

Keyword Search on Temporal Graphs

Abstract

Archiving graph data is important in many applications. In a mutual project, we would like to collect 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.

Existing system

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. One of the approaches is to run one of the existing graph keyword search methods on such a temporal graph to find candidate results oblivious to the timestamps, and then remove the invalid ones (i.e. the results where nodes and edges do not have a common timestamp) at the end.

Disadvantages:

- Such approaches suffers from low efficiency and low result quality
- Although invalid results are discarded by post-processing
- Much time may be wasted in generating such results
- Valid results may be missing.

Proposed system:

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.

Advantages:

- This approach gives good efficiency and good result quality
- Takes less time to generating such results
- Produce valid results.

System Configuration

Hardware Configuration:-

- ✓ Processor - Pentium –IV
- ✓ Speed - 1.1 Ghz
- ✓ RAM - 256 MB(min)
- ✓ Hard Disk - 20 GB
- ✓ Key Board - Standard Windows Keyboard
- ✓ Mouse - Two or Three Button Mouse
- ✓ Monitor - SVGA

Software Configuration:-

- ✓ Operating System : Windows XP
- ✓ Programming Language : JAVA
- ✓ Java Version : JDK 1.6 & above.
- ✓ Back end :MY SQL

www.takeoffprojects.com