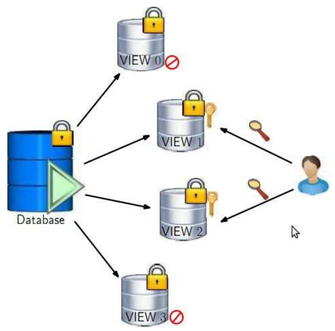
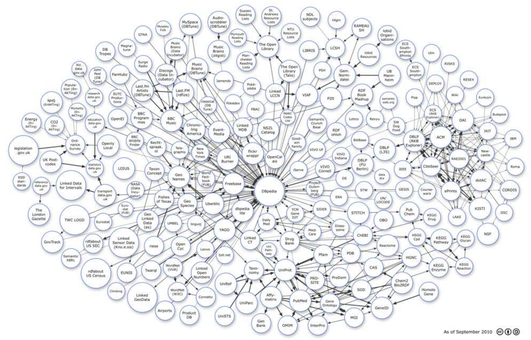
Scalable Multi-Query Optimization for SPARQL

Wangchao Le, Anastasios Kementsietsidis, Songyun Duan, Feifei Li

Motivation 1: Access Control for RDF data



Motivation 2: Web Data Integration on Query Endpoints



Two Types of SPARQL Queries for RDF Data

Type 1: Q := SELECT RD WHERE GP

Type 2: Q_{OPT} := SELECT RD WHERE GP (OPTIONAL GP_{OPT})$^+$

*GP: encoded by conjunctive triples: (sub pred obj).

*$GP_{OPT}{}^+$: left-join in relational database.

An Example to Use Left-Join in SPARQL

subj	pred	obj
p1	name	"Alice"
p1	zip	10001
p1	mbox	alice@home
p1	mbox	alice@work
p1	www	http://home/alice
p2	name	"Bob"
p2	zip	"10001"
p3	name	"Ella"
p3	zip	"10001"
p3	www	http://work/ella
p4	name	"Tim"
p4	zip	"11234"

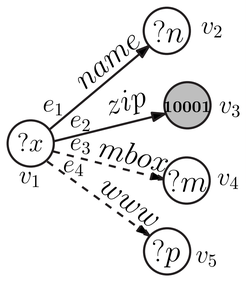
(a) Input data D

SELECT ?*name*, ?*mail*, ?*hpage*
WHERE { ?x name ?*name*, ?x zip 10001,
OPTIONAL {?x mbox ?*mail* }
OPTIONAL {?x www ?*hpage* }}

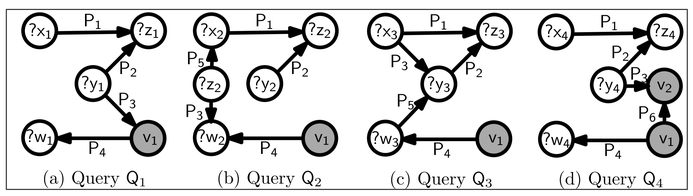
(b) Example query Q_{OPT}

name	mail	hpage
"Alice"	alice@home	
"Alice"	alice@work	
"Alice"		http://home/alice
"Bob"		
"Ella"		http://work/ella

(c) Output $Q_{OPT}(D)$

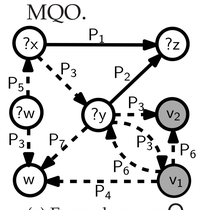


An Example to Evision The Optimization



(a) Query Q_1 (b) Query Q_2 (c) Query Q_3 (d) Query Q_4

Structure-based MQO.



(e) Example query Q_{OPT}

Seletivity

pattern p	α(p)
?x P$_1$?z	15%
?y P$_2$?z	9%
?y P$_3$?w	18%
?w P$_4$ v_1	4%
?t P$_5$ v_1	2%
v_1 P$_5$?t	7%
?w P$_6$?u	13%

(f) Selectivity

Rewrite 4 queries in 1 SPARQL query.

SELECT *
WHERE { ?x P$_1$?z, ?y P$_2$?z,
OPTIONAL {?y P$_3$?w, ?w P$_4$ v_1 }
OPTIONAL {?t P$_3$?x, ?t P$_5$ v_1, ?w P$_4$ v_1 }
OPTIONAL {?x P$_3$?y, v_1 P$_5$?y, ?w P$_4$ v_1 }
OPTIONAL {?y P$_3$?u, ?w P$_6$?u, ?w P$_4$ v_1 }
}

Blending selectivity in rewriting.

SELECT *
WHERE { ?w P$_4$ v_1,
OPTIONAL {?x_1 P$_1$?z_1, ?y_1 P$_2$?z_1, ?y_1 P$_3$?w }
OPTIONAL {?x_2 P$_1$?z_2, ?y_2 P$_2$?z_2, ?t_2 P$_3$?x_2, ?t_2 P$_5$ v_1 }
OPTIONAL {?x_3 P$_1$?z_3, ?y_3 P$_2$?z_3, ?x_3 P$_3$?y_3, v_1 P$_5$?y_3 }
OPTIONAL {?x_4 P$_1$?z_4, ?y_4 P$_2$?z_4, ?y_4 P$_3$?u_4, ?w P$_6$?u_4 }
}

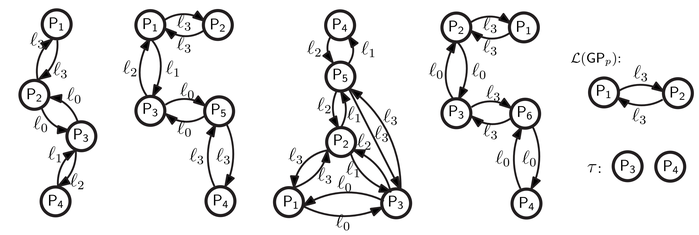
Finding Maximal Common Connected Substructures for SPARQL MQO

Theorem 1 *Given two graphs, finding the maximal common connected subgraphs amounts to finding the maximal common connected induced subgraphs in their linegraphs.*

Theorem 2 *Given two graphs, finding the maximal common connected induced subgraphs amounts to finding the maximal cliques with strong covering trees in their product graph.*

*__Challenges__: (I) deal with hundreds of graphs in one shot; (II) blend selectivity into the structure-based MQO.

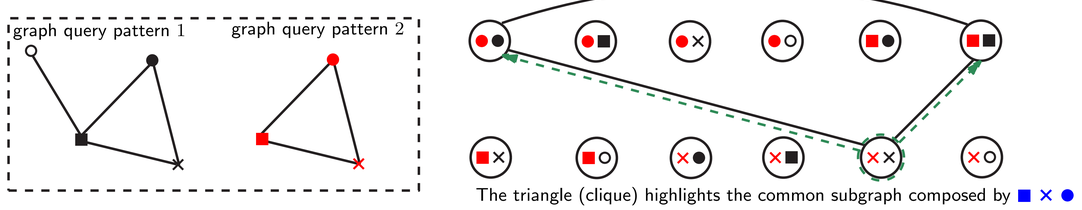
Build & Optimize Linegraphs and Product graph



(a) $\mathcal{L}(Q_1)$ (b) $\mathcal{L}(Q_2)$ (c) $\mathcal{L}(Q_3)$ (d) $\mathcal{L}(Q_4)$ (e) Subqueries

- Prune by missed matched constants.
- Mask every linegraph edge e with $\cap m^-[e]$=$\cap m^+[e]$'=$\emptyset$
- *e.g.*, $\cap m_i^+[P_3] = \cap m_i^-[P_3] = \emptyset$

Clique, Maximal Common Induced Subgraph and Strong Covering Tree

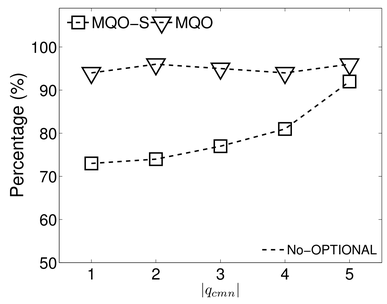


The triangle (clique) highlights the common subgraph composed by ■ × ●

Blend Cost into MQO

$$\text{Cost}(Q) = \begin{cases} \text{Min}(sel(t)) & Q \text{ is a } \textbf{Type 1} \text{ query}, t \in \text{GP} \\ \text{Min}(sel(t)) + \Delta & Q \text{ is a } \textbf{Type 2} \text{ query}, t \in \text{GP} \end{cases}$$

Observation: >90% of query evaluation time for our MQO is on evaluating the common structure (do it once for all queries), resulting in less time in evalut-ing the non-common substructures; while the pure strucuture-based MQO (MQO-S) is sensitive to the variances of common substructures rewritten, leading to more overhead in evaluating the non-common sub-structures.



Experiments on varying selectivity

Parameter	Symbol	Default	Range
Dataset size	D	4M	3M to 9M
Number of queries	$\|\mathcal{Q}\|$	100	60 to 160
Query size (num of trpl. patterns)	$\|Q\|$	6	5 to 9
Number of seed queries	κ	6	5 to 10
Size of seed queries	$\|q_{cmn}\|$	$\sim \|Q\|/2$	1 to 5
Max selectivity of patterns in Q	$\alpha_{max}(Q)$	random	0.1% to 4%
Min selectivity of patterns in Q	$\alpha_{min}(Q)$	1%	0.1% to 4%

MQO−S: MQO based on structure; MQO: Cost-based MQO; No-MQO: No MQO.