Keyword Based Temporal Query Execution Over Graph

Ms. Pooja R. Ghuge
School of Computer Science
Sandip University, Nashik, Maharashtra

Prof. Mohan Nikam
School of Computer Science
Sandip University, Nashik, Maharashtra

Abstract—Graph mining is required in variety of applications in real world data processing. It is implemented in social network, scientific graph database, bibliography analysis,etc. Most of the studies requires temporal results over graph data. In the literature keyword based search is proposed but it is inefficient to answer keyword queries. The proposed system works on querying the graph based on temporal information. Query contains keyword search as well as the temporal search. To fire a query, a user friendly automatic query generator panel is proposed. User can selects the required parameter to fire a query on temporal graph. Temporal nodes are used to support temporal queries. To travel the graph best path iterator is used with a ranking criteria. Along with a temporal keyword queries, this work also proposes a collaborative aggregate functions to generate archived results against the temporal information.

Index Terms—graph, temporal graph, graph keyword search, temporal query, aggregate query

INTRODUCTION

In variety of domains such as social network, bibliography generation, etc. graph dataset is generated. The large graph dataset contains huge information. From such graph dataset, summarized information retrieval and archived result extraction is required. The graph data along with time information is known as temporal graph data. In some collaborative projects, the graph data is partitioned based on the time information and each partition is archived and kept separately. Each partition is called as graph snapshot. The data is extracted for these graph snapshot as and when required based on the temporal information and keywords in a query. But this partitioning technique restricts the some queries that want to relate existing and current scenario based on the temporal information. Such temporal queries help to analyse change, to study life cycle, to identify trends etc. Also in prediction system temporal data analysis is prime function.

A lot of research work has been done on keyword search on graph. But only keyword matching technique is not sufficient to answer temporal queries.

Following are some examples of queries on temporal graph:

1) Find articles on space science after 2010
2) Find people who joined facebook after 2017

Temporal relation model is able to answer such queries. The relation model defines a relational database and sql technique are efficient to generate result for such queries [2]. But every application not generates the relational database. Some ap- plication requires graph database and graph database is not efficiently converted in to the relational database. XML is again one option. Temporal xml model is proposed in literature to answer to answer temporal queries on xml.[3]

Keyword based searching on temporal graph can be done using temporal graph snapshot. The temporal information is matched with the graph snapshot partition information and keyword matching is done in every graph snapshot that matches the temporal criteria. But graph based temporal in- formation preservation using graph snapshot requires higher storage space. The query execution requires high processing time as number of traversal increases based on matched temporal graph snapshot.

For computer based query execution, a predefined query format is required. SQL queries are structured queries. A query parser processes the query and extracts the desired meaning of query and accordingly finds the matched results. But sql query has structural complexity and training is required to execute a query on dataset. Such queries cannot be handled directly by any casual user. A simple structure of query or visual query generator is required for casual user to generate computer based temporal query.

A keyword matching and temporal aspect matching queries retrieves the desired result but result formatting is also required for example result should be formatted in increasing order of temporal information, or grouped by object name or generate a collaborative result like count of matched results, etc.

A query generation on temporal graph covers multiple aspects such as query structure, query generation and its parsing, query facilities like ranking, sorting, grouping, etc. This query should be executed efficiently. There is need to balance a trade of between efficiency of query execution and efficiency in graph storage.

In the following section various graph based search tech- niques are discussed along with their scope and limitations, followed by problem formulation and new system architecture is proposed.

I. REVIEW OF LITERATURE

Keyword search on graph is proposed with minimal tree semantics. This technique filters the graph using sub tree. From graph, sub trees are generated based on matched keywords. Based on the tree height top k results are extracted. Tree height represents the relevance between query and result. Small height represents high relevance. This type of keyword search using tree semantics do not support temporal queries[4]. Unlike tree formation and sub-tree generation with mini- mum length, graph path traversal technique such as Dijkstras algorithm[6] is proposed for keyword searching on graph. This technique finds the shortest path between keywords present in query. A bank technique is used to find sub-graph from graph dataset. The sub-graph is detected using shortest path identification from matched keyword nodes. The Banks algorithm uses Dijkstras algorithm to find shortest path. To find the entire result, the Dijkstras algorithm runs multiple times on graph. Hence banks algorithm requires longer time for execution.[5].

In the databases temporal information can be preserved. The temporal information is the transaction time or a valid time of touple. The database containing temporal information called as temporal database. This database preserves the relational database structure. SQL3[9], TQUEL[7] and TSQL2[8] are the languages that works with temporal database. User can fire query using these languages on temporal database. But the structure of these languages is complicated and hence cannot be directly used by casual user. Temporal database is also unable to represent graph dataset.

Temporal graph are time evolving graph also called as dynamic graph. Based on time varying history, user can extract matched single nodes with matched temporal information[10]. To match multiple nodes distributed system is proposed. A distributed system helps to store huge graph dataset and extracts multiple nodes by generating subgraph of matched nodes[11].

For large graph processing , graph partitioning technique is proposed. The graph partitioning can be done based on structural information or temporal information. The advantages and challenges of these techniques are also discussed in literature.[14]

Temporal query support is proposed using graph snapshot. A temporal information is matched with snapshot information and keyword based search technique is applied in each graph snapshot. To match temporal information with graph snapshot, clusters are generated based on the time information. For key- word matching and subgraph generation Dijkstras algorithm is used. The result is generated collectively by executing the keyword matching algorithm on matched graph snapshot.[12] Road network or communication networks are the example of dynamic graphs that preserves the temporal information. The graph structure varies with respect to time. Structural variation represents the connectivity among nodes. In dynamic graphs, connectivity along with edge weight varies with re- spect to time. Processing of such dynamic temporal graph is proposed in literature.[14]

Z. Liu, C. Wang and Y. Chen [1] proposes a technique to fire multi keyword query on temporal graph. This technique over- comes the problem of storage space. Instead of graph snapshot, a single graph is preserved with temporal information. This also reduces the execution time as single graph is traversed for query execution. The graph is directed graph and each edge has time interval information. Dijkstras algorithm is used find shortest path between matched nodes of graph. A custom query structure is proposed. A query contains query keywords, predicate and ranking function. Predicate includes the search constraints whereas ranking function includes sorting order. A custom query processor parses the query and generates the result. Casual user need to study the structure first. If appropriate structured query is not provided, query parser will failed to generate the resultant sub-graph.

ANALYSIS AND PROBLEM FORMULATION

Most of the applications generates graph dataset. Traditional datasets are unable to handle graph dataset. Some applications require temporal information in graph. A keyword query with temporal filter is required. Temporal graph snapshot supports temporal queries. But such graph snapshot processing requires higher storage space and high processing time. A single graph with temporal information resolves the storage problem. A custom query syntax requires training for casual users. Hence user friendly graphical interface is required for query generation. Along with keyword matching, statistical summarized result generation is also required.

I. SYSTEM OVERVIEW

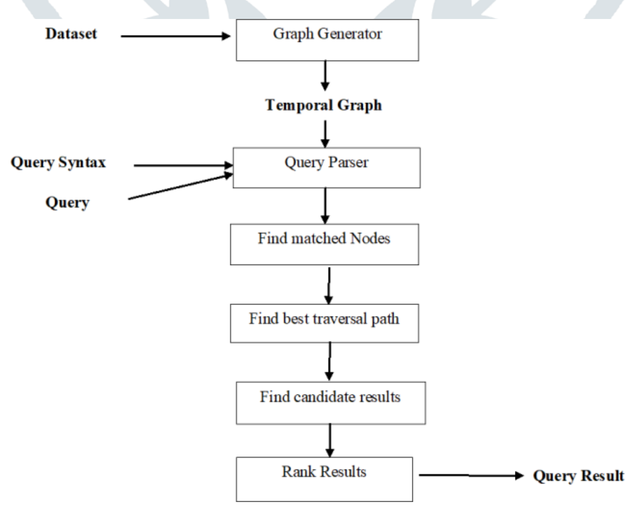

Fig. 1. System Architecture

II. Above figure shows the system architecture.

A. System Working:

The system provides keyword searching over temporal graph. Temporal information is stored with every node and edge in graph. A graphical user interface is provided that automatically generates query as per the user requirement. Sys- tem executes the query by traversing a graph and extracts the candidate results. By applying ranking, sorting and aggregate functions system generates final result. XML dataset is input to the system. xml is parsed and temporal graph dataset is generated as a preprocessing task. User fire query on temporal graph dataset using graphical user interface. The query contains mainly 4 sections:

1) keyword : it contains searching keyword list.
2) predicate: it represents the temporal information filtering criteria
3) ranking: This section represent the result formatting such as sorting
4) aggregation: A summary generation functions such as sum, count, average is present in this section.

For graph traversal best path iterator algorithm is used. This algorithm finds the shortest path between matched keywords nodes using dijkstras algorithm. Selected path nodes are fil- tered and saved in priority queue with the help of NTD triplet. NTD triplet contains Node value, time Interval information and Distance value.

After generating candidate results ranking and aggregate functions are applied and final result is generated.

III. CONCLUSION

Keyword based searching over graph dataset is proposed along with temporal information filter. To fire a query, a custom query generator is proposed that helps casual user to built a query without predefined training. Along with the keyword searching and temporal filtering, result formatting and aggregate result generation is also proposed. The result formatting constrains result sorting where as aggregate func-tion generates statistical summarised results.

REFERENCES

[1] Ziyang Liu, Chong Wang, Yi Chen, ”Keyword Search on Temporal Graphs,” in IEEE Transactions on Knowledge and Data Engineering Vol. 29, No: 8, Aug. 2017

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

[3] F. Rizzolo and A. A. Vaisman. Temporal XML: Modeling, Indexing, and Query Processing. VLDB J., 17(5):11791212, 2008.

[4] K. Golenberg, B. Kimelfeld, and Y. Sagiv. Keyword Proximity Search in Complex Data Graphs. In SIGMOD Conference, pages 927940, 2008.

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

[6] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan. Keyword Searching and Browsing in Databases using BANKS. In ICDE, pages 431440, 2002.

[7] R. T. Snodgrass. The Temporal Query Language TQuel. ACM Trans. Database Syst., 12(2):247298, 1987.

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

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

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

[11] A. Fard, A. Abdolrashidi, L. Ramaswamy, and J. A. Miller. Towards Efficient Query Processing on Massive Time-evolving Graphs. In Collab- orateCom, pages 567574, 2012.

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

[13] B. Ding, J. X. Yu, and L. Qin. Finding Time-Dependent Shortest Paths over Large Graphs. In EDBT, pages 205216, 2008.

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