Optimization of Bit Matrix Index for Temporal RDF

Tianyi Zhou, Yu Liu, Hang Zhang, Feng Gao, Xiaolong Zhang

CCKS2022

Introduction

- Temporal knowledge is crucial in many knowledge-based systems. Using existing RDF databases to manage temporal RDF data requires combining RDF Reification, which will increase the amount of data as well as data complexity.
- We provide an extension to bit matrix architecture (TBitStore) to support the indexing of temporal RDF. TBitStore constructs an index over both Subject-Object key and temporal information. Moreover, it leverages a temporal statistics-based index to optimize the query plans.

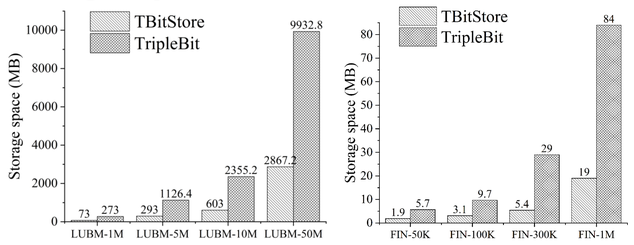
Method

1. We propose a novel storage structure on top of TripleBit that constructs SO (Subject-Object) keys by combining the subject and object of each predicate and associating SO keys with the temporal information, to store and retrieve temporal information. To save space, the system converts the time information into integers for storage and comparison.
2. To optimize query plan generation when processing temporal queries, we propose a time-oriented statistics index that divides the statistical information related to the predicate into multiple components which have different temporal intervals, and a 2-dimensional K-D tree is employed to enable the time-bounded search of all components.

Evaluation

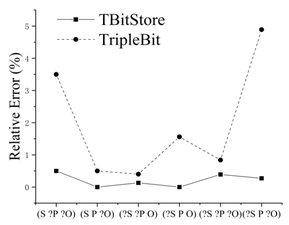
We compare and analyze the system with TripleBit database in terms of storage space occupation, selectivity estimation accuracy, and temporal query efficiency.

Storage Space



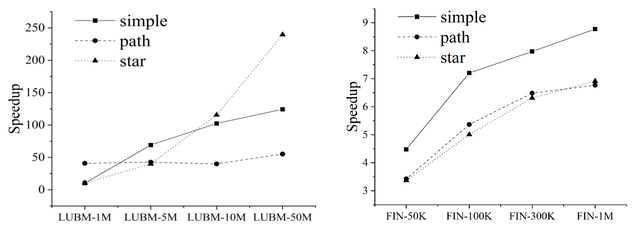
For the same dataset, the space occupied by TripleBit is larger than that of TBitStore.

- **Selectivity Estimation**



The Relative error of TBitStore and TripleBit for query pattern selectivity estimates.

- **Query Performance**



TBitStore improves the retrieval efficiency of three different types of temporal queries.

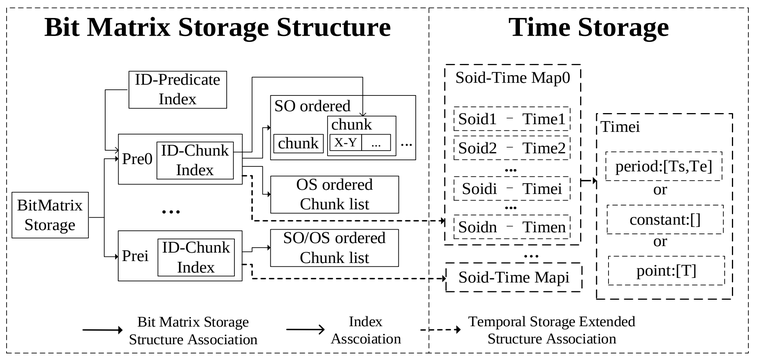
Conclusions

- Compared to the existing bit matrix database, the storage space is reduced by 50%-80%, while the time retrieval efficiency is improved by several times, up to two orders of magnitude.
- In future work, the system will be tested with more datasets and compared with more temporal storage systems. We will also further investigate the management of temporal hypergraph data.

Supplement

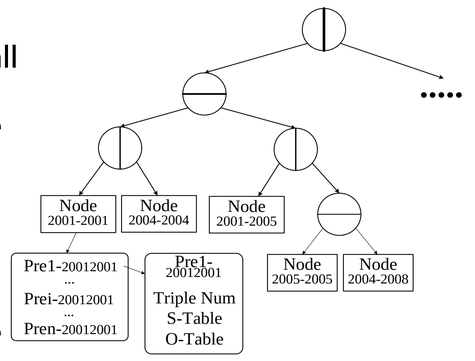
- **Storage Structure**

In terms of temporal RDF data storage, TBitStore applies the bit matrix model to manage the subject, predicate, and object data in temporal RDF data and adopts the key-value mapping approach to store and manage the temporal information of the triple.



- **Temporal Statistics Index**

The index uses the K-D tree to implement the range query for all predicate temporal components, and the tree is constructed based on temporal nodes with start and end times, and each node stores components with the same start and end time as the node.



- **Dataset**

Dataset source	Dataset name	Number of triples
LUBM	LUBM-1M	1316700
	LUBM-5M	5358624
	LUBM-10M	11108166
	LUBM-50M	53845210
FIN	FIN-50K	54200
	FIN-100K	106000
	FIN-300K	363216
	FIN-1M	1089604

The experiments use LUBM as well as the financial equity dataset (referred to below as FIN) as the test dataset. For the LUBM dataset, we add time information to the triple according to the semantics of each triple predicate. Both LUBM after adding time information and FIN are represented using RDF reification.

Tianyi Zhou:
zhoutianyi1998@foxmail.com
Yu Liu:
liuyu@wust.edu.cn

2022全国知识图谱与语义计算大会
China Conference on Knowledge Graph and Semantic Computing
秦皇岛 8.24-8.27