International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 6 | ISSN : 2456-3307

Keyword Search on Hyper Graph Data Bases

D. Pavani
M.Tech Scholar, Computer Science and Engineering, JNTUA College of Engineering, Ananthapuramu, Andhra Pradesh, India

ABSTRACT

Archiving graph data is necessary in many applications. This graph data is handled with graph databases. They also consist of temporal graph. The temporal graph consists of temporal data. Temporal data refers to data, where it changes over time. Querying on temporal graphs the existing approaches are insufficient as they consume more time. This paper supports keyword search on temporal graphs efficiently by using hyper graphs. The main advantage of Hyper Graphs over Temporal graphs is keyword evolution time can be reduced drastically.

Keywords : Temporal Graphs, Hyper Graphs, Keyword Search.

I. INTRODUCTION

Question recovery frameworks are executed at different area particular Search Engine (e.g. DBLP, SNAP) to bring proposed results [1].User utilizes catchphrase look framework over picked information sets. A client creates a watchword question, submits it to the framework, and recovers pertinent answers. Web indexes utilizes an approach that may enable it to recognize how important the outcomes it presentations to searchers may really be, and how likely those outcomes are to demonstrate an assortment of results when a searcher utilizes an inquiry term that may cover a scope of themes in future. Age old earlier methodologies utilized Human Reviewers being one alternative for keeping an eye on the pertinence of indexed lists by physically screening the outcomes for each inquiry. Sadly, overlooking or not completely misusing the time measurement can be adverse for a huge group of inquiries for which we ought to consider the topical pertinence as well as the time angle as well. Such sort of questions can be prepared and gotten with the assistance of a RDBMS (SQL) [2] based frameworks effectively and productively, however is exceptionally intricate doing likewise on Graph Databases(GDB's).Graph databases resemble the up and coming age of social databases, yet with top of the line bolster for "connections", or those verifiable associations showed by means of outside keys in the customary social databases. Every hub (substance or quality) in the chart database demonstrate straightforwardly and physically contains a rundown of relationship-records that speak to its connections to different hubs. These relationship records are sorted out by sort and course and may hold extra traits. At whatever point you run what might as well be called a JOIN activity, the database just uses this rundown and has guide access to the associated hubs, wiping out the requirement for a costly pursuit/coordinate calculation. Anyway the time affectability measures are confounded in GDB's, so we require a superior technique to help time cut questions over GDB's.

II. PROBLEM DEFINITION

2.1 Background of Keyword Search on Graphs:

There is existing work on catchphrase look on non-temporal charts. Every hub in the diagram has a name, speaking to its tag or esteem. Hubs and edges in the chart may have weights. A watchword in the inquiry can coordinate words in the names of at least one hub in the information. A response to an inquiry is a sub tree of the information diagram that contains matches to all inquiry watchwords. Sub trees with littler size are viewed as better outcomes and positioned higher. We embrace a comparative model in this paper, aside from that hubs are additionally connected with time interims, and the pursuit grammar is improved with worldly provisions, as characterized in the accompanying.

2.2 Data Model:

We define the temporal gr5aph model in a similar way as the temporal XML model used [3]. The data is modelled as a directed graph, where each node n and edge e is annotated.

2.3 Query Syntax

We bolster stretched out watchword inquiries to transient diagrams, where a clients can issue an inquiry with discretionary worldly predicates and fleeting positioning capacities as characterized underneath.

The query syntax for searching temporal graph
is defined as:

_ <Q> ::= <KEYWORD>+ <PRED>* <RF>*

Where "<KEYWORD>+" represents one or more keywords, each of which may match the labels of data nodes, "<PRED>*"represents zero or more temporal predicates, and "<RF>*" represents zero or more ranking factors.

Table 1. Queries corresponding to the questions Q1-Q3

Q1:	"**Mary, John**",-ascending order of start time.
Q2:	"**Mike, friend**", - descending order of duration.
Q3:	"**Microsoft, employee**", result time precedes **2016.**

We support the following temporal predicates:

_ <PRED> ::= RESULT TIME
– {PRECEDES / FOLLOWS} tx
– MEETS tx
– OVERLAPS [tx; ty]
– {CONTAINS / CONTAINED BY} [tx; ty]

"Result time" alludes to the time when the outcome exists (Definition 2.2). While different predicates are plain as day, predicate "meets tx" implies that the outcome is substantial in tx, and is either invalid in whenever moment before tx, or invalid in whenever moment after tx. Predicates can be consolidated utilizing AND, OR and NOT.

We bolster the accompanying positioning components:

_ <RF>::= RANK BY
– DESCENDING ORDER OF {RELEVANCE /RESULT
END TIME /DURATION}
– ASCENDING ORDER OF RESULT START TIME

The linguistic structure of worldly predicates takes after the punctuation of TSQL2, the fleeting expansion to the SQL dialect standard. It can express any connections between the outcome time and a period moment tx or a period interim [tx; ty. For each match of symmetric connections, we just show one of them in the figure. For instance, Table 1 indicates Q1-Q3 under this linguistic structure. We take after best in class ways to deal with characterize importance as the weighted outcome tree estimate, the littler, the better. To encourage clients to express fleeting questions, the framework can furnish a graphical interface with drop-down menus that rundown choices for client choice.

III. EXISTING SYSTEM

The existing query model considers the following aspects.

If the relevant time period for a time-sensitive query is unspecified, several query processing approaches are possible. One option is to consequently recommend, in light of the inquiry terms, significant time range for the question and enable clients to unequivocally choose fitting time interim. As an alternative that demands less input from the users, it automates the previous procedure and prioritizes results from periods that we automatically identify as relevant. We would then be able to normally characterize the significance of a report as a mix of point likeness and time pertinence. Uses the following algorithms to evaluate and query GDB with time factor using Algorithms1 Best Path Iterator 2 Ranking by Duration 3 Searching Temporal Graphs. We have characterized a basic inquiry sentence structure that backings fleeting predicates and positioning components. Adjusted from TSQL2, this language structure can express all connections between the outcome time and a given time interim [4] [5]. We have created two principled speculation of Dijkstra's calculation for most limited ways.

One is to deal with transient diagrams where hubs have timestamps to accomplish depiction reducibility. The other is to describe the kind of positioning capacities that it can bolster, past the separation work in the customary setting. With these approaches, it can be said that automatic incorporating of time line hits to query results increases the performance of search engines and also provides better service to the user.

IV. PROPOSED SYSTEM

In this, we use the hyper graphs instead of the normal graphs. **Edges** as known from standard **graphs** model (directed or undirected) has 1:1 connections. **Hyper edges** as known from **hyper graphs** model (directed or undirected) has n:n connections. We propose to exploit stronger semantic relationship in the hyper graph for Query Search and Temporal Re-Ranking. Hyper graph based systems topic similarities computations are much faster and qualitative compared to plain GDB's. Unlike a graph that has an edge between two vertices, a set of vertices are connected by the hyper edge in a hyper graph. Common graph-based learning methods usually only consider the pair wise relationship between two vertices, ignoring the higher-order relationship among three or more vertices. Using this Dijkstra algorithm in the proposed method obtain better results. Compared to graph based methods hyper-graph based methods yield better performance with respect to relevancy and time.

4.1 Modules:

1. Host Setup

In this module the master node and worker/slave nodes are created.

Master node -

Master node is responsible for storage of data and parallel computation of the data. It allots the work to the slave nodes

Slave node –

Slave node keeps the slices of data in it and performs the work that master node is given to it.

2. Data Source Manager

It allocates the data to the workers from the dataset.

3. Optimized HGDB Query implementation

In this a search engine is created in which a query is given for searching.

4. Query Processor

This consists of Data source Server Pool Manage, Pool Load Status Calculator; Data source Server Job Assigner, Data source Server Job Transfers and Job Status Acknowledger.

5. Data Source Server

It is responsible for Job Arrivals, Job Accomplisher, Data Repository Selections, and Query Results of the slaves or worker nodes.

6. Data set

DBLP (Digital bibliography and Library project) data set is initially uploaded, which consists of journal information like title, author. The whole DBLP is sliced into number of datasets.

V. RESULT EVALUATION

The keyword is given to the search engine for searching. The keyword is being searched in the data sets as it is selected by user. For example, here consider the keyword as graph given to search engine and three data sets are selected. Then it gave 30 number of results in 4.58 sec in existing system where as it gave 30 number of results in 2.41sec with using hyper graph databases as it is shown in below figure 1.

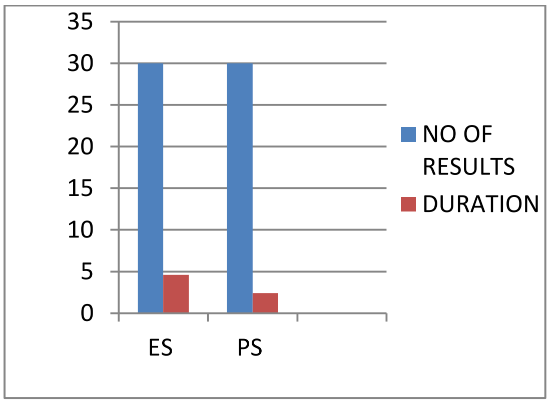
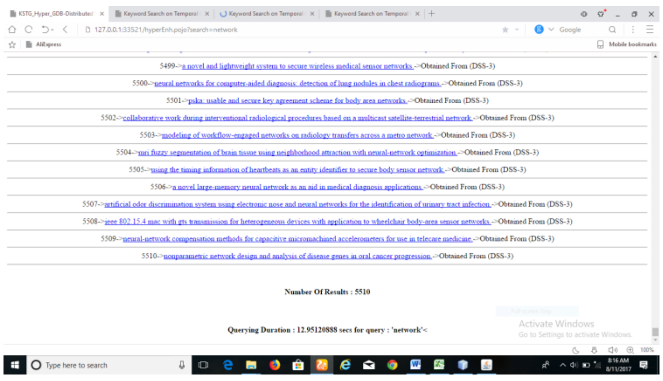


Figure 1. output representation using graph

The below shown two figures shows the output of existing system and the proposed systems respectively. In fig:2.we have given a keyword named "NETWORK(query)" to search then using existing system it has shown 5510 resulta in 16.2457(querying duration) where as with our proposed system using HYPER GRAPH method the same no of results i.e 5510 results have been shown in only 12.91 sec(querying duration) for same keyword "NETWORK(query)" as shown in fig:3.

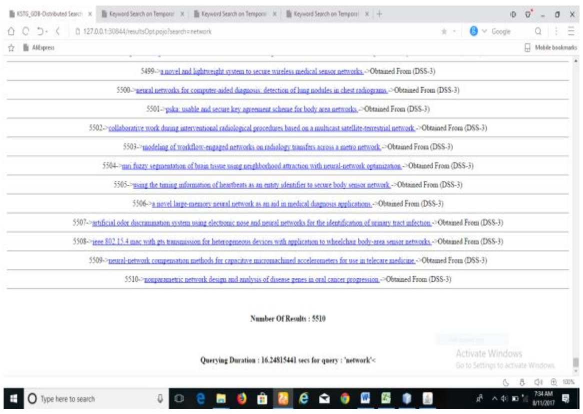

KSTG_Hyper_GDB-Distributed
Keyword Search on Temporal
Keyword Search on Temporal
Keyword Search on Temporal
127.0.0.1:33521/hyperEnh.pojo?search=network
Google
AliExpress
Mobile bookmarks
Activate Windows
Go to Settings to activate Windows.
Type here to search
8:16 AM
8/11/2017
100%

5499->a novel and lightweight system to secure wireless medical sensor networks.->Obtained From (DSS-3)
5500->neural networks for computer-aided diagnosis: detection of lung nodules in chest radiograms.->Obtained From (DSS-3)
5501->pska: usable and secure key agreement scheme for body area networks.->Obtained From (DSS-3)
5502->collaborative work during interventional radiological procedures based on a multicast satellite-terrestrial network.->Obtained From (DSS-3)
5503->modeling of workflow-engaged networks on radiology transfers across a metro network.->Obtained From (DSS-3)
5504->mri fuzzy segmentation of brain tissue using neighborhood attraction with neural-network optimization.->Obtained From (DSS-3)
5505->using the timing information of heartbeats as an entity identifier to secure body sensor network.->Obtained From (DSS-3)
5506->a novel large-memory neural network as an aid in medical diagnosis applications.->Obtained From (DSS-3)
5507->artificial odor discrimination system using electronic nose and neural networks for the identification of urinary tract infection.->Obtained From (DSS-3)
5508->ieee 802.15.4 mac with gts transmission for heterogeneous devices with application to wheelchair body-area sensor networks.->Obtained From (DSS-3)
5509->neural-network compensation methods for capacitive micromachined accelerometers for use in telecare medicine.->Obtained From (DSS-3)
5510->nonparametric network design and analysis of disease genes in oral cancer progression.->Obtained From (DSS-3)

Number Of Results : 5510

Querying Duration : 12.95120888 secs for query : 'network'<

Figure 2. output of existing system


KSTG_GDB-Distributed Search
Keyword Search on Temporal
Keyword Search on Temporal
Keyword Search on Temporal
127.0.0.1:30844/resultsOpt.pojo?search=network
Google
AliExpress
Mobile bookmarks
Activate Windows
Go to Settings to activate Windows.
Type here to search
7:34 AM
8/11/2017
100%

5499->a novel and lightweight system to secure wireless medical sensor networks.->Obtained From (DSS-3)
5500->neural networks for computer-aided diagnosis: detection of lung nodules in chest radiograms.->Obtained From (DSS-3)
5501->pska: usable and secure key agreement scheme for body area networks.->Obtained From (DSS-3)
5502->collaborative work during interventional radiological procedures based on a multicast satellite-terrestrial network.->Obtained From (DSS-3)
5503->modeling of workflow-engaged networks on radiology transfers across a metro network.->Obtained From (DSS-3)
5504->mri fuzzy segmentation of brain tissue using neighborhood attraction with neural-network optimization.->Obtained From (DSS-3)
5505->using the timing information of heartbeats as an entity identifier to secure body sensor network.->Obtained From (DSS-3)
5506->a novel large-memory neural network as an aid in medical diagnosis applications.->Obtained From (DSS-3)
5507->artificial odor discrimination system using electronic nose and neural networks for the identification of urinary tract infection.->Obtained From (DSS-3)
5508->ieee 802.15.4 mac with gts transmission for heterogeneous devices with application to wheelchair body-area sensor networks.->Obtained From (DSS-3)
5509->neural-network compensation methods for capacitive micromachined accelerometers for use in telecare medicine.->Obtained From (DSS-3)
5510->nonparametric network design and analysis of disease genes in oral cancer progression.->Obtained From (DSS-3)

Number Of Results : 5510

Querying Duration : 16.24815441 secs for query : 'network'<

Figure 3. output of proposed system

VI. CONCLUSION

We begin the examination of the issue of looking short-lived outlines. We propose a direct yet expressive watchword based inquiry semantic structure that empowers transient information to be demonstrated as either predicates or situating factors. We propose a most ideal way iterator, which finds the "best" courses between two data center points in each time minute with respect to situating limits where the rank of a way is monotonically non-developing an edge expansion. By then we propose estimations to successfully survey this kind of request on a temporary outline to make top-k comes to fruition. The capability and suitability of the proposed approach are checked through wide observational examinations.

By using the hyper graph databases searching time has been reduced when compared to Graph databases. Dijkstra's algorithm is used in the hyper graph databases for better performance. The time complexity for searching a keyword has been reduced to half time.

VII. REFERENCES

[1]. Ziyang Liu, Chong Wang, and Yi Chen. Keyword Search on Temporal Graph 2016

[2]. 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.

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

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

[5]. R. Bin-Thalab, N. El-Tazi, and M. E. El-Sharkawi. TMIX: Temporal ModelforIndexingXMLDocuments. InAICCSA,pages1–8,2013.

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

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

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