

# XReal: An Interactive XML Keyword Searching

Zhifeng Bao  
School of Computing  
National University of  
Singapore  
baozhife@comp.nus.edu.sg

Jiaheng Lu  
School of Information and  
DEKE, MOE  
Renmin University of China  
jiahengl@gmail.com

Tok Wang Ling  
School of Computing  
National University of  
Singapore  
lingtw@comp.nus.edu.sg

## ABSTRACT

Keyword search over XML data usually brings irrelevant results especially when the keywords in a user query have ambiguities. We demonstrate a statistic-based approach to identify the search targets and constraints of a user query in the presence of keyword ambiguities, and come out a relevance oriented result ranking scheme called XML TF\*IDF. Since the search intention of a same query may even vary from user to user, we provide an interactive search strategy by allowing user to simply tick their desired search targets from a list of suggestions recommended by the search engine. In this way, we can acquire more precise results and also take the burden of learning the schema of XML data off users.

## Categories and Subject Descriptors

H.3.3 [Information Systems]: Information Search and Retrieval

## General Terms

Management, Experimentation

## 1. INTRODUCTION

In this paper, we would like to demonstrate how to build an effective XML keyword search engine. Specifically, we aim to resolve the following issues that are unique to XML keyword search.

- (1) Identify the target that a user query intends to search for. As compared to the traditional IR-style web search whose search target is certainly flat document, the search target of an XML keyword query is usually implicit or unknown.
- (2) Identify and quantify the possibility of potentially various search constraints of a user query.
- (3) Rank query results in consideration of their relevance scores with the query and their structural features.

Unfortunately, recent literatures focus on designing either the result matching semantics by enforcing the occurrences of each query keyword in a subtree as compact as possible, or the result ranking scheme based on a certain matching semantics. However, regardless of the validity of matching semantics itself, without figuring out the search target of a user query, the matching results associated with different search targets are messed up together, which badly annoys user in result consumption.

We take a widely adopted matching semantics called smallest lowest common ancestor (SLCA) [2] as example to illustrate the importance of search target identification. Each SLCA result of a

keyword query is a subtree containing all query keywords but has no subtree which also contains all the keywords.

EXAMPLE 1. Consider  $Q = \text{"customer, interest, art"}$  issued on the bookstore data in Figure 1, most likely intending to find customers who are interested in art. The SLCA results can be classified in four types: (1) the customer interested in art (e.g. customer C2, C4), (2) the customer whose name contains "art" and has an interest (e.g. C3), (3) the customer whose address contains "art" and has an interest, (4) the book whose title contains all keywords (book B1). SLCA neither distinguishes the search target i.e. customer or book, nor distinguishes the above four search constraints. □

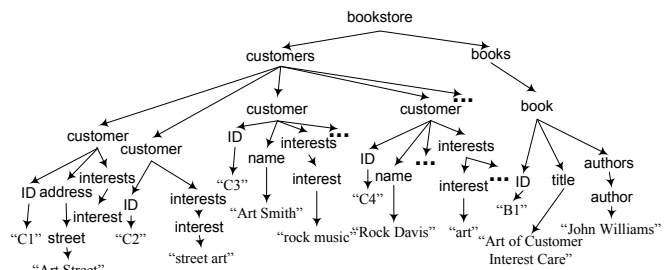


Figure 1: bookstore data with customer and book info

Another problem which is not studied by any existing work is the *keyword ambiguity problem*. Since XML data contains both the structural and content information, XML keyword queries usually contain various ambiguities: (1) A keyword can appear both as an XML tag name and as a text value of some other nodes. (2) A keyword can appear as the text values of different types of XML nodes and carry different meanings. (3) A keyword can appear as an XML tag name in different contexts and carry different meanings.

The keyword ambiguity problem may lead to various interpretations of the search target and constraint. E.g. in Figure 1, *customer* and *interest* appear as both an XML tag name and a text value; *art* appears as a text value of interest, address and name node. It is such ambiguity that causes two interpretations of search target and four interpretations of search intention of  $Q$  as mentioned in Example 1. Therefore, we need to find the potential interpretations, and also quantify their respective confidence to be the desired search target (constraint).

## 2. TECHNIQUES IN XREAL

In order to resolve the above challenges in the presence of keyword ambiguities, we present our XML keyword search prototype XReal. In particular, we distinguish the type of a node in XML data by its prefix path from root node, and the search target is referred as node type. Then we devise a series of novel statistic terms, such as XML document frequency (XML DF) and XML term frequency (XML TF) for a certain node type  $T$ . Next, we propose a series of

guidelines to capture human intuitions for the job of measuring the confidence of a certain node type  $T$  as the desired search target of a query  $Q$ . After the desired search target  $T$  is fixed, we compose the result as a subtree rooted at  $T$ . Lastly, we propose a novel  $XML\ TF*IDF$  result ranking scheme, which not only inherits the objective relevance nature between user query and matching results as done in traditional  $TF*IDF$ , but also captures the confidence of a node  $n$  to be searched via (as a constraint) and the structural relationship of nodes. Readers can refer to [1] for detailed formulae and rationale behind. An example is given to illustrate how XReal infers user's desired result and puts it as a top-ranked answer.

**EXAMPLE 2.** Recall the query in Example 1. XReal interprets that customer is the 1<sup>st</sup> desired search target, as all three keywords have high frequency of occurrences in customer nodes; while book is the 2<sup>nd</sup> desired target. Similarly, since keywords "interest" and "art" have high frequency of occurrences in subtrees rooted at interest nodes, it is considered with high confidence that this query wants to search via interest nodes, and we incorporate this confidence into our ranking formula. Besides, customers interested in "art" should be ranked before those interested in (say) "street art". Thus, C4 is ranked before C2, and further before customer with address in "art street" (e.g. C1) or named "art" (e.g. C3). □

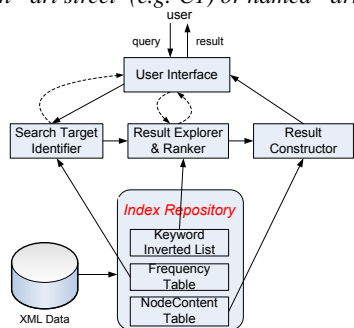


Figure 2: System Architecture of XReal

### 3. SYSTEM OVERVIEW

As shown in Figure 2, XReal system consists of four core parts.

- **Index Repository.** In order to improve the efficiency of query answering, we build three major types of indices. (1) keyword inverted lists, each of which records a list of dewey labels of nodes that directly contain a certain keyword  $k$ , together with the node type and some associated statistics data (for result ranking use later). (2) NodeContent table, which builds a mapping from the dewey label of a node  $n$  to the content associated with  $n$  for result display. (3) Frequency table  $F$ , which stores the XML term frequency for each combination of a keyword  $k$  and a node type  $T$ . The indices are built offline, and the details can be found in [1].
- **Search target identifier,** which is responsible for measuring the confidence of all node types in XML data to be the desired search target of a user query  $Q$ , and offers the most promising search target candidates for user to choose.
- **Result explorer and ranker,** which finds the matching result  $r$  under the search target ticked by user in last step, and meanwhile computes the relevance of each  $r$  w.r.t  $Q$ .
- **Result Constructor,** which constructs the final result for display and user consumption.

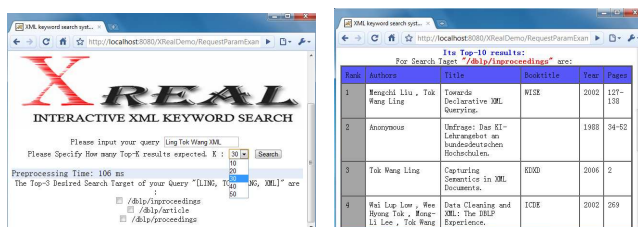
### 4. DEMONSTRATION

In this demo, we would like to show two major features of XReal.

*First,* we aim at showing an important yet unstudied topic in XML keyword search: resolve the keyword ambiguity problem in

identifying the search target and search constraint of a user query. We will illustrate how XReal adopts a heuristic and interactive way to guide user to tell the search engine his potential search intention through a series of live interactions, without requiring user to learn any query language or schema of XML data. When a user issues a query to XReal, our search target identifier first computes the confidence of each node type in XML data as the desired search target, and returns the top- $n$  targets ( $n$  is user-specified and is 3 by default). User can choose his favored target (by just ticking the checkboxes associated with the suggested targets), and specify how many top- $k$  results expected (as shown in Figure 3(a)). Then, XReal will keep interacting with user whenever any ambiguity is encountered in judging the search constraints, e.g. whether a keyword can be the value of different node types. If so, it returns the possible interpretations of search constraints of the query for user to choose again. Through the above two-step interaction (as denoted by the dashed arrows in Figure 2), both the keyword ambiguity in resolving the search target and constraint are cleared off, and the final results are computed and ranked for final display (as shown in Figure 3(b)). We believe our interaction design provides user a much improved search experience and helps minimize the user efforts in result consumption, as compared to mixing and returning the individual results of different search intentions as a whole.

Note that, for novice user who is unwilling to participate the interactions, XReal will automatically choose the most promising search target and search constraint (according to its confidence measurement schemes) to conduct the query answering in a one-stop service.



(a) Search Target Identifier (b) Result Explorer & Ranker

Figure 3: Snapshot of Search Target Identifier

*Second,* we may show some advanced search features for expert user to investigate the rationale and importance of various ranking factors designed in our  $XML\ TF*IDF$  ranking scheme [1] in depth. Specifically, XReal will offer user a list of important ranking factors for him to specify those that compose the overall ranking scheme. User can compare the relevance of results returned by various combination of these ranking factors. DLBP (500MB), a large-scaled real-life data set, is used throughout the demo.

In a nutshell, this work exploits purely the statistics of underlying XML database to address search intention identification, result retrieval and relevance oriented ranking as a single problem for XML keyword search, without relying on any schema information of XML data such as DTD or XML Schema.

### 5. ACKNOWLEDGMENTS

Jiaheng Lu was partially supported by 863 National High-Tech Research Plan of China (No: 2009AA01Z133), IBM Research - China Fund(Project No. JSA200909010) and SRF for ROCS, SEM.

### 6. REFERENCES

[1] Z. Bao, T. W. Ling, B. Chen, and J. Lu. Effective XML keyword search with relevance oriented ranking. In *ICDE*, 2009.  
 [2] Y. Xu and Y. Papakonstantinou. Efficient keyword search for smallest LCAs in XML databases. In *SIGMOD*, 2005.