Bridging the Semantic Gap between RDF and SPARQL using Completeness Statements

Fariz Darari, Simon Razniewski, and Werner Nutt

Faculty of Computer Science, Free University of Bozen-Bolzano, Italy
fariz.darari@stud-inf.unibz.it,{razniewski,nutt}@inf.unibz.it

Abstract. RDF data is often treated as incomplete, following the Open-World Assumption. On the other hand, SPARQL, the standard query language over RDF, usually follows the Closed-World Assumption, assuming RDF data to be complete. This gives rise to a semantic gap between RDF and SPARQL. In this paper, we address how to close the semantic gap between RDF and SPARQL in terms of certain answers and possible answers using completeness statements.

Keywords: SPARQL, RDF, data completeness, OWA, CWA

1 Introduction

Due to its open and incremental nature, data on the Semantic Web is generally incomplete. This can be formalized using the Open-World Assumption (OWA)[1]. SPARQL, on the other hand, interprets RDF data under closed-world semantics. While for positive queries, this does not create problems [2], SPARQL queries with negation make only sense under closed-world semantics. This ambiguous interpretation of RDF data poses the question how to determine which semantics is appropriate in a given situation.

As a particular example, suppose we want to retrieve all Oscar winners that have tattoos. For obvious reasons, no sources on the Semantic Web contain complete information about this topic and hence the Open-World Assumption applies. On the other hand, suppose we want to retrieve all Oscar winners. Here, an RDF version of IMDb[1] would contain complete data and hence one may intuitively apply closed-world reasoning.

In [3], completeness statements were introduced, which are metadata that allow one to specify that the CWA applies to parts of a data source. We argue that these statements can be used to understand the meaning of query results in terms of certain and possible answers [4], which is especially interesting for queries with negation: Suppose an RDF data source has the completeness statement "complete for all Oscar winners". The result of a SPARQL query for Oscar winners contains not only all certain but also all possible answers. The result of a SPARQL query for Oscar winners with tattoos contains only all certain

[1] http://www.imdb.com/oscars/

answers. Moreover, the result of a SPARQL query for people with tattoos that did not win an Oscar contains all certain answers, but not all possible answers. The result of a SPARQL query for Oscar winners not having tattoos contains all possible answers but none of the answers is certain.

In this paper, we discuss how to assess the relationship between certain answers, possible answers and the results retrieved by SPARQL queries in the presence of completeness statements.

2 Formalization

SPARQL Queries. A basic graph pattern (BGP) is a set of triple patterns [5]. In this work, we do not consider blank nodes. We define a *graph pattern* inductively as follows: (1) a BGP P is a graph pattern; (2) for a BGP P, a `NOT-EXISTS` pattern $\neg P$ is a graph pattern; (3) for graph patterns P_1 and P_2, $P_1 \texttt{ AND } P_2$ is a graph pattern. Any graph pattern P can be equivalently written as a conjunction of a BGP (called the positive part and denoted as P^+) and several `NOT-EXISTS` patterns $\{ \neg P_1, \ldots, \neg P_n \}$ (referred to as the negative part and denoted as P^-). A query with negation has the form $Q = (W, P)$ where P is a graph pattern and $W \subseteq \mathit{var}(P^+)$ is the set of distinguished variables. The evaluation of a graph pattern P over a graph G is defined in [5]:

$$[\![P^+ \texttt{ AND } \neg P_1 \texttt{ AND } \ldots \texttt{ AND } \neg P_n]\!]_G = \{ \mu \mid \mu \in [\![P^+]\!]_G \wedge \forall i \,.\, [\![\mu(P_i)]\!]_G = \emptyset \}$$

The result of evaluating (W, P) over a graph G is the restriction of $[\![P]\!]_G$ to W. This fragment of SPARQL queries is safe, that is, for a query with negation Q and a graph G, the evaluation $[\![Q]\!]_G$ returns finite answers. We assume all queries to be consistent, that is, there is a graph G where $[\![Q]\!]_G \neq \emptyset$.

Completeness Statements. Formally, a *completeness statement* C is of the form $\mathit{Compl}(P_1 | P_2)$ where P_1, called the pattern, and P_2, called the condition, are BGPs. Intuitively, such a statement expresses that the source contains all instantiations of P_1 that satisfy condition P_2 (e.g., all Golden Globe winners that won an Oscar).

In line with the Open-World Assumption of RDF, for a graph G, we call any graph G' such that $G' \supseteq G$ an *interpretation* of G [2]. We associate to a completeness statement C the `CONSTRUCT` query $Q_C = (\texttt{CONSTRUCT } \{ P_1 \} \ \{ P_1 \texttt{ AND } P_2 \})$. A pair (G, G') of a graph and one of its interpretations *satisfies* a completeness statement C, written $(G, G') \models C$ if $[\![Q_C]\!]_{G'} \subseteq G$ holds. It satisfies a set $\mathcal{C}$ of completeness statements, written $(G, G') \models \mathcal{C}$ if it satisfies every element in $\mathcal{C}$. A set of completeness statements $\mathcal{C}$ *entails* a statement C, written $\mathcal{C} \models C$, if for all pairs (G, G') of a graph G and an interpretation G' of G such that $(G, G') \models \mathcal{C}$, it is the case that $(G, G') \models C$.

Completeness statements restrict the set of interpretations:

Definition 1 (Valid Interpretation with Completeness Statements). *Let G be a graph and $\mathcal{C}$ be a set of completeness statements. An interpretation $G' \supseteq G$ is valid w.r.t. $\mathcal{C}$ iff $(G, G') \models \mathcal{C}$. We write the set of all valid interpretations of G w.r.t. $\mathcal{C}$ as $\mathit{int}(G, \mathcal{C})$.*

Definition 2 (Certain and Possible Answers with Completeness Statements). *Let Q be a query, G be a graph and $\mathcal{C}$ be a set of completeness statements. Then the certain and possible answers of Q over G w.r.t. $\mathcal{C}$ are $CA_{\mathcal{C}}^{Q}(G) = \bigcap_{G' \in int(G,\mathcal{C})} [\![Q]\!]_{G'}$ and $PA_{\mathcal{C}}^{Q}(G) = \bigcup_{G' \in int(G,\mathcal{C})} [\![Q]\!]_{G'}$, respectively.*

If $\mathcal{C}$ is empty, then $PA_{\mathcal{C}}^{Q}(G)$ and $CA_{\mathcal{C}}^{Q}(G)$ correspond to the classical certain and possible answers [4]. With completeness statements, some query results may correspond to the certain and possible answers while others may not.

Example 1 (Possible Answers). Consider again the query asking for people that have tattoos. Since this is private information we have no knowledge how complete our data source is, and hence the set of possible answers is (nearly) infinite: Anyone not explicitly stated to have tattoos might still have tattoos.

On the other hand, consider the query for people who won an Oscar. It is relatively easy for a data source to be complete on this topic, by comparing: e.g., to the Internet Movie Database (IMDb). If a data source is complete for all Oscar winners, then there are no further possible answers.

The reasoning for queries with negation is more complex. By [2], under the OWA, for a monotonic query Q and a graph G the certain answers are $[\![Q]\!]_G$, while for queries with negation, the certain answers are empty. With completeness statements, the certain answers of a query with negation can be non-empty.

Example 2 (Certain Answers). Consider first a query for people that won an Oscar but no Golden Globe. If a data source is complete both for Oscar winners and Golden Globe winners, then the query result contains all possible and all certain answers.

On the other hand, a query for people with tattoos that did not win an Oscar would only return certain answers, but not all possible answers, because there are probably many more people with tattoos that did not win an Oscar.

The result of a query for people that won an Oscar and do not have tattoos contains all possible answers but no certain answers, because we do not know for certain which Oscar winners have tattoos and which do not.

We next define for each query some completeness statements that allow one to capture the crucial information for getting certain or possible answers. Knowing about the crucial statements helps in data acquisition identify which data is needed in order to achieve desired answer semantics.

Definition 3 (Crucial Statements). *For a query $Q = (W, P)$, the* positive crucial statement *of Q, denoted as C_Q^{+}, is the statement $Compl(P^{+} | \texttt{true})$. The* set of negative crucial statements *of Q, denoted as $\mathcal{C}_Q^{-}$, is the set $\{ Compl(P_1 | P^{+}), \ldots, Compl(P_n | P^{+}) \}$, where $P_1, \ldots, P_n$ are from the negative part P^{-} of Q.*

The next theorems show that the crucial statements can be used to infer relationships between certain answers, possible answers and SPARQL query results.

Theorem 1 (Bounded Possible Answers). *Let $\mathcal{C}$ be a set of completeness statements and Q be a positive query. Then*

$$\mathcal{C} \models C_Q^+ \qquad \textit{implies} \qquad \textit{for all graphs } G,\ \ PA_{\mathcal{C}}^Q(G) = CA_{\mathcal{C}}^Q(G)\ (= [\![Q]\!]_G).$$

While the equality $[\![Q]\!]_G = CA_{\mathcal{C}}^Q(G)$ always holds, the new insight is that $[\![Q]\!]_G = PA_{\mathcal{C}}^Q(G)$. This means the query results cannot miss any information w.r.t. reality.

Theorem 2 (Queries with Negation). *Let $\mathcal{C}$ be a set of completeness statements and Q be a query. Then*

1. $\mathcal{C} \models \mathcal{C}_Q^-$ *implies* *for all graphs G, $CA_{\mathcal{C}}^Q(G) = [\![Q]\!]_G$;*
2. $\mathcal{C} \models C_Q^+ \wedge \mathcal{C}_Q^-$ *implies* *for all graphs G, $PA_{\mathcal{C}}^Q(G) = CA_{\mathcal{C}}^Q(G) = [\![Q]\!]_G$.*

The first item means that if $\mathcal{C} \models \mathcal{C}_Q^-$, then every answer returned by the query is a certain answer. The second item means that if additionally $\mathcal{C} \models C_Q^+$, then there also cannot be any other possible answers than those returned by $[\![Q]\!]_G$. The completeness statement entailment problems can be solved using standard query containment techniques [6].

3 Discussion

We have shown that in the presence of completeness statements, the semantics of SPARQL may correspond to the certain answer or possible answer semantics. Our work is based on the observation that parts of data on the Semantic Web are actually complete. In future research, we would like to consider explicit negative RDF knowledge and completeness statements over it as an alternative for getting certain and possible answers. In this work, we assume all information contained in a graph is correct. An interesting case is when this assumption does not hold in general. We would like to investigate correctness statements as the dual of completeness statements. A further study is also needed for an effective way of maintenance of completeness statements to cope with information changes. One possible way is to add timestamps to completeness statements. An extended version with proofs of this paper is available at `http://arxiv.org/abs/1408.6395`.

References

1. Raymond Reiter. *On Closed World Data Bases.* 1977.
2. Marcelo Arenas and Jorge Pérez. Querying Semantic Web Data with SPARQL. In *PODS*, 2011.
3. Fariz Darari, Werner Nutt, Giuseppe Pirrò, and Simon Razniewski. Completeness Statements about RDF Data Sources and Their Use for Query Answering. In *ISWC 2013*, pages 66–83. Springer Berlin Heidelberg, 2013.
4. Serge Abiteboul, Paris C. Kanellakis, and Gösta Grahne. On the Representation and Querying of Sets of Possible Worlds. *Theor. Comput. Sci.*, 78(1):158–187, 1991.
5. Steve Harris and Andy Seaborne. SPARQL 1.1 Query Language. Technical report, W3C, 2013.
6. Simon Razniewski and Werner Nutt. Completeness of Queries over Incomplete Databases. *PVLDB*, 4(11):749–760, 2011.