PRONA: A Plugin for Well-Designed Approximate Queries in Jena

Zhenyu Song^{1,3}, Xiaowang Zhang^{1,3,*}, and Zhiyong Feng^{2,3}

¹ School of Computer Science and Technology, Tianjin University, Tianjin, China ² School of Computer Software, Tianjin University, Tianjin, China

³ Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin, China * Corresponding author: xiaowangzhang@tju.edu.cn

Abstract. The time of answering a SPARQL query with its all exact solutions in large scale RDF dataset possibly exceeds users' tolerable waiting time, especially when it contains the OPT operations. It becomes essential to make a trade-off between the query response time and solution accuracy. We propose PRONA - an plugin for well-designed approximate queries in Jena, which provides help for users to answer well-designed SPARQL queries by approximate computation. The main features of PRONA comprise SPARQL query engine with approximate queries, as well as various approximate degrees for users to choose.

1 Introduction

Resource Description Framework (RDF) is the standard data model in the Semantic Web. SPARQL recommended by W3C has become the standard language for querying RDF data since 2008.

OPT operation takes an core role in UNION-free well-designed patterns. For simplification, we directly call well-designed patterns instead of UNION-free welldesigned patterns. OPT operation aims to extend solutions for users[3]. It may take more time to obtain all exact solutions than only "non-optional" solutions. Removing some "optional" parts of well-designed queries is a natural idea to obtain approximate queries, which contributes to less query response time. For instance, consider a pattern Q as follows:

Q = ((?x, rdf:type, artist) OPT ((?x, country, ?y) OPT (?x, company, ?z))).Based on this natural idea, there are two approximate patterns with less

OPT operators as follows:

 $\begin{array}{l} - \ Q_1 = (?x, \textit{rdf:type, artist}); \\ - \ Q_2 = ((?x, \textit{rdf:type, artist}) \ \text{OPT} \ (?x, \textit{country}, ?y)). \end{array}$

However, consider $Q_3 = ((?x, rdf:type, artist) \text{ OPT } (?x company, ?z))$, it is not approximate query since (?x, company, ?z) directly depends on (?x, country, ?y). The notion of approximation has been proposed in [1]. However, it did not provide a fine-grained approximation method. Jena^[2] is a free and open source Java framework for building semantic web and linked data applications. But it does not provide approximate queries for users to answer well-designed queries.

In this paper, we focus on well-designed SPARQL queries, whose "optional" parts are really optional. Moreover, it is maximal among all fragments of LSQ [4]. Compared to our previous work [5], furthermore, we develop a plugin for Jena to answer well-designed SPARQL queries with approximate queries, which combines our approximate method and query process in Jena.

2 Preliminaries

OPT Normal Form A UNION-free pattern P is in *OPT normal form* [3] if P meets one of the following two conditions:

- P is constructed by using only the AND and FILTER operators;
- $-P = (P_1 \text{ OPT } P_2)$ where P_1 and P_2 patterns are in OPT normal form. For instance, the pattern Q stated in Section 1 is in OPT normal form.

Three rewriting rules [3] can be applied to transform non OPT normal form into OPT normal form: let P, Q, R be patterns and C a constraint,

- $-(P \text{ OPT } R) \text{ FILTER } C \equiv (P \text{ FILTER } C) \text{ OPT } R;$
- $(P \text{ OPT } R) \text{ AND } Q \equiv (P \text{ AND } Q) \text{ OPT } R;$
- $P \operatorname{AND} (Q \operatorname{OPT} R) \equiv (P \operatorname{AND} Q) \operatorname{OPT} R.$

Well-Designed Patterns A UNION-free pattern P is well-designed if the followings hold:

- P is safe, that is, each subpattern of the form Q FILTER C of P holds the condition: $var(C) \subseteq var(Q)$.
- for every subpattern $P' = (P_1 \text{ OPT } P_2)$ of P and for every variable ?x occurring in P, the following condition hold: If ?x occurs both inside P_2 and outside P', then it also occurs in P_1 .

For instance, the pattern Q in Section 1 is a well-designed pattern.

Note that the OPT operation provides really optional left-outer join due to the weak monotonicity [3].

3 Approximate Queries

OPT-depth in OPT Normal Form To characterize the different levels of optional patterns, we define *OPT-depth* of patterns in OPT normal form.

Definition 1 (OPT-depth). Let P be a pattern in OPT normal form. We use dep(P) to denote its OPT-depth as follows:

- dep(P) = 0 if P is an AF-pattern;

 $- dep(P) = \max\{dep(P_1), \dots, dep(P_m)\} + 1 \text{ if } \mathcal{O}(P) = \{P_1, \dots, P_m\}.$

For instance, the OPT-depth of the pattern Q stated in Section 1 is 2.

Approximate Queries Intuitively, approximate patterns are subpatterns obtained by reducing their OPT-depths.

Definition 2 (k-approximation). Let P be a pattern in OPT normal form $(P_0 \text{ OPT } P_1 \text{ OPT } \dots \text{ OPT } P_m)$ and k be a natural number. The k-approximate pattern of P (written as $P^{(k)}$) can be obtained in the following inductive way:

- $P^{(k)} = BGP(P)$ if k = 0;
- $P^{(k)} = P_0 \text{ OPT } P_1^{(k-1)} \text{ OPT } \dots \text{ OPT } P_m^{(k-1)} \text{ if } 1 \le k \le dep(P) 1; \\ P^{(k)} = P \text{ if } k \ge dep(P).$

 $P^{(k)}$ is more closed to P with higher value of k. For instance, in Section 1, $Q^{(0)} = Q_1$ and $Q^{(1)} = Q_2$.

PRONA Plugin 4

PRONA Overview PRONA is written in Java in a 3-tier design shown in Figure 1(a). The bottom layer consists of the Jena framework and the ARQ^4 query engine, both used as a black box for evaluating queries. Before answering SPAR-QL queries, the second layer provides the rewriting process and approximation evaluation, which lead to the generation of approximate queries.

GUI is shown in Figure 1(b). For single query solution s, we denote its amount of domain as dom(s). For instance, consider solution $\mu = \{?x \to a, ?y \to b\},\$ dom(μ)=2. Given an approximate query solution S, which contains solution s_1 , s_2, \dots, s_n . It is notable that the total of domain $(\operatorname{dom}(s_1) + \operatorname{dom}(s_2) + \dots + \operatorname{dom}(s_n))$ $\operatorname{dom}(s_n)$) reflects solution precision.

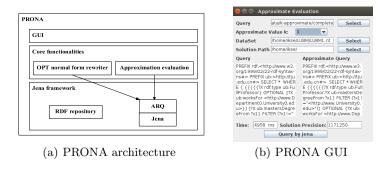


Fig. 1. PRONA Overview

Experiments The purpose of our experiments is to evaluate (1) the performance improvement of approximate well-designed SPARQL queries, and (2) the solution precision percentage after approximate queries.

In our experiments, LUBM⁵ is used as dataset. Two 4-approximation welldesigned SPARQL queries are designed. Q_1 contains 4 OPT operators, 6 triple patterns and 6 variables. Q_2 contains 14 OPT operators, 17 triple patterns and 16 variables. Approximate solution precision dividing original query solution precision leads to the solution precision percentage.

It has shown in Figure 2 and Figure 3 that both query response time and solution precision reduce with the increment of approximate degree (k value

⁴ http://jena.sourceforge.net/ARQ

⁵ http://swat.cse.lehigh.edu/projects/lubm

decreases). Solution precision percentage decreases about 10% with 25% query response time decreasing when k changes from 4 to 3 in Figure 3.

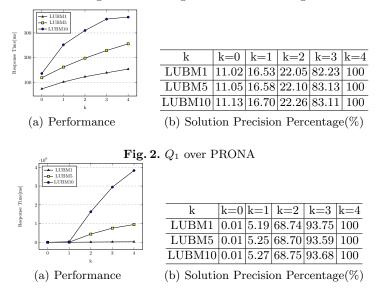


Fig. 3. Q_2 over PRONA

5 Conclusion

In this paper, we propose PRONA which helps users answer well-designed SPAR-QL queries by approximate computation. In the future, we are going to handle other non-well-designed patterns and deal with more operations such as UNION.

Acknowledgement. This work is supported by the program of the National Key Research and Development Program of China (2016YFB1000603) and the National Natural Science Foundation of China (NSFC) (61502336).

References

- P. Barcelo, R. Pichler, and S. Skritek. Efficient evaluation and approximation of well-designed pattern trees. In *Proc. of PODS 2015*, pages 131–144, 2015.
- J. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson. Jena: implementing the semantic web recommendations. In *Proc. of WWW 2004*, pages 74–83, 2004.
- J. Prez, M. Arenas, and C. Gutierrez. Semantics and complexity of SPARQL. ACM Transactions on Database Systems, 34(3):30–43, 2009.
- M. Saleem, M. I. Ali, A. Hogan, Q. Mehmood, and A.-C. N. Ngomo. LSQ: The linked SPARQL queries dataset. In Proc. of ISWC 2015, pages 261–269, 2015.
- Z. Song, X. Zhang, Z. Feng, X. Wang, and G. Rao. LSQ: The linked SPARQL queries dataset. In Proc. of SemiBDMA 2016, to appear, 2016.