Relevant Answers for XML Keyword Search: A Skyline Approach

Khanh Nguyen and Jinli Cao

Department of Computer Science and Computer Engineering
La Trobe University, Melbourne, Australia
{tuan.nguyen,j.cao}@latrobe.edu.au

Abstract. Identifying relevant results is a key task in XML keyword search (XKS). Although many approaches have been proposed for this task, effectively identifying results for XKS is still an open problem. In this paper, we propose a novel approach for identifying relevant results for XKS by adopting the concept of Mutual Information and skyline semantics. Specifically, we introduce a measurement to effectively quantify the relevance of a candidate by using the concept of Mutual Information and provide an effective mechanism to identify the most relevant results amongst a large number of candidates by using skyline semantics. Extensive experimental studies show that in overall our approach is more effective than existing approaches and can identify relevant results and top k results in acceptable computational costs.

1 Introduction

XML is rapidly emerging as a standard for representing, publishing and exchanging data over the Internet. With the great success of keyword search over flat documents, keyword search over XML data has recently attracted lots of attentions of researchers from both database and information retrieval. Keyword search provides a friendly mechanism to access XML data without requiring the knowledge of the structured query languages and possibly complex data schemas. However, the limited expressiveness and the ambiguity of keyword queries cause identifying relevant results a very challenging task of XML keyword search.

A candidate of keyword search over XML databases is a subtree covering all query keywords. The baseline approach uses Lowest Common Ancestor (LCA) semantics from graph theory [2] to identify the result of a given keyword query. This approach returns all candidates, thus it has high recall but very low precision. Recently, many proposals [15,9,7,11,12] have been made to boost precision of the baseline approach. The common ideas of these work are (i) *Relevant evaluation*: defining heuristic-based rules that a relevant result has to satisfy; (ii) *Pruning*: eliminating all LCA nodes which do not satisfy the defined rules. It has been experimentally proved by [13] that these approaches not only miss relevant results but also return irrelevant results.

To improve the quality of results for keyword search, we need to deal with all of the following requirements: (R_1) effectively measuring relevant degree of a

L. Chen, P. Triantafillou, and T. Suel (Eds.): WISE 2010, LNCS 6488, pp. 216–224, 2010.

candidate; (R_2) providing an effective mechanism to identify the most relevant results amongst a large number of candidates.

In this paper, we investigate the challenging problems for fulfilling aforementioned requirements. Specifically, we introduce a measurement to quantify the relevance of a candidate by using the concept of Mutual Information for fulfilling requirement (R_1) . The requirement (R_2) is solved by using skyline semantics [4,5] which is proven as an effective mechanism to select the most relevant results (skyline answers).

Mutual Information is a central concept of information theory [8]. It is a quantitative measure of the dependency of two random variables. In other words, the Mutual Information of X and Y measures how much information X can tell us about Y and vice versa. In the context of an XML tree we see that the more information two nodes u and v can tell about each other, the more meaningful relationship between them is holding. More generally, the more information each node in a subtree tells about other nodes, the more relevancy the node associates to the query. From that observation, we adapt the *Mutual Information* to measure the relevancy degree of a candidate answer in this paper.

Skyline query can provide a set of relevant answers, even though those answers may not be the satisfactory ones in all criteria. Skyline queries have been well studied over relational databases [3,6,1,14]. However, applying skyline queries into the context of XML keyword search has not received many attentions yet. In this paper, we propose a novel approach for identifying results of interest for XKS using skyline semantics. We also introduce three different ranking criteria, based on the dominance relationship of skylines to retrieve the top k results. The contributions of this paper are described as follows. The contributions of this paper are described as follows.

- Proposed a novel approach to evaluate the relevancy degree of a candidate answer using the concept of Mutual Information from information theory.
- Introduced an approach to identify most relevant results amongst a large number of candidates using skyline semantics.
- Conducted extensive experiments on real data sets to prove the effectiveness and efficiency of our algorithms.

The remainder of this paper is organized as follows. Section 2 presents some preliminaries. In section 3, we briefly introduce Mutual Information (MI) and its related concepts. Then, we propose normalized MI to quantify the relevant degree of a candidate. Our approach for identifying relevant results using skyline semantics is presented in section 4. Algorithms are developed in section 5. Section 6 discusses experimental results and finally, conclusions are given in section 7.

2 Preliminaries

2.1 Data Model and Query

An XML database is modeled as a rooted labeled tree $T = (r, V, E, L, C, D)$, where V is the set of nodes, $r \in V$ is the root, E is the set of parent-child

edges between nodes in V, $C \subset V$ is a subset of the leaf nodes of the tree called content nodes, L assigns a label to each member of $V \setminus C$, and D assigns a data value (e.g., a string) to each content node. We assume no node has both leaf and non-leaf children, and each node has at most one leaf child. Each subtree $S = (r', V', E', L', C', D')$ of T is a tree such that $V' \subseteq V, E' \subseteq E, L' \subseteq L$, and $C' \subseteq C$.

A keyword query Q is a sequence $w_1, \ldots w_n$ of words. A subtree S is a *candidate answer* to Q if its content nodes contain at least one instance of each keyword in Q. If there is more than one subtree of S containing the same instances of search keywords, we only choose the smallest subtree.

2.2 Related Work

In this section, we will discuss related work of XML keyword search. The concept of Mutual Information and skylines will be also briefly introduced.

Result Identification. The baseline algorithm returns all candidate answers as the result. This approach has perfect recall but very low precision. Recently, several attempts as Smallest LCA (SLCA) [15], Exclusive LCA(ELCA) [9], XSEarch [7], Compact Valuable LCA (CVLCA) [11] and Meaningful LCA (MLCA) [12] have been made to boost the precision of the baseline approach. The common idea of these approaches is to evaluate the relevance of a candidate in a boolean way. It means that a candidate either is evaluated as a relevant candidate or is not at all depending on whether it satisfies a set of pre-defined heuristics rules. However, given the ambiguity of keyword query in terms of search intentions, it is difficult (sometimes impossible) to exactly conclude a candidate as a relevant answer or otherwise. To more effectively measure the correlation between content of nodes in a subtree, we adopt the concept of *Mutual Information* from information theory [8] which has been widely used in mining the meaningful correlation between attributes in a relation. The details will be introduced in next section.

Skylines. Skyline queries have received a lot of attentions over the recent years, and several algorithms have been proposed [3,6,1,14,10]. Given a set of points in a d-dimension space. The skyline is defined as the subset containing those points that are not dominated by any other point, whereas a point p dominates p' if p is better than or equal to p' in all the dimensions and strictly better in at least one dimension. Thus, the best answer for such query exists in the skyline.

BNL [3], SFS [6] and SaLSa [6] are generic, in the sense that they do not require any specialized access structure to compute the skyline and can therefore be applied even when the points are the results of some other operations. Other works [14,10] rely on the existence of appropriate indexes, such as B^+-tree or R-tree to speed-up skyline computations. Note that these approaches only apply on static data, where the over-head for building the indexes is amortized across multiple queries. In our setting, the underlying data (candidates) are depended on the query. In this case, building indexes at query time is very expensive, thus it is not suitable.

3 Normalized Mutual Information

In this section, we review the concept of Mutual Information (MI) and its related concepts, and then make it applicable in measuring the meaningful relationship between two nodes.

Entropy and *Mutual Information (MI)* are two central concepts in information theory [8]. Entropy is a measure of the uncertainty of a random variable, while MI quantifies the mutual dependence of two random variables.

Definition 1 (Entropy). *Let X be a discrete random variable that takes on values from the set $\mathcal{X}$ with a probability distribution function $p(x)$. The entropy of X is defined as*

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x)$$

Definition 2 (Mutual Information). *Mutual information of two random variables is a quantity that measures the mutual dependence of the two variables. Given two discrete random variables X and Y, their mutual information can be defined as:*

$$I(X;Y) = \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$

where $p(x,y)$ is is the joint probability distribution function of X and Y, and $p(x)$ and $p(y)$ are the marginal probability distribution functions of X and Y respectively.

Property 1 *$I(X;Y) \leq H(X)$ and $I(X;Y) \leq H(Y)$.*

Property 1 indicates that the MI of two nodes is bounded by the minimum of their entropy. The proof of this property can be found in [8]. Since the entropy of different nodes varies greatly, the value of MI also varies from different pairs of nodes. To make MI a good measure to quantify the closely relativeness of two nodes in a candidate, we require the MI of two nodes independent from their entropy. For this purpose, we propose normalized MI as follows.

Definition 3 (Normalized Mutual Information). *The Normalized Mutual Information (NMI) of two random variables X and Y is defined as:*

$$\widetilde{I}(X;Y) = \frac{I(X;Y)}{max\{H(X), H(Y)\}}$$

In next section, we will adopt this concept to measure the relationship between two nodes in a subtree. Then, skyline semantics is used to identify a set of relevant results.

4 Identifying Relevant Results Using Skyline Semantics

Let S be a candidate answer of Q in XML database T, we measure its relevance by calculating the NMI of every pair of content nodes. The question is how we can identify the candidate S relevant to Q. Normally, an aggregation function (*i.e., sum, average*) can be obtained to get the total NMI of all content nodes in S as $score(S) = \sum_{c_i,c_j \in C_S} \widetilde{I}(c_i; c_j)$, where C_S is a set of content nodes in candidate S. The relevance of S can be decided based on a pre-defined threshold α. For instance, if $score(S) \geq \alpha$, S can be considered as a desired result; otherwise it is an irrelevant result. However, selecting a suitable threshold α is not easy, because it is varied from query to query. A low threshold causes returning many answers (including less or not desired ones). In contrast, a high threshold may miss some desired answers. Even though we can let user select the threshold at query time, this is not a good option because users may need to query different times with different thresholds to get their desired results.

We apply skyline semantics which is an effective approach to select the most desired answers amongst numerous candidates. For every keyword $w_i \in Q$, we calculate the set $M_i = \{m | m$ is a leaf node in candidate T of Q and m contains $w_i\}$. The NMI of two keywords w_i and w_j in a candidate subtree S is defined as $max\{\widetilde{I}(m_i; m_j)\}$, where $m_i \in M_i$ and $m_j \in M_j$. Given a keyword query $Q = \{w_1, \ldots, w_n\}$ we measure the NMI of each pair of keywords in a candidate subtree S and store them in vector $D_S = [\widetilde{I_k}(w_i; w_j) | w_i, w_j \in Q \wedge (i < j)]$, where $k = 1, \ldots, C_n^2$ while n is the number of keywords in Q. To choose the most relevant result set to Q, we apply skyline semantics over the candidate set. The vector D_S plays a role as skyline dimensions of the candidate S. Because the high MNI of two nodes indicates their high relativeness, we refer those candidates with high values in their skyline dimensions. More formally, we define the *dominance* relationship between candidates in our context as follows.

Definition 4 (Dominance). *Let S and S' are two candidate answers of Q over an XML database T. S' dominates S, denoted as $S' \succ S$ if,*

- $\forall i (1 \leq i \leq d) D_S[i] \leq D_{S'}[i]$,
- *and* $\exists j (1 \leq j \leq d) D_S[j] < D_{S'}[j]$

where d is where d is the number of values in vector D_S and $d = C_n^2$. The $D_S[i]$ is the i-th value in D_S.

In words, $S' \succ S$ means that the relationship between every pair of keywords in S' is at least as meaningful as the relationship between the corresponding pairs in S. Consequently, S' is more relevant than S is to query Q if $S' \succ S$. Therefore, the problem of identifying relevant results of Q becomes finding a set of non-dominated results by adapting the skyline semantics.

Definition 5 (Relevant Results). *Given a keyword query Q and an XML database T. R is a set of relevant results for query Q over T if for each $S \in R$, there does not exist any other candidate S' of Q that S' dominates S.*

5 Algorithms

In this section, we introduce the algorithms for identifying relevant results based on skyline semantics To accelerate the query processing time, indexes are built off-line at the time we parse the XML database. The efficiency of the algorithms will be analyzed in details in next section.

5.1 Candidate Generation

Generating the candidates is the first step of XKS. The efficient computation has been well study in previous literature [15,16,17]. In this paper, we adapt the algorithm proposed in [17] which is experimentally proved to be the fastest one. Due to the limited space, the details of the algorithm is omitted here. The difference of our approach from others is that we concurrently measure the Normalized Mutual Information (NMI) between each pair of keywords in every candidate during the generating of candidates. The resultant candidates are stored in a sorted list by values of corresponding NMI vectors.

5.2 Skyline Answers

We integrate the skyline computation into the process of candidate generation. More specifically, at the time of generating a candidate, we also check whether it is a skyline answer. By doing this, the skyline computation is simplified to be a

Algorithm 1. *Skyline Answers*

Input: keyword query Q, XML databases T.
Output: a set of skyline answers $\mathcal{R}$.

```
1: R = ∅;
2: while there are more candidates of query Q in T do
3:    nextCan = select a candidate;
4:    isDominated = false;
5:    for for each (R ∈ R) do
6:       if R ≻ nextCan then
7:          isDominated = true;
8:       else
9:          if nextCan ≻ R then
10:            remove R from R;
11:         end if
12:      end if
13:   end for
14:   if isDominated = false then
15:      R ← insert nextCan;
16:   end if
17: end while
18: return R
```

dominance check procedure(as shown by Algorithm 1). The algorithm works as follows. *(i) Initialization (line 1)*: the result set is set to empty set. *(ii) Repeatedly generating a new candidate (line 2)*: it can be adapted from [17] with some minor modifications. We omit the details here due to space limit. *(iii) Dominance check (lines 3 - 13)*: for each new generated candidate, we apply skyline semantics to see whether it is a relevant result. *(iv) Updating results (line 14-16)*: if the new candidate is not dominated by any candidates so far, it is added to the result set.

6 Experimental Analysis

The experiments were conducted on a 3.2GHz P4 CPU running Windows XP Professional with 1GB of RAM. The algorithms were implemented in Java. We used Oracle Berkeley DB[1] as a tool for creating indexes. We have tested on two real data sets, including: DBLP[2] and IMDB[3].

6.1 Result Quality

We compare the search quality of our relevant results identification using skyline semantics (*referred as SkyAns from here*) with other state-of-art approaches (i.e., SLCA [15], XSEarch [7], CVLCA) [11] and MLCA [12]) The quality is measured in three popular metrics in IR literature, including *Precision (P), Recall (R) and F-measure*. The *F-measure* shows the trade-off between precision and recall and is computed as:

$$F - measure = \frac{(1+\beta^2)PR}{\beta^2 P + R}$$

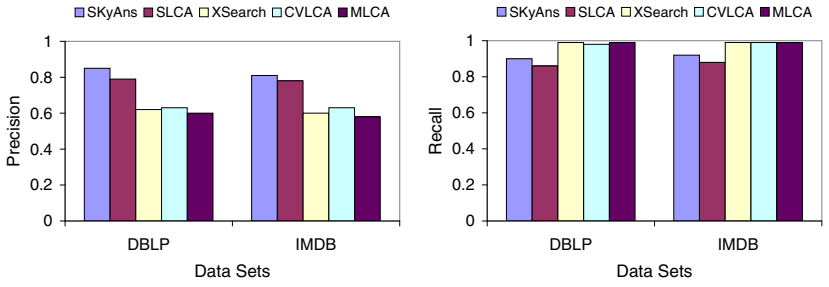
where $\beta = 1$ weights precision and recall equally; $\beta < 1$ emphasizes precision, while $\beta > 1$ focuses on recall.

We tested 20 queries on each data set. The correct answers for those queries are obtained by running the corresponding schema-aware XQuery, and the correctness of the answers is verified manually. We recorded the precision and recall for each query and take the average as the precision and recall on each data set. The results are summarized in Fig. 1. The results show that our approach is more effective than all other counterparts. However, The recall is lightly lower due to strict semantics of skylines. To further evaluate the overall of result quality, we take F-measure with different values of β (see Fig. 3). The result from Fig. 3 indicates the overall quality of our approach is higher than all their counterparts.

6.2 Computational Costs

The computational cost is tested on both extracted data sets with size of 200 MB. For each data set, we test a set of 10 queries with the average of 5 keywords.

[1] http://www.oracle.com/technology/products/berkeley-db/index.html
[2] http://dblp.uni-trier.de/xml/
[3] http://www.imdb.com/



(a) Precision **(b)** Recall

Fig. 1. Result quality

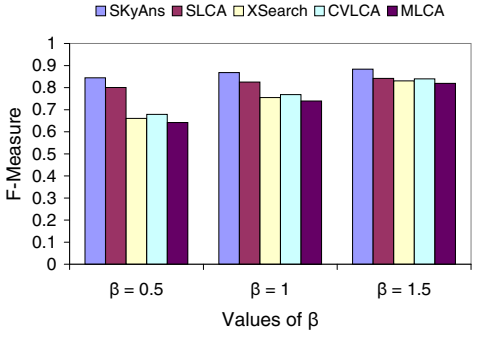


Fig. 2. Overall of result quality

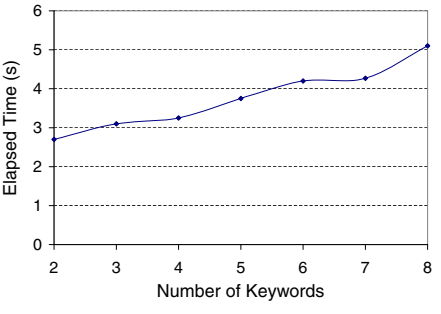


Fig. 3. Computational costs

The response time is average time of the corresponding 10 queries, as shown in Fig. 3. The result shows that our approach responds in acceptable time (only few seconds).

7 Conclusions

In this paper, we have addressed some crucial requirements towards effective XML keyword search, including: measuring relevancy degree of a candidate and identifying desired results amongst numerous candidates. To fulfil those requirements, we have proposed an approach for relevancy measurement using Mutual Information concept from information theory. The skyline semantics are obtained for desired result identification.Finally, extensive experiments have been conducted and the results show that our work is very promising in terms of effectiveness and efficiency.

References

1. Bartolini, I., Ciaccia, P., Patella, M.: Efficient sort-based skyline evaluation. ACM Trans. Database Syst. 33(4), 1–49 (2008)
2. Bender, M.A., Farach-Colton, M., Pemmasani, G., Skiena, S., Sumazin, P.: Lowest common ancestors in trees and directed acyclic graphs. Journal of Algorithms 57, 75–94 (2005)
3. Börzsönyi, S., Kossmann, D., Stocker, K.: The skyline operator. In: Proceedings of the 17th International Conference on Data Engineering, Washington, DC, USA, pp. 421–430. IEEE Computer Society, Los Alamitos (2001)
4. Borzsonyi, S., Stocker, K., Kossmann, D.: The skyline operator. In: International Conference on Data Engineering, vol. 0, p. 421 (2001)
5. Chomicki, J., Godfrey, P., Gryz, J., Liang, D.: Skyline with presorting. In: ICDE, pp. 717–816 (2003)
6. Chomicki, J., Godfrey, P., Gryz, J., Liang, D.: Skyline with presorting. In: International Conference on Data Engineering, vol. 0, p. 717 (2003)
7. Cohen, S., Mamou, J., Kanza, Y., Sagiv, Y.: XSEarch: a semantic search engine for XML. VLDB Endowment, 45–56 (2003)
8. Cover, T.M., Thomas, J.A.: Elements of information theory. Wiley Interscience, New York (1991)
9. Guo, L., Shao, F., Botev, C., Shanmugasundaram, J.: XRANK: ranked keyword search over xml documents. In: SIGMOD 2003: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, pp. 16–27. ACM, New York (2003)
10. Kossmann, D., Ramsak, F., Rost, S.: Shooting stars in the sky: an online algorithm for skyline queries. In: VLDB 2002: Proceedings of the 28th International Conference on Very Large Data Bases, pp. 275–286. VLDB Endowment (2002)
11. Li, G., Feng, J., Wang, J., Zhou, L.: Effective keyword search for valuable lcas over xml documents. In: CIKM 2007: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, pp. 31–40. ACM, New York (2007)
12. Li, Y., Yu, C., Jagadish, H.V.: Enabling schema-free xquery with meaningful query focus. The VLDB Journal 17(3), 355–377 (2008)
13. Liu, Z., Chen, Y.: Reasoning and identifying relevant matches for xml keyword search. In: VLDB 2008: Proceedings of the 34th International Conference on Very Large Data Bases, pp. 921–932 (2008)
14. Tan, K.-L., Eng, P.-K., Ooi, B.C.: Efficient progressive skyline computation. In: VLDB 2001: Proceedings of the 28th International Conference on Very Large Data Bases, pp. 301–310 (2001)
15. Xu, Y., Papakonstantinou, Y.: Efficient keyword search for smallest lcas in xml databases. In: SIGMOD 2005: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, pp. 527–538. ACM, New York (2005)
16. Xu, Y., Papakonstantinou, Y.: Efficient lca based keyword search in xml data. In: EDBT 2008: Proceedings of the 11th International Conference on Extending Database Technology, pp. 535–546. ACM, New York (2008)
17. Zhou, R., Liu, C., Li, J.: Fast elca computation for keyword queries on xml data. In: EDBT 2010: Proceedings of the 13th International Conference on Extending Database Technology, pp. 549–560. ACM, New York (2010)