International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 4 | ISSN : 2456-3307

Finding Temporal Graphs Using Keywords

Ramaswamy Satyanarayana Sankar, K Somasekhar
[1]Student,Department of MCA, RCR Institutions of Management & Technology, Tirupathi, Andhra Pradesh, India
[2]Assistant Professor, Department of MCA, RCR Institutions of Management & Technology, Tirupathi, Andhra Pradesh, India

ABSTRACT

Archiving graph data over history is demanded in many applications, such as social network studies, collaborative projects, scientific graph databases, and bibliographies. Typically people are interested in querying temporal graphs. Existing keyword search approaches for graph-structured data are insufficient for querying temporal graphs. This paper initiates the study of supporting keyword-based queries on temporal graphs. We propose a search syntax that is a moderate extension of keyword search, which allows casual users to easily search temporal graphs with optional predicates and ranking functions related to timestamps. To generate results efficiently, we first propose a best path iterator, which finds the paths between two data nodes in each snapshot that is the "best" with respect to three ranking factors. It prunes invalid or inferior paths and maximizes shared processing among different snapshots. Then we develop algorithms that efficiently generate top-k query results. Extensive experiments verified the efficiency and effectiveness of our approach.

I. INTRODUCTION

Archiving graph data is important in many applications. For example, in social network analysis, scientists are often interested in analyzing the temporal dynamics of social relationships in order to understand how things change and to predict trends. We even want to archive the whole Web (i.e. the pages and their links) so that our future generations are able to see what is happening today, and analyze how things evolve. In a collaborative project, we would like to archive previous versions of data (such as workflows or programs) so that an earlier version can be recovered in case of a mistake. Beyond the basic operations such as retrieving a snapshot from the archive and tracing the history of an element, often users want to query temporal graphs, as illustrated in some examples below.

To address such query needs, studies have been performed on designing a temporal relational model and a temporal XML tree model and developing query languages on temporal data, including TQUEL [26], TSQL2 and SQL3 for temporal relational data, and TXPath for temporal XML trees. However, performing structured temporal queries on temporal databases is not suitable for some applications for two reasons. First, many applications have graph structured data which can not be efficiently handled by relational databases as generally one join is required to navigate each edge. Second, structured queries are difficult for casual users to learn and are error-prone even for expert users.

To allow casual users who are not computer experts (as found in Web and scientific applications) to express queries on temporal graphs, it is critical to support keyword-based searches on temporal graphs.

There is much research on keyword search on graphs that do not encode temporal information. Given a data graph, each query result is a minimal tree that contains all query keywords, and the results are ranked by the reverse of weighted tree sizes. To generate top-k results efficiently, a widely-adopted approach is based on Dijkstra's shortest path algorithm, which explores the graph from each keyword match using a shortest path iterator. A result is found if a node is visited by iterators from all query keywords. In this way, results are likely generated in the order of their sizes, thus enabling efficient top-k processing.

II. EXISTING SYSTEM

Archiving graph data is important in many applications. For example, in social network analysis, scientists are often interested in analyzing the temporal dynamics of social relationships in order to understand how things change and to predict trends. Archiving graph data over history is demanded in many applications, such as social network studies, collaborative projects, scientific graph databases, and bibliographies. Typically people are interested in querying temporal graphs. Existing keyword search approaches for graph-structured data are insufficient for querying temporal graphs.

Disadvantages:

1. Existing keyword search approaches for graph-structured data are insufficient for querying temporal graphs.
2. Efficiency and effectiveness is less.

III. PROPOSED SYSTEM

This paper initiates the study of supporting keyword-based queries on temporal graphs. We propose a search syntax that is a moderate extension of keyword search, which allows casual users to easily search temporal graphs with optional predicates and ranking functions related to timestamps. To generate results efficiently, we first propose a best path iterator, which finds the paths between two data nodes in each snapshot that is the "best" with respect to three ranking factors. It prunes invalid or inferior paths and maximizes shared processing among different snapshots. Then we develop algorithms that efficiently generate top-k query results. Extensive experiments verified the efficiency and effectiveness of our approach.

Advantages:

1. Efficiency and effectiveness is high.
2. Search approaches for graph-structured data are highly sufficient for querying temporal graphs.

IV. CONCLUSION

We initiate the study of the problem of searching temporal graphs. We propose a simple yet expressive keyword based query syntax that allows temporal information to be specified as either predicates or ranking factors. We propose a best path iterator, which finds the "best" paths between two data nodes in each time instant with respect to ranking functions where the rank of a path is monotonically non-increasing upon an edge expansion. Then we propose algorithms to efficiently evaluate this type of queries on a temporal graph to generate top-k results. The efficiency and effectiveness of the proposed approach are verified through extensive empirical studies

V. REFERENCES

[1]. Tsql2 and sql3 interactions. http://www.cs.arizona.edu/people/ rts/sql3.html.

[2]. VisTrails. http://vistrails.org.

[3]. WebArchive Project.http://oak.cs.ucla.edu/ cho/research/ archive.html.

[4]. A.Balmin, V.Hristidis, and Y.Papakonstantinou.ObjectRank: Authority-Based Keyword Search in Databases.In VLDB, pages 564–575, 2004.

[5]. G.Bhalotia, A.Hulgeri, C.Nakhe, S.Chakrabarti, and S.Sudarshan.Keyword Searching and

Browsing in Databases using BANKS.In ICDE, pages 431–440, 2002.

[6]. R.Bin-Thalab and N.El-Tazi.TOIX: Temporal Object Indexing for XML Documents.In DEXA, pages 235–249, 2015.

[7]. R.Bin-Thalab, N.El-Tazi, and M.E.El-Sharkawi.TMIX: Temporal Model for Indexing XML Documents.In AICCSA, pages 1–8, 2013.

[8]. J.Coffman and A.C.Weaver.A Framework for Evaluating Database Keyword Search Strategies.In CIKM, 2010.

[9]. B.Ding, J.X.Yu, and L.Qin.Finding Time-Dependent Shortest Paths over Large Graphs.In EDBT, pages 205–216, 2008.

[10]. B.Ding, J.X.Yu, S.Wang, L.Qin, X.Zhang, and X.Lin.Finding Top-k Min-Cost Connected Trees in Databases.In ICDE, pages 836–845, 2007.

[11]. A.Fard, A.Abdolrashidi, L.Ramaswamy, and J.A.Miller.Towards Efficient Query Processing on Massive Time-evolving Graphs.In CollaborateCom, pages 567–574, 2012.

[12]. K.Golenberg, B.Kimelfeld, and Y.Sagiv.Keyword Proximity Search in Complex Data Graphs.In SIGMOD Conference, pages 927–940, 2008.

[13]. H.He, H.Wang, J.Yang, and P.S.Yu.BLINKS: Ranked Keyword Searches on Graphs.In SIGMOD Conference, pages 305–316, 2007.

[14]. V.Hristidis, L.Gravano, and Y.Papakonstantinou.Efficient IRStyle Keyword Search over Relational Databases.In VLDB, pages 850–861, 2003.

[15]. W.Huo and V.J.Tsotras.Efficient Temporal Shortest Path Queries on Evolving Social Graphs.In SSDBM, pages 38:1–38:4, 2014.

[16]. C.S.Jensen, R.T.Snodgrass, and M.D.Soo.The TSQL2 Data Model.In The TSQL2 Temporal Query Language.1995.

[17]. V.Kacholia, S.Pandit, S.Chakrabarti, S.Sudarshan, R.Desai, and H.Karambelkar.Bidirectional Expansion For Keyword Search on Graph Databases.In VLDB, pages 505–516, 2005.

[18]. D.Kempe, J.Kleinberg, and A.Kumar.Connectivity and inference problems for temporal networks.In Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing, STOC '00, pages 504–513, New York, NY, USA, 2000.ACM.

[19]. B.Kimelfeld and Y.Sagiv.Finding and Approximating Top-k Answers in Keyword Proximity Search.In PODS, 2006.

[20]. G.Koloniari, D.Souravlias, and E.Pitoura.On Graph Deltas for Historical Queries.CoRR, abs/1302.5549, 2013.

[21]. Y.Luo, X.Lin, W.Wang, and X.Zhou.SPARK: Top-k Keyword Query in Relational Databases.In SIGMOD Conference, pages 115–126, 2007.

[22]. L.Qin, J.X.Yu, L.Chang, and Y.Tao.Querying Communities in Relational Databases.In ICDE, pages 724–735, 2009.

[23]. C.Ren, E.Lo, B.Kao, X.Zhu, and R.Cheng.On Querying Historical Evolving Graph Sequences.PVLDB, 4(11), 2011.

[24]. F.Rizzolo and A.A.Vaisman.Temporal XML: Modeling, Indexing, and Query Processing.VLDB J., 17(5):1179–1212, 2008.

[25]. M.Sayyadian, H.LeKhac, A.Doan, and L.Gravano.Efficient Keyword Search Across Heterogeneous Relational Databases.In ICDE, pages 346–355, 2007.

[26]. R.T.Snodgrass.The Temporal Query Language TQuel.ACM Trans.Database Syst., 12(2):247–298, 1987.

[27]. M.Steinbauer and G.Anderst-Kotsis.DynamoGraph: A Distributed System for Large-scale, Temporal Graph Processing, its Implementation and First Observations.In WWW, 2016.

[28]. L.Zhang, T.Tran, and A.Rettinger.Probabilistic Query Rewriting for Efficient and Effective Keyword Search on Graph Data.In PVLDB, 2014.

[29]. B.Zong, X.Xiao, Z.Li, Z.Wu, Z.Qian, X.Yan, A.K.Singh, and G.Jiang.Behavior Query Discovery in System-Generated Temporal Graphs.PVLDB, 9(4):240–251, 2015.