ANNA: Answering Why-Not Questions for SPARQL

Siyu Yao, Jun Liu, Meng Wang, Bifan Wei, Xuelu Chen

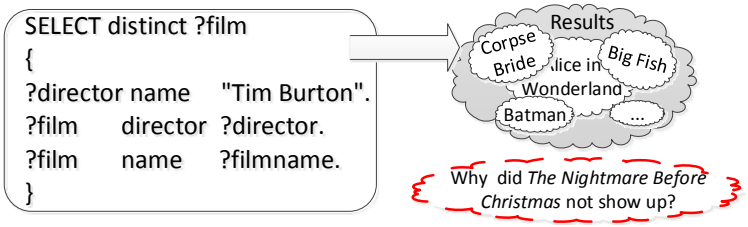
MOEKLINNS Lab, Department of Computer Science,
Xi'an Jiaotong University,710049, China
cheryl@stu.xjtu.edu.cn, liukeen@mail.xjtu.edu.cn, wangmengsd@stu.xjtu.edu.cn,
weibifan@mail.xjtu.edu.cn, shirley.chen0217@gmail.com

Abstract. Considerable effort has been made to improve the functionality and usability of SPARQL search engines. However, explaining missing items in the results of SPARQL queries or the so-called why-not questions remains in its infancy. Existing explanation models cannot be trivially extended to SPARQL queries because of the SPARQL-specific features in the data model and query operations. In this demonstration, we present a novel explanation system, ANNA (Answering why-Not questioNs for spArql), to explain why-not questions using a divide-and-conquer strategy. ANNA can visualize explanations to help users revise their initial queries to make the expected result-items presented. Experimental results on DBpedia prove that ANNA can generate high-quality explanations within a reasonable amount of time.

Keywords: Why-Not, SPARQL, RDF Graph, Query, Basic Graph Pattern

1 Introduction

Given that writing SPARQL queries is an error-prone and tedious task, users often make mistakes or cannot obtain the expected results. When such situations happen, users will naturally ask a question, specifically, a why-not question. For example, a user wants to find all films directed by *Tim Burton*. Therefore, the user submits a SPARQL query over DBpedia[1], as shown in Fig. 1(a). However, the results confuse the user. Why did *The Nightmare Before Christmas* not show up?



(a) A SPARQL query (b) A why-not question

Fig. 1. SPARQL query and query results.

Various possibilities may be considered to answer the why-not question shown in Fig. 1(b). The film may not be directed by *Tim Burton*, or the film does not have the *director* property in DBpedia. The user may find determining the real answer difficult and can hardly sift through the initial SPARQL query.

[1] http://wiki.dbpedia.org/Datasets, released in September, 2014

This situation illustrates the significance of our system, namely, Answering why-Not questioNs for spArql (ANNA[2]). Many explanation models have been created to answer why-not questions for relational databases, social image searches and top-k queries [1–3]. The data model of SPARQL queries is the Resource Description Framework (RDF), and query operations are based on graph pattern matching. The differences in these two aspects make existing models unable to be trivially extended to SPARQL queries. ANNA can generate corresponding explanations according to the given why-not questions. ANNA initially identifies which parts of a SPARQL query should be responsible for removing the expected items and then generates explanations using a divide-and-conquer strategy. With the help of the explanations returned by ANNA, users can refine their initial SPARQL queries.

2 Preliminary

A SPARQL query Q consists of triple patterns and operators (*FILTER*, *DISTINCT*, *MINUS*, *LIMIT*, *ORDER BY*, etc.). The evaluation of Q over the RDF dataset DS can be divided into two levels, namely, basic graph pattern (BGP) level and operator level. At the BGP level, the BGP P of Q is evaluated to match the RDF graphs in DS. If $P(DS) \neq \varnothing$, then the operators use $P(DS)$ to provide the query result $Q(DS)$.

Given $DS, Q, Q(DS)$, we represent a why-not question as a mapping $v \to s$, where v is a variable in Q, and the RDF term s is a solution of v. A mapping $v \to s$ indicates why an RDF item s does not appear in $Q(DS)$.

An explanation ψ represents the reason for a why-not question $v \to s$. The explanation for the absence of an item s is given in the following two forms: (1) **A modified BGP**, which is similar to the original BGP. The modified BGP should match an RDF graph from DS with a variable v bound to s. (2) **A set of tuples**, which is denoted by $\{(op_i, m_i)\}_l$. Each tuple (op_i, m_i) indicates a questionable query operator op_i and the corresponding matched RDF graph m_i that contains the expected item s.

3 ANNA

After analyzing the SPARQL query evaluation, we find that **restrictive BGP expressions** (BGP level) and **questionable query operators** (operator level) are the two reasons why the expected items may be absent from the query result. Accordingly, ANNA is designed to address why-not questions using a divide-and-conquer strategy. Figure 2 shows the ANNA framework, which consists of three modules.

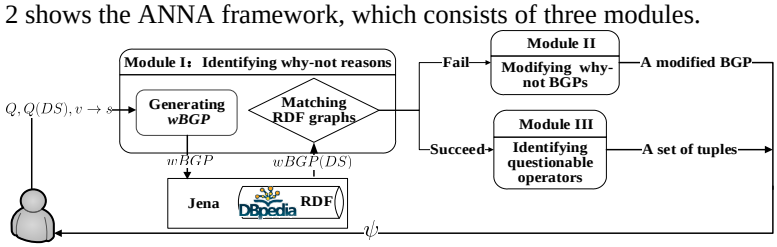


Fig. 2. ANNA framework.

[2] Demonstration is available online at http://kfm.skyclass.net/anna/index.jsp

Module I Identifying Why-not Reasons: This module identifies the level from which the expected item s is removed in a two-step process.
a) All the variables of BGP P are replaced in accordance with $v \rightarrow s$ to generate a why-not BGP $wBGP$. In consideration of the SPARQL query in Section 1, the variable $?filmname$ is adjusted to *The Nightmare Before Christmas* in accordance with $?filmname \rightarrow The\ \ Nightmare\ \ Before\ \ Christmas$.
b) $wBGP$ is matched to DS (the dataset for ANNA is the DBpedia data stored by Jena TDB[3]). If $wBGP(DS) \neq \varnothing$, then the why-not reason is located at the operator level; otherwise, it is located at the BGP level.
Module II Modifying Why-not BGPs: This module aims to identify and modify the inappropriate triple patterns in $wBGP$, which are blamed for $wBGP(DS) = \varnothing$. ANNA generates a modified why-not BGP $wBGP'$ via a graph-based approach, as follows:
a) Each triple pattern t_j of $wBGP$ is added to $wBGP'$ initialized as $\varnothing$ by a biased breadth-first traversal over the line graph [4] of $wBGP$. When each t_j is added, ANNA matches $wBGP'$ over DS. Therefore, we implement a heuristic rule, Equation (1), to select t_j to improve the efficiency of $wBGP'$ matching.

$$\langle s_j, p_j, o_j\rangle \prec \langle ?, p_j, o_j\rangle \prec \langle s_j, p_j, ?\rangle \prec \langle s_j, ?, o_j\rangle \prec \langle ?, p_j, ?\rangle \prec \langle ?, ?, o_j\rangle \prec \langle s_j, ?, ?\rangle \prec \langle ?, ?, ?\rangle \quad \textbf{(1)}$$

b) If $wBGP'(DS) = \varnothing$ after adding t_j to $wBGP'$, then t_j is replaced with a modified t'_j, which is computed by the query relaxation approach proposed in [5]. The left of $wBGP$ is then added to $wBGP'$. If $wBGP'(DS) \neq \varnothing$, then the traversal is completed, else return to step a.
Module III Identifying Questionable Operators: This module aims to address why-not questions at the operator level. Questionable query operators are filtered out, and ψ is returned and denoted by $\{(op_i, m_i)\}_l$. The main procedures are as follows:
a) A SPARQL operator tree $\tau(Q)$ is constructed by parsing query Q according to [6].
b) A set of operators, OP, is generated from $\tau(Q)$ by a post-order traversal on $\tau(Q)$.
c) For each $op \in OP$ and each matched RDF graph $m \in wBGP(DS)$, if any subgraphs of m do not belong to $opGraph$, which is the output of op, then op filters out m from the query processing. The tuple (op, m) is subsequently added to ψ.

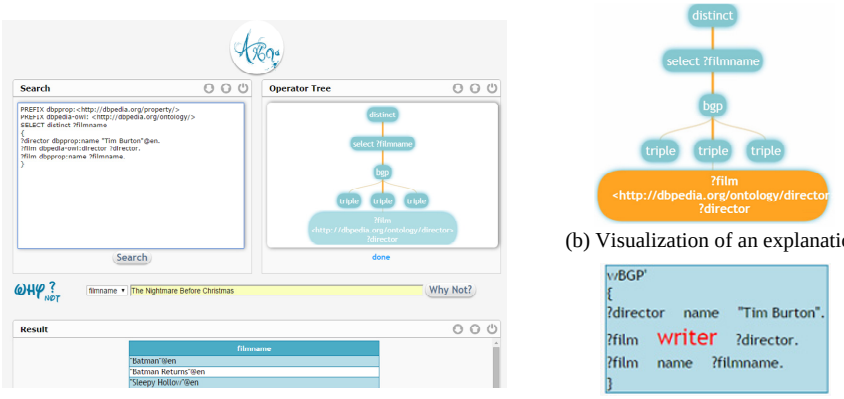
4 Demonstration

The entire system is performed through a web application written in Java. We briefly illustrate how ANNA works through the preceding example.

The user submits a query using the search panel shown in Fig. 3(a). After the results return, the user can pose a why-not question $v \rightarrow s$. The procedures are as follows:
(i) Select v from the drop-down menu (e.g., $?filmname$).
(ii) Fill in the blank with s (e.g., $The\ \ Nightmare\ \ Before\ \ Christmas$).

The explanation generated by ANNA is returned as shown in Fig. 3(b), and is highlighted in the operator tree shown in Fig. 3(c). For the preceding example, the explanation is a modified BGP generated from $wBGP$ as $director$ is replaced with $writer$.

[3] http://jena.apache.org



(a)A screenshot of ANNA for submitting a why-not question

(b) Visualization of an explanation

(c) An explanation

Fig. 3. Demonstration of ANNA

A total of 61 why-not questions are obtained from 42 SPARQL queries[4] to evaluate the effectiveness and efficiency of ANNA. The satisfaction of the explanations is measured by a five-point Likert scale, and 76.5% of the explanations are considered *strongly agree*. The experimental results prove that ANNA can generate high-quality explanations within a reasonable amount of time at both BGP (approximately 5 s) and operator levels (approximately 1.8 s).

5 Conclusion and Future Work

For the first time, we develop a novel explanation system called ANNA. Two main lines are prioritized in future work. First, we aim to transform ANNA into a Java library that can be extended to any RDF database. Second, we intend to utilize union and optional graph patterns to address why-not questions for SPARQL queries.

Acknowledgements
The research was supported in part by the Doctoral Fund of Ministry of Education of China under Grant No. 20130201130002 and No. IRT13035.

References

1. M. Herschel, and M. A. Hernández, Explaining missing answers to SPJUA queries, in PVLDB, pages 185-196, 2010.
2. S. S. Bhowmick, A. Sun, and B. Q. Truong, Why not, WINE?: towards answering why-not questions in social image search, in ACM MM, pages 917-926, 2013.
3. Z. He, and E. Lo, Answering why-not questions on top-k queries, IEEE Transactions on Knowledge and Data Engineering, 26 (6): 1300-1315, 2014.
4. Khmelnitskaya and A. B., Values for rooted-tree and sink-tree digraph games and sharing a river[J]. Theory & Decision, 69(4): 657-669, 2010.
5. C. A. Hurtado, A. Poulovassilis, and P. T. Wood, A relaxed approach to RDF querying, presented at the ISWC, 2006.
6. J. Pérez, M. Arenas, and C. Gutierrez, "Semantics and complexity of SPARQL," *ACM Transactions on Database Systems*, vol. 34, pages 1-45, 2009.

[4] http://kfm.skyclass.net/anna/queryset.html