarrow$ "*ROOT*"

▷ 1a. Check Target Concept Status
▷ 1a(i). Target by User

for all $qt_{structure} \in Q_K$ **do**
if match($qt_{structure}$, $TARGET_i$) **then**
$TARGET_{i_{source}} \leftarrow USER$
$SELECTED \leftarrow qt_{structure}$
remove $qt_{structure}$ from Q_K
else

▷ 1a(ii). Target by Sys

$TARGET_{i_{source}} \leftarrow SYS$
end if
end for
i + +;

▷ 2. Find Constraint Concept

for all $qt_{content} \in Q_K$ **do**
  $CPT_{qt_{content}} \leftarrow getConceptPerTerm(Q_K, QG_{CTX})$
  ▷ 2a. Check Constraint Concept Status
  for all $qt_{structure} \in Q_K$ **do**
    ▷ 2a(i). Constraint by User
    if match($qt_{structure}$, $CPT_{qt_{content}}$) **then**
      $CONSTRAINT_{jqt_{content}} \leftarrow qt_{structure}$
      $CONSTRAINT_{jqt_{content_{source}}} \leftarrow USER$
    $SELECTED \leftarrow qt_{structure}$
      remove $qt_{structure}$ from Q_K
    else
      ▷ 2a(ii). Constraint by Sys
      $CONSTRAINT_{jqt_{content}} \leftarrow getBestConcept(qt_{content}, CPT_{Q_K})$
      $CONSTRAINT_{jqt_{content_{source}}} \leftarrow SYS$
    end if
  end for
  j + +;
  end for
  ▷ 3. Assign Remain Structural Keywords
for all $qt_{structure} \in Q_U$ do
  if *not*(*SELECTED*)**then**
    $TARGET_i \leftarrow qt_{structure}$
    $TARGET_{i_{source}} \leftarrow USER$
    i + +;
  end if
end for
$QI \leftarrow TARGET$
$QI \leftarrow CONSTRAINT$

6. Evaluation setup

The experiment in this paper evaluates whether the proposed context-based term weighting approach, CTX+TW is able to select target and constraint concepts better compared to the baselines. For constraint selection, the evaluation focuses on two aspects, the ability to select correct constraint and the stability of CTW+TW implemented on different basic term weighting functions like TFIEF, BM25 and Language Model. For target selection, the evaluation focuses on the ability of CTX+TW in selecting correct target or subtarget. Lastly, the evaluation is also carried out based on different aspects such as complexity of information needs, query size and query with or without structural hints.

In this evaluation, the test suite consists of a text-centric data collection, a set of information needs in keywords queries form, a set of relevance assessment and performance metrics for the accuracy of concepts selection.

6.1 Data collection

Evaluations were performed using structured document under domains of conference. The SIGIR 2008-2010 Web Sites Collection (referred as SIGIR Web thereafter) consists of the structured version of the three years of conference site Web pages. It composes of Web pages in XML. This collection is text-centric, as it is developed from text contents, and not generated from database. It has a complex XML structure, and each article contains conference contents of different length. On average, an article contains 1,234 XML nodes. This collection itself has characteristics such as different complexities of

the document structures, diversities of its structure types and size of its elements/contents, therefore providing a diverse experimental setting for assessing our problem.

6.2 Topic set

The topics used in the evaluation are prepared in two manners, a synthetic set and a real user set. The synthetic set of topic is used to evaluate the efficiency of our algorithm such that various features of the algorithm can be justified. Nevertheless, for fairness of the evaluation, we have also included real user topic set to demonstrate the applicability of the algorithm on real information needs.

For synthetic topic set, the topics are created by the author such that it can be used to test the features of concepts selection. The topics cover variations in terms of information needs, like different query lengths, specific needs, general needs and so forth. The topics also cover functional tests with different information needs patterns.

For real user topic set, the topics are created by users who are familiar with Web search activity. The users were given a task to suggest topics based on a set of structured documents of the chosen collection. Although the collection is fixed, the users were encouraged to create topics that reflect possible queries that they would use during normal search routine. At the same time, users were also asked to suggest possible results entry points that they would like see as answers for the created topics. The results would be used for relevance assessment.

For both preparations of topics, the topics are further classified by its nature of specific or general.

6.2.1 Specific. A topic that requests for a particular or a list of known objects. For these topics, relevant answers can be single or multiple elements, normally in the form of objects or entities like tel, movie, url, add, etc. In these topics, users know what they are looking for and what answers they are expecting.

6.2.2 General. A topic that requests for information which is non-specific and general, covering a broader type of information. This topic normally results in more than one answer elements, whereby the returned element types can be of multiple types, e.g. given a topic asking for information about query representation in the domain of conference, the answers could be a workshop, a paper, a keynote, an abstract, etc. Most of the time, for this topic type, users will learn about the topic by browsing and going through the information returned.

Some examples of the topics for evaluation are shown in Table I.

Table I.
Some topics for SIGIR sites collection

Topic [Specific/General]	Description
room rate email river view [S]	I am looking for the room rate and email of River View Hotel
text processing summary presenter [S]	I want to find out who is the presenter and what is the summary of text processing tutorial
baeza-yates tutorial [S]	I am looking for tutorial presented by Baeza-Yates
33rd Annual ACM SIGIR Conference sponsors [S]	Who are the sponsors for 33rd Annual ACM SIGIR Conference
probabilistic models [G]	I am looking for information about probabilistic models
google industry track [G]	I am want to find out about Google's participate in industry track

6.3 Relevance assessment

Once the topics for evaluation are created, it is also necessary to have a set of assessment to judge the outcome of experiment carried out on these topics. At the algorithm level, we measure topics based on the generated structured query. For this, user is required to provide the golden standard, i.e. an equivalent structured query, for the topic he creates. To make it easier for user, we let the users suggest the structural information required using an interface, rather than asking them to write in the form of structured queries syntax. Web page interfaces (corresponding to their structures resources) are used to let user suggest possible focus points the correct information are located, i.e. Best Entry Point (BEP) that qualifies as the answer to his topic. BEP indicates where in a document that a user should start reading (Piwowarski *et al.*, 2008) (Reid *et al.*, 2006). For example, for topic "text processing summary presenter" in Table VI, the evaluator has selected the entry points that resolve to these elements, i.e. /article[1]/sigir2009[1]/full day tutorials[1]/summary[1], /article[1]/sigir2009[1]/full day tutorials[1]/presenter[1] and /article[1]/sigir2009[1]/full day tutorials[1]. Once the relevant entry points are known, we can obtain possible structures to generate structured queries for assessment. We will measure the structures accuracy of a query in terms of its entry concepts and term concepts.

6.3.1 Entry concept. As BEP refers to entry point of a particular physical element; at query level, it is more appropriate to generalize the entry point to the structure of an element, rather than referring to a specific element. We name this entry point as entry concept. Revisit the same topic, evaluator has specified that concepts "summary", "presenter", "full-day tutorials" are all appropriate as entry point for the topic. There can be more than one concepts for each topic. This is because elements in XML are nested and varied in sizes, thus it is common to encounter situations where the concepts of both parent and child elements are relevant, but to a different extent.

The accuracy of an entry concept is measured based on the concept coverage. Concept coverage evaluates whether the entry concept is structurally correct or otherwise. Here we adopt a similar scale used for measuring component coverage in standard structured retrieval evaluation (Manning *et al.*, 2008). The coverage can be classified into four types, to indicate different weights for different level of concepts.

(1) *Exact Coverage* (cov_{exact}): This concept contains exactly what the topic is seeking.

(2) *Too Broad* (cov_{broad}): This concept contains what the topic is seeking; however, it also contains other information.

(3) *Too Small* (cov_{small}): This concept contains what the topic is seeking, either partially, or not meaningful. E.g., an entry concept like /article/sigir2009/full day tutorials/presenter/last name for topic "text processing summary presenter" would be too small. This concept contains what the topic is seeking, either partially, or not meaningful, e.g., an entry concept like /presenter/last name for topic "text processing summary presenter" would be too small.

(4) *No Coverage* (cov_{no}): This concept does not contain what the topic is seeking.

6.3.2 Constraint concept. Besides entry concept, we also measure the correctness of the constraint concept of a content term. This concept filters a term to a specific structure type such that other irrelevant structures can be omitted in the retrieval. Hence, if a user

refines a term "andrew trotman" to the concept "author", other structures will not be considered during retrieval. The accuracy of a constraint concept can be classified into three categories as follows.

(1) *Not Relevant (rel_{not})*: This concept is not able to reflect the meaning of the content term.

(2) *Somehow Relevant ($rel_{somehow}$)*: This concept can somehow reflect the meaning of the content term.

(3) *Relevant (rel_{exact})*: This concept is able to reflect the meaning of the content term.

It is necessary to include $\text{rel}_{\text{somehow}}$ in the standard to handle a less rigid refinement of concepts used for a term. For example, for a topic where "title" is a relevant concept for a term "linguistic processing", a broader or less strict meaning like "paper" or "list of accepted papers" are also by some means relevant, and can be accepted as well.

6.4 Performance metrics

To assess the structured query generated by our query transformation framework, we measure the accuracy of its target concept and constraint concept using the following performance metrics.

6.4.1 Entry concept accuracy. For assessing target concept, we compare the concept of the structured query with the entry concept specified by users. Each entry concept is scored as follows:

$$C_{COV}(ec) = \begin{cases} 0 & \text{if ec} = \text{cov}_{\text{no}} \\ 0.5 & \text{if ec} = \text{cov}_{\text{small}} \\ 0.5 & \text{if ec} = \text{cov}_{\text{broad}} \\ 1 & \text{if ec} = \text{cov}_{\text{exact}} \end{cases}$$

To summarize the performance, a single-figure measure is used by taking the average of entry concepts for many topics. Given a topic, qi ϵ Q, ec_{ij} is the set of entry concepts obtained from topic i, then the average over Q is:

$$C_{COV_{AVG}}(Q) = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{n}\sum\nolimits_{j=1}^{n} C_{COV}(ec_{ij}), \textit{where } n \text{ is total } ec \text{ per } i$$

6.4.2 Constraint concept accuracy. The accuracy of constraint concept is measured by its relevancy to the term in the context of its topic. Each term's concept is scored as follows:

$$CREL(c) = \begin{cases} 0 & \text{if c} = \text{rel}_{\text{not}} \\ 0.5 & \text{if c} = \text{rel}_{\text{somehow}} \\ 1 & \text{if c} = \text{rel}_{\text{exact}} \end{cases}$$

Given a topic, q_i ϵ Q, t_{ij} is the set of content terms from topic i. c_{ij} is the first concept selected for content term t_{ij}. The average over Q is:

$$C_{REL_{AVG}}(Q) = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{n}\sum_{j=1}^{n} C_{REL}(c_{ij}), \textit{where } n \text{ is total } \textit{t}\text{erm per } i$$

Note that c_{ij} is the first ranked concept for t_{ij}. To analyze the ranked concepts, we use $C_{REL}(c_{topk})$ instead of $C_{REL}(c)$ where top k number of concepts are taken in account.

The scoring of top_k concepts is as follows:

$$C_{REL_{AVG}}(Q,k) = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{n}\sum_{j=1}^{n} C_{REL}(C_{topkij})$$

7. Results and discussion

In this section, the results of experiment in finding target concept and constraint concepts are presented. The context-based term weighting approach shall be referred as CTX+TW thereafter.

7.1 Interpreting constraint concepts

Table II shows the comparison of concepts found using CTX+TW approach, and its baseline, NCTX [12], for some tested topics. These concepts are assessed based on the benchmark concepts suggested by user. From the experiment, we find that CTX+TW is able to infer a better constraint concept for content term of a topic when a term has ambiguous concepts (e.g. T_2 and T_4). In such cases, our algorithm is able to suggest a relevant concept based on the context of the topic. Otherwise, for non-ambiguous term (e.g. T_1), the outcome of constraint concepts would be similar to the baseline.

If we looked at the term, "grand copthorne waterfront hotel" in T_1, it is only related to one concept type in this collection, which is a "name" of a "hotel". Hence, for this kind of term, it will always have the same constraint concept, as there is no ambiguous in its term usage. In fact, in this case, a filtering concept can even be omitted in the construction of structured query, as it does narrow down the scope of term, whereas, for the term "bruce croft", it has different concepts describing its role in different parts of collection. In this case, CTX+TW scopes down the constraint concepts to those relevant to a given topic based on the query context graph. Therefore, the term is filtered with a concept, "senior pc committee" when we issue a topic that looks for information about committee (T_2), while it is filtered with a concept, "responder" when we issue a topic that looks for information about industry (T_3).

NCTX is less accurate when a term has ambiguous concepts, as it does not capture different concept usages within the same collection. The selected concept is based on the

Table II. Some cases of constraint concepts interpretation for content term in query

Id	Topic	Content term	User	CTX+TW	NCTX
T_1	grand copthorne waterfront hotel address	grand copthorne waterfront hotel	hotel, name	name	location, name
T_2	bruce croft committee	bruce croft	senior_pc_committee	senior_pc_committee	bio, responder
T_3	bruce croft industry track	bruce croft	responder, bio	responder	bio, responder
T_4	presentation google	google	company	company	affiliation

highest rank, such as for a term "google" for topic T_4, it still ranks concept, "affiliation" higher rather than "company", because the former is a more frequent concept.

To measure the overall performance of CTX+TW in selecting constraint concepts, the average concept accuracy is taken based on a set of topics. Table III shows the results comparing CTX+TW approach over NCTX. For demonstrating the stability of our approach on different IR scoring models, we carry out the evaluation of three most used scoring models in the structured retrieval literature, i.e. TFIEF, Okapi BM25 and LM. Our result shows that CTX+TW is able to surpass its NCTX baseline in the overall accuracy as in Table III. From this result, we have the following observations.

- CTX+TW shows its stability on various scoring models as in Figure 2. If we look at top-1 concept, when stronger scoring models are used like Okapi BM25 and LM, CTX+TW improves in its overall accuracy, from 0.500 (for CTX+TW_{TFIEF}) to 0.841 (for CTX+ TW_{BM25}) and 0.886 (for CTX + TW_{LM}). This is due to element size normalization factor used in the latter retrieval models that emphasizes on direct term and concept relation such as <name>grand copthorne waterfront</name>, rather than indirect one like <description> […] […]. grand copthorne waterfront is located […] […].</description>.
- In addition to top-1 concept, we are also interested to find out whether constraint concepts at ranks 2 and 3 are relevant as shown in Table III, as it could be helpful to include these concepts in the situation where user interaction is allowed. Along with its baseline, CTX+TW is able to achieve better accuracy when top-2 and top-3 concepts are considered. The concept accuracies for all the scoring models are increased to 0.841 (for CTX + TW_{TFIEF}) and 0.955 (for both CTX + TW_{BM25} and CTX + TW_{LM}).

7.2 Interpreting target concepts

In the experiment of finding target concepts, we compare our interpretation method with a popular baseline, SLCA and an improved method of finding sub-tree, FCA (frequent common ancestor). From the experiment, we find that our query interpretation approach is able to find a better target concept in two situations. First, in a nested situation, for example, for topic T_4 in Table IV, the search term "trotman" is nested under multiple concepts like "workshops/workshop/organizers/organizer/name". When a seek content is located under such nested structures, the SLCA approach will return the nearest

Table III.
Constraint concept accuracy (C_{RELAVG}) based on top K SIGIR collection

Measure	$NCTX_{TFIEF}$	CTX + TW_{TFIEF}
Top-1	0.364	0.500
Top-2	0.636	0.591
Top-3	0.727	0.841
	$NCTX_{BM25}$	CTX + TW_{BM25}
Top-1	0.500	0.841
Top-2	0.727	0.955
Top-3	0.773	0.955
	$NCTX_{LM}$	CTX + TW_{LM}
Top-1	0.659	0.886
Top-2	0.682	0.955
Top-3	0.864	0.955

parents for all the terms, which gives us "organizers" for topic T_4. However, this concept is regarded as too small, as it will return less meaningful element.

What this topic is seeking is actually a type of concepts that can reflect the cooperation between these three persons, such as article, tutorial, workshop, etc. Hence, in this case, the preferred concept would be "workshop". Compared to SLCA, CTX+TW extends its targets selection to multiple levels of sub-tree, which enable us to obtain an addition target candidate, "workshop", which is structurally near to the query terms. For FCA, we can see that it has similar performance with CTX_TW, as it is selecting the common concept which appear more frequent than others. The advantage of CTX+TW over FCA is that more than one possible ancestors could be obtained, as long as the ancestors satisfy the context of query.

Second, when structural keywords are used in query, such as T_1 and T_2, our query interpretation algorithm can identify these keywords as target concepts. Using the SLCA approach, a target concept is the root of the SLCA sub tree, whereas our algorithm is able to handle a target concept that is contained within the sub-tree. For example, for topic T_1, the query is looking for address of a hotel. Using SLCA or FCA, we obtain a sub-tree rooted at "hotel". This sub-tree contains both content keyword, "grand copthorne waterfront", and structural keyword, "address". There is no measure to utilize structural keywords given in a query as target concepts. Our query interpretation approach addresses this limitation by introducing an algorithm that can suggest structural keywords used in query as target concepts within a sub-tree.

To measure the overall performance of CTX+TW in selecting target concepts, the average concept coverage is taken based on a set of topics. Table V shows the results comparing CTX+TW approach over SLCA and FCA. In this test, we measure how accurate is the best suggested target concept compared to its baselines. Two measures are used to evaluate the accuracy of target concepts when they are assessed under either loose or strict manner. For both measures, we can see that our approach has higher accuracy compare to its baselines.

Id	Topic	User	CTX+TW	SLCA	FCA
T_1	grand copthorne waterfront hotel address	hotel, address	address	hotel	hotel
T_2	grand copthorne waterfront deluxe room rate	rate, room rate	room rate	hotel	hotel
T_3	wei che huang andrew trotman	paper	paper	authors	paper
T_4	trotman geva kaamps	workshop	organizers, workshop	organizers	workshop

Table IV. Some cases of target concepts interpretation for query

Measure C_{COVAVG}	SLCA	FCA	CTX + TW
Loose	0.574	0.629	0.759
Exact	0.185	0.296	0.667

Table V. Target concept accuracy for SIGIR collection

7.3 The effect of query characteristics

Based on the evaluation setting, we further explore to see how query characteristics affect the accuracy of an interpreted query. We have made three observations from the result in Table VI.

7.3.1 Complexity of information needs. Query with specific information needs obtains better accuracy for both target and constraint concepts compare query with generic information needs. This is because the search intention is clearer in the former, such as "grand copthorne waterfront address" (T_1, Table III). When a specific query is given, there are more hints for query interpretation to find its target concept as well as constraint concept. This results in higher accuracy for specific query. Whereas when a generic query is given, there are often many possible suggested concepts. This increases the error rate as there may be non-relevant ones.

7.3.2 Usage of structural keywords. Query that uses both content and structural keywords (QCAS) gives better concept accuracy compare to query that uses content only keywords (QC) for both target and constraint concepts. The main reason is that when structural keywords are used in query, our query interpretation algorithm will be able to identify these keywords in the query, and used them in a more effective way as either target concept or constraint concept, whereas for query without structural keywords, the selection of target concept or constraint concept is based on the query context, which may results in incorrect concepts selection.

7.3.3 Size of query. A longer query gives better description of the query context, thus giving better concept accuracy compare to a shorter one. However, in this evaluation, we have only tested up to four query terms (each term can have more than one word). We have not yet tested query with terms longer than four.

8. Conclusions

In this paper, we studied the problem of finding target and constraint concept for constructing XML query. To improve the identification of these concepts in the situation where collection has higher structural complexities, we present a context-based term weighting model to allow weighting of terms under within contextual parts of documents. A query context graph and algorithms [*Select Target and Constraint (QC)* and *Select Target and Constraint (QCAS)*] are proposed to identify target/subtarget and

Table VI.
Query characteristic on query interpretation performance

Query characteristic	Target concept accuracy, C_{COVAVG}	Constraint concept accuracy, C_{RELAVG}
Complexity of information needs		
General	0.500	0.333
Specific	0.868	0.894
Usage of structural keywords		
Without	0.500	0.500
With	0.912	0.938
Size of query		
1 term	0.400	0.333
2 terms	0.722	0.875
$>$ 3 terms	0.923	0.929

constraint for cases like keywords query with content keywords, keywords query with content and structural keywords. For performance measurement, a test suite is devised for evaluating the outcome of concepts selection.

References

Bao, Z., Lu, J., Ling, T.W. and Chen, B. (2010), "Towards an effective xml keyword search", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 22 No. 8, pp. 1077-1092.

Boag, S., Berglund, A., Chamberlin, D., Sime´on, J., Kay, M., Robie, J. and Fernandez, M.F. (2007), "XML path language (XPath) 2.0", W3C recommendation, W3C, January, available at: www.w3.org/TR/2007/REC-xpath20-20070123/

Bray, T., Paoli, J. and Sperberg-McQueen, C.M. (1997), "Extensible markup language (xml)", *World Wide Web Journal*, Vol. 2 No. 4, pp. 27-66.

Carmel, D., Maarek, Y., Mandelbrod, M., Mass, Y. and Soffer, A. (2003), "Searching xml documents via xml fragments", *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, NY*, pp. 151-158.

Chamberlin, D.D. (2002), "Xquery: an xml query language", *IBM Systems Journal*, Vol. 41 No. 4, pp. 597-615.

Cohen, S., Mamou, J., Kanza, Y. and Sagiv, Y. (2003), "Xsearch: a semantic search engine for xml", *Proceedings of the 29th International Conference on Very Large Data Bases - Volume 29, VLDB '03, VLDB Endowment*, pp. 45-56.

Fallside, D.C. and Walmsley, P. (2004), "Xml schema part 0: primer second edition", W3C Recommendation, October.

Gan, K.H. and Phang, K.K. (2014a), "A query transformation framework for automated structured query construction", *Journal of Information Science*, Vol. 40 No. 2, pp. 249-263.

Gan, K.H. and Phang, K.K. (2014b), *An Intermediate Query Model for Structured Retrieval's Queries Construction*, iiWAS, Hanoi, pp. 289-295, 4-6 December.

Graupmann, J., Biwer, M., Zimmer, C., Zimmer, P., Bender, M., Theobald, M. and Weikum, G. (2004), "Compass: a concept-based web search engine for html, xml, and deep web data", *Proceedings of the 30th International Conference on Very large Data Bases, VLDB Endowment*, Vol. 30, pp. 1313-1316.

Hsu, W., Lee, M.L. and Wu, X. (2004), "Path-augmented keyword search for xml documents", *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence, ICTAI '04, IEEE Computer Society, Washington, DC*, pp. 526-530.

Huffman, S.B. and Baudin, C. (1997), "Toward structured retrieval in semi-structured information spaces", *IJCAI (1)*, Morgan Kaufmann, Burlington, pp. 751-757.

Itakura, K.Y. and Clarke, C.L.A. (2010), "A framework for bm25f-based xml retrieval", in Crestani, F., Marchand-Maillet, S., Chen, H.H., Efthimiadis, E.N. and Savoy, J. (Eds), *SIGIR*, ACM, New York, NY, pp. 843-844.

Kim, J.Y., Xue, X.B. and Croft, W.B. (2009), *A Probabilistic Retrieval Model for Semi Structured Data*, ECIR, MA, pp. 228-239.

Li, J., Liu, C., Zhou, R. and Ning, B. (2009), "Processing xml keyword search by constructing effective structured queries", in Li, Q., Feng, L., Pei, J., Wang, X.S., Zhou, X. and Zhu, Q.M. (Eds), *APWeb/WAIM, Vol. 5446 of Lecture Notes in Computer Science*, Springer, New York, NY, pp. 88-99.

Li, R. and van der Weide, T.P. (2009), "Language models for xml element retrieval", in Geva, S., Kamps, J. and Trotman, A. (Eds), *INEX, Vol. 6203 of Lecture Notes in Computer Science*, Springer, New York, NY, pp. 95-102.

Liu, Z., Walker, J. and Chen, Y. (2007), "Xseek: a semantic xml search engine using keywords", in Koch, C., Gehrke, J., Garofalakis, M.N., Srivastava, D., Aberer, K., Deshpande, A., Florescu, D., Chan, C.Y., Ganti, V., Kanne, C.C., Klas, W. and Neuhold, E.J. (Eds), *VLDB*, ACM, New York, NY, pp. 1330-1333.

Manning, C.D., Raghavan, P. and Schu¨tze, H. (2008), *Introduction to Information Retrieval*, Cambridge University Press, Cambridge.

Ogilvie, P. and Callan, J. (2002), "Language models and structured document retrieval", in Fuhr, N., Go¨vert, N., Kazai, G. and Lalmas, M. (Eds), *INEX Workshop*, ERCIM, Dagstuhl, Germany, pp. 33-40.

Petkova, D., Croft, W.B. and Diao, Y. (2009), "Refining keyword queries for xml retrieval by combining content and structure", *ECIR*, Toulouse, pp. 662-669.

Piwowarski, B., Trotman, A. and Lalmas, M. (2008), "Sound and complete relevance assessment for xml retrieval", *ACM Transactions on Information Systems*, Vol. 27 No. 1, pp. 1:1-1:37.

Reid, J., Lalmas, M., Finesilver, K. and Hertzum, M. (2006), "Best entry points for structured document retrieval - Part I: characteristics", *Information Processing and Management*, Vol. 42 No. 1, pp. 74-88.

Robertson, S. and Zaragoza, H. (2009), "The probabilistic relevance framework: Bm25 and beyond", *Foundations and Trends in Information Retrieval*, Vol. 3 No. 4, pp. 333-389.

Salton, G. and Buckley, C. (1988), "Term-weighting approaches in automatic text retrieval", *Information Processing and Management*, Vol. 24 No. 5, pp. 513-523.

Taha, K. and Elmasri, R. (2010), "Xcdsearch: an xml context-driven search engine", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 22 No. 12, pp. 1781-1796.

Tannier, X. (2005), "From natural language to nexi, an interface for inex 2005 queries", *INEX*, *Dagstuhl Castle*, pp. 373-387.

Theobald, M., Bast, H., Majumdar, D., Schenkel, R. and Weikum, G. (2008), "Topx: efficient and versatile top-k query processing for semistructured data", *The VLDB Journal*, Vol. 17 No. 1, pp. 81-115.

Trotman, A. (2009), "Narrowed extended xpath i", in Liu, L. and zsu, M.T.O¨ (Eds), *Encyclopedia of Database Systems*, Springer, New York, NY, pp. 1876-1880.

Wang, Q., Li, Q. and Wang, S. (2007), "Preliminary work on xml retrieval", *INEX*, *Dagstuhl Castle, pp. 70-76*.

Corresponding author
Keng Hoon Gan can be contacted at: khgan@usm.my