RACE: Finding and Ranking Compact Connected Trees for Keyword Proximity Search over XML Documents

Guoliang Li[1], Jianhua Feng[1], Jianyong Wang[1], Bei Yu[2], and Yukai He[1]
[1]Department of Computer Science and Technology
Tsinghua University, Beijing, China
[2]School of Computing
National University of Singapore, Singapore
{liguoliang,fengjh,jianyong}@tsinghua.edu.cn; yubei@comp.nus.edu.sg;
heyk05@mails.tsinghua.edu.cn

ABSTRACT

In this paper, we study the problem of keyword proximity search over XML documents and leverage the efficiency and effectiveness. We take the disjunctive semantics among input keywords into consideration and identify meaningful compact connected trees as the answers of keyword proximity queries. We introduce the notions of Compact Lowest Common Ancestor (CLCA) and Maximal CLCA (MCLCA) and propose Compact Connected Trees (CCTrees) and Maximal CCTrees (MCCTrees) to efficiently and effectively answer keyword queries. We propose a novel ranking mechanism, RACE, to Rank compAct Connected trEes, by taking into consideration both the structural similarity and the textual similarity. Our extensive experimental study shows that our method achieves both high search efficiency and effectiveness, and outperforms existing approaches significantly.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Miscellaneous

General Terms

Algorithms, Performance, Languages

Keywords

Lowest Common Ancestor (LCA), Compact LCA (CLCA), Maximal CLCA(MCLCA)

1. INTRODUCTION

Keyword search is a proven and widely accepted mechanism for querying in textual document systems and World Wide Web. The research community has recently recognized the benefits of keyword search and has been introducing keyword search capability into XML documents [2, 4, 5, 6, 7].

In this paper, we study the problem of keyword proximity search over XML documents by considering the disjunctive semantics (i.e., the OR predicate) among the input keywords, and provide a novel ranking mechanism for effective keyword search, by taking into account both the structural similarity from the DB point of view and the textual similarity from the IR viewpoint. We introduce the notions of Compact LCA (CLCA) and Maximal CLCA (MCLCA) to capture the focuses of keyword queries, and propose Compact Connected Trees (CCTrees) and Maximal CCTrees (MCCTrees) to efficiently and effectively answer keyword proximity queries. Moreover, we devise a novel ranking mechanism, RACE, to Rank compAct Connected trEes. RACE not only considers the textual similarity like document relevancy in IR literature, but also incorporates the structural similarity into the ranking function from the DB point of view.

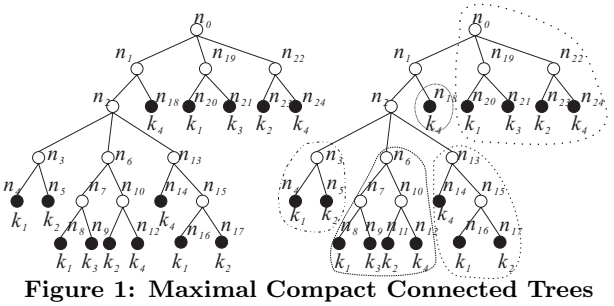

Figure 1: Maximal Compact Connected Trees

2. COMPACT CONNECTED TREES

Traditional methods usually compute the LCAs of content nodes to answer keyword queries. However, it is inefficient to compute all the LCAs as given a keyword query $\{k_1, k_2, \cdots, k_m\}$, there are $\prod_{i=1}^{m}|\mathcal{I}_i|$ combinations of LCA candidates, where $\mathcal{I}_i$ denotes the set of *content nodes* that directly contain keyword k_i. To address this problem, we introduce the concepts of *Compact LCA* (CLCA) and Compact Connected Trees (CCTrees).

Definition 2.1. *(CLCA and CCTree) Given q content nodes, $v_1 \in \mathcal{I}_1$, $v_2 \in \mathcal{I}_2, \cdots, v_q \in \mathcal{I}_q$, and $w = LCA(v_1, v_2, \cdots, v_q)$. w is said to dominate v_i w.r.t. $\{k_1, k_2, \cdots, k_q\}$, if $w \succeq LCA(v'_1, \cdots, v'_{i-1}, v_i, v'_{i+1}, \cdots, v'_q)$, $\forall v'_1 \in \mathcal{I}_1, v'_2 \in \mathcal{I}_2, \cdots, v'_{i-1} \in \mathcal{I}_{i-1}$, $v'_{i+1} \in \mathcal{I}_{i+1}, \cdots, v'_q \in \mathcal{I}_q$. w is a CLCA w.r.t. $\{k_1, k_2, \cdots, k_q\}$, if w dominates each v_i for $1 \le i \le q$. The tree rooted at a CLCA and containing the paths from the root to the nodes dominated by the root, is called a CCTree.*

A CLCA is the LCA of some relevant nodes and the irrelevant nodes cannot share a CLCA. For example, in Figure 1, n_3 is the CLCA of n_4 and n_5 w.r.t. $\{k_1, k_2\}$, however, n_2 is not the CLCA of n_4 and n_{17}, although n_2 is their LCA. Because n_3 dominates n_4, and n_{15} dominates n_{17}, but there is no node which dominates both n_4 and n_{17}. We observe that n_3 and n_{15} are more relevant to $\{k_1, k_2\}$ than n_2.

WWW 2008, April 21–25, 2008, Beijing, China.
ACM 978-1-60558-085-2/08/04.

The subtree rooted at n_3 is a CCTree. CLCA is orthogonal to SLCA [7] and avoids false negatives introduced by SLCA. For example, in Figure 1, n_0 and n_6 are both CLCAs w.r.t. $\{k_1,k_2,k_3,k_4\}$, and they dominate $\{n_{20},n_{21},n_{23},n_{24}\}$ and $\{n_8,n_9,n_{11},n_{12}\}$, respectively. n_0 is a false negative for SLCA as n_0 has a LCA descendant n_6. CLCA can avoid those false negatives and thus is a more meaningful methodology to answer keyword queries. We give the least upper bound of the number of CLCAs as stated in LEMMA 2.1, which is much smaller than the number of LCAs.

LEMMA 2.1. *There are at most* $2\sum_{i=1}^{m}|\mathcal{I}_i|$-1 *CLCAs w.r.t. a query* $\mathcal{K}=(k_1,k_2,\cdots,k_m)$ *and an XML document* $\mathcal{D}$ *in terms of the disjunctive semantics (i.e., the OR predicate).*

DEFINITION 2.2. *(MCLCA and MCCTree) Given a keyword query* $\mathcal{K}=\{k_1,k_2,\cdots,k_m\}$ *and* $\mathcal{K}_i=\{k_{i_1},k_{i_2},\cdots,k_{i_q}\}\subseteq\mathcal{K}$. *Suppose* $w=CLCA(v_{i_1},v_{i_2},\cdots,v_{i_q})$, *where* $v_{i_1}\in\mathcal{I}_{i_1},\cdots,v_{i_q}\in\mathcal{I}_{i_q}$. w *is a Maximal CLCA (MCLCA), if* $\forall k'\in(\mathcal{K}\text{-}\mathcal{K}_i)$, $v'_k\in\mathcal{I}_{k'}$, $\nexists w'$, *which dominates both* v'_k *and every* v_{i_j} *for* $1\leq j\leq q$. *The CCTree rooted at an MCLCA is called an MCCTree.*

To effectively answer keyword search, we propose the concepts of Maximal CLCA and Maximal CCTree. An MCLCA is also a CLCA, which has no ancestors that still dominate some other content nodes besides the content nodes dominated by the MCLCA. Therefore, an MCLCA dominates a maximal set of content nodes and is more meaningful than a CLCA. An MCCTree is the CCTree rooted at an MCLCA and contains more keywords than CCTrees. For example, in Figure 1, the four circled trees are MCCTrees.

3. RACE

$TF{\cdot}IDF$ based methods for ranking relevant documents have been proved to be effective for keyword proximity search in text documents. However, traditional ranking techniques in IR literature may not be effective to rank MCCTrees, as besides the term frequency (tf) and inverse document frequency (idf), MCCTrees also contain rather rich structural information. We take into account both the structural similarity and traditional IR metrics to rank MCCTrees.

There are three parameters - the number of content nodes in $\mathcal{T}$, n_c, the number of distinct input keywords contained in $\mathcal{T}$, n_k, and the number of all nodes in $\mathcal{T}$, n_s, which will affect the score assigned to an MCCTree, and we will employ these three parameters to rank MCCTrees. Intuitively, the larger n_c, the higher the score of the MCCTree should be; on the other hand, the larger n_k, the more likely the MCCTree is relevant to $\mathcal{K}$. On the contrary, n_s should be inverse with the score of the MCCTree. In addition, the succinctness of the MCCTree should be reflected in the structural similarity function, and the more succinct of the MCCTree, the higher score of the structural similarity should be. Based on above observations, we can compute the structural similarity.

Accordingly, we combine the textual similarity and structural similarity to effectively rank the MCCTrees.

4. EXPERIMENTAL STUDY

We have conducted a set of experiments to evaluate the performance of our approach. We used real dataset DBLP in our experiments. The raw file was about 420MB. The experiments were conducted on an Intel(R) Pentium(R) 2.4GHz computer with 1GB of RAM. The algorithms were implemented in Java. We compared RACE with state-of-the-art methods, XSEarch[1], XRank[2], GDMCT[3] and MSLCA [7]. We selected six groups of queries, $G_1,\cdots,G_6$. Each group has ten queries and the queries in the same group have the same number of keywords. For example, each query in G_3 has 3 keywords. Figure 2 illustrates the experimental results on search efficiency and Figure 3 gives the experimental results on search quality.

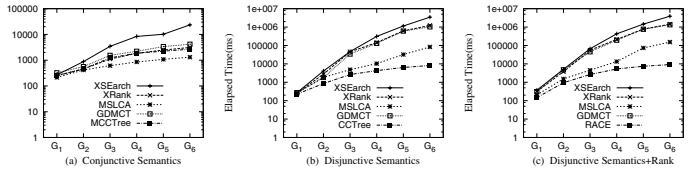

Figure 2: Efficiency of various algorithms

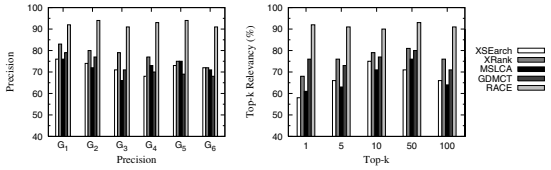

Figure 3: Top-k answer relevancy

5. CONCLUSION

In this paper, we have investigated the problem of keyword proximity search over XML documents. We proposed the notions of CLCA and MCLCA to capture the focuses of keyword queries and adopted CCTrees and MCCTrees to effectively and efficiently answer keyword proximity queries. We demonstrated a novel ranking mechanism, RACE, to Rank the compAct Connected trEes, by taking into account both structural similarity from the DB viewpoint and textual similarity from the IR point of view. The experimental results show that our approach achieves high search efficiency and quality, and outperforms existing methods significantly.

6. ACKNOWLEDGEMENT

This work is partly supported by the National Natural Science Foundation of China under Grant No.60573094, the National High Technology Development 863 Program of China under Grant No.2007AA01Z152 and 2006AA01A101, the National Grand Fundamental Research 973 Program of China under Grant No.2006CB303103.

7. REFERENCES

[1] S. Cohen, J. Mamou, Y. Kanza, and Y. Sagiv. Xsearch: A semantic search engine for xml. In *VLDB*, 2003.

[2] L. Guo, F. Shao, C. Botev, and J. Shanmugasundaram. Xrank: Ranked keyword search over xml documents. In *SIGMOD*, 2003.

[3] V. Hristidis, N. Koudas, Y. Papakonstantinou, and D. Srivastava. Keyword proximity search in xml trees. In *IEEE TKDE 18(4)*, 2006.

[4] G. Li, J. Feng, J. Wang, and L. Zhou. Efficient keyword search for valuable lcas over xml documents. In *CIKM*, 2007.

[5] G. Li, J. Feng, J. Wang, and L. Zhou. SAILER: An Effective Search Engine for Unified Retrieval of Heterogeneous XML and Web Documents. In *WWW*, 2008.

[6] G. Li, B.C. Ooi, J. Feng, J. Wang, and L. Zhou. EASE: Efficient and Adaptive Keyword Search on Unstructured, Semi-structured and Structured Data. In *SIGMOD*, 2008.

[7] C. Sun, C. Y. Chan, and A. K. Goenka. Multiway slca-based keyword search in xml data. In *WWW*, 2007.