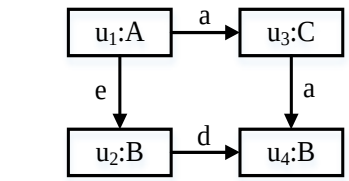
A Subgraph Matching Algorithm Based on Subgraph Index for Knowledge Graph

Yunhao SUN, Guanyu Li, Jingjing DU, Bo NING, Heng CHEN

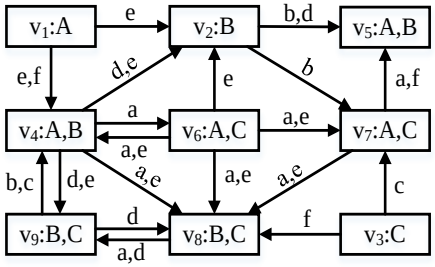
Frontiers of Computer Science, DOI: 10.1007/s11704-020-0360-y

Problems & Ideas

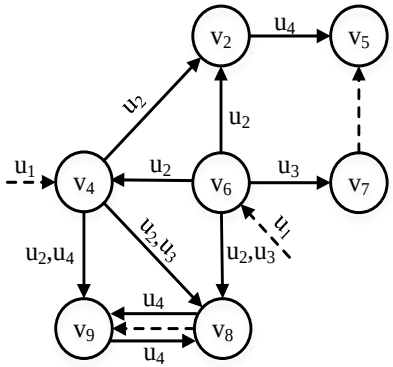
- Challenges of Subgraph Matching on Knowledge Graph
  - Subgraph Matching on knowledge graph has heavier redundant calculations than general graph, that is caused by the dense adjacent structure of multi-graph characteristic.
  - Dissimilar query vertices can cause a special candidate redundancy.
  - Subgraph Matching on knowledge graph requires a more complex evaluation for near-optimal matching trees than general graph.
- Our Approach: Sugraph Index and multi-label weight matrix
  - We develop a subgraph index of matching-driven flow graph to previously reducing the redundant calculations.
  - Based on multi-label weight matrix, we construct a heuristic algorithm to select a near-optimal matching tree for minimizing the intermediate results.



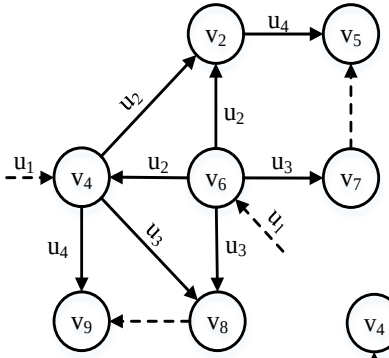
(a) Query Multi-Graph



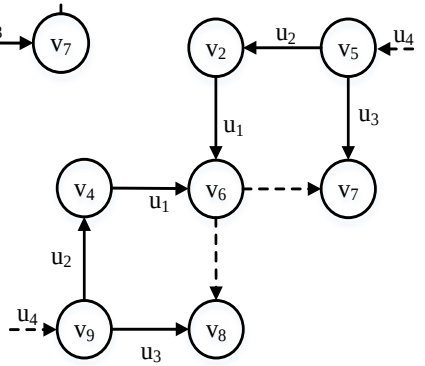
(b) Data Multi-Graph



(c) Top-Down Construction



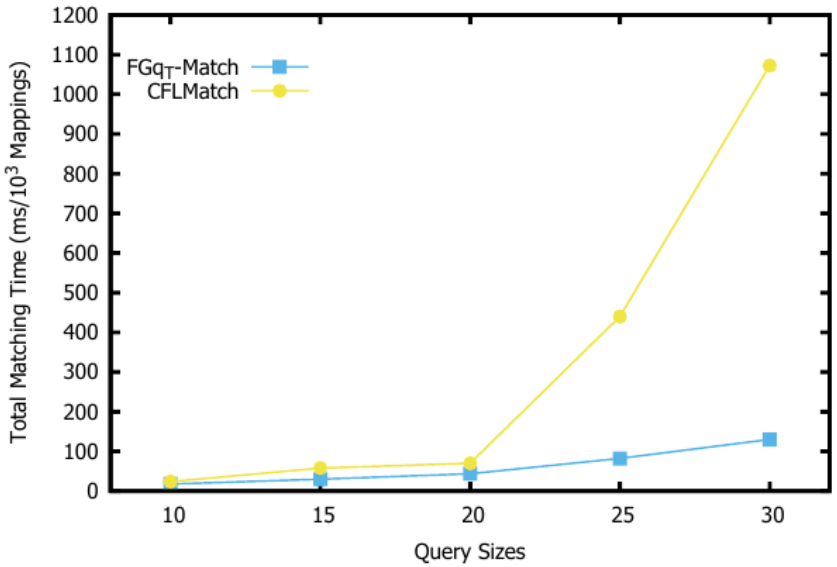
(d) Bottom-Up Refinement



(e) Subgraph index of different matching order

Experimental Result and Conclusions

- **Experimental Result**: In order to further evaluate the time-efficiency of algorithms, the total matching time of different query scales on DBpedia dataset is measured in Figure 21 (a). Experiments show that the total matching time-efficiency of FGqT-Match is 1 order of magnitude faster than CFLMatch on DBpedia dataset when query size reaches 30.



(a) Total Time on DBpedia

Fig.21: Total Time-Efficiency of Query Graphs on Dbpedia Dataset

- **Conclusions**: our algorithm of FGqT-Match has higher time-efficiency than the algorithms of CFLMatch and TurboHOM, and it benefits from the precise design of knowledge multi-graph. Experiments shown that the size of our FGqT is much smaller than compact path index of CFLMatch and the time-efficiency of our FGqT-Match is 1 order of magnitude faster than CFLMatch on DBpedia dataset when query size reaches 30. Further, experiments shows the multi-label evaluation for optimal matching tree is much more effective than the traditional infrequent strategy.