Virtualized Ontology Query By Example

Lucas Peres[1], Ticiana L. Coelho da Silva[1], Jose Macedo[1], and David Araujo[1]

Insight Data Science Lab, Fortaleza - CE, BR
{lucasperes,ticianalc,jose.macedo,david}@insightlab.ufc.br
http://www.insightlab.ufc.br

Abstract. The Web has evolved to a large variety of data usually published in RDF from multiple domains. A recurrent problem in recent literature concerns to perform a search over RDF instead of using structured queries in triple-pattern-based languages like SPARQL, which only expert programmers can precisely specify their information needs. In this paper, we propose Von-QBE, an open source tool to query over RDF databases without any technical knowledge about RDF or the queried ontology structure. This differs from the-state-of-art tools by being schema-based instead of instance-based. It can be impracticable to use instance-based approaches in big data scenarios where the RDF data is huge and demands lots of computational resources to keep the knowledge base in memory. Moreover, most of these solutions need the knowledge base materialized into RDF(or *triplified*), which can be costly for legacy bases. We present various demonstration scenarios using the IMDB movie ontology.

Keywords: RDF schema · SPARQL query · Query by Example

1 Introduction

The Web has evolved from a network of linked documents to one where both documents and data are linked, resulting in what is commonly known as the Web of Linked Data, that includes a large variety of data usually published in RDF from multiple domains. As any database model, RDF requires formal query languages to retrieve information. Tools that perform a search over RDF data become increasingly important since writing structured queries in triple-pattern-based languages like SPARQL [8] can be extremely difficult for non-technical users.

Consider the example question, such as "Find the title of action movies produced in Northern America and the name of their company". A possible SPARQL query formulation, assuming a user familiar with the schema of the underlying knowledge base and knows which entities are present at the data instances, could consist of the following:

```
SELECT DISTINCT ?x ?title ?company_name
WHERE {
```

```
   ?x a mo:Movie; mo:title ?title;
    mo:isProducedBy ?y; mo:belongsToGenre [ a mo:Brute_Action ] .
   ?y :companyName ?company_name; :hasCompanyLocation [ a mo:Northern_America ] .
}
```

This complex query requires a user's familiarity with the knowledge base, which in general, no user (technical or not) should be expected to have. Basically, Von-QBE (stands for Virtualized Ontology Query By Example) addresses the problem: Given a natural language question Q_N and an underlying ontology O, its goal is to translate Q_N into a formal query Q_S as SPARQL that captures the information need to be expressed by Q_N. Von-QBE focuses on queries that emphasize classes and relations between them, not considering aggregation, disjunctive and negation queries.

A considerable number of question answering approaches for RDF data has been proposed, to name a few [6], [1], [7], and [9]. They address the same problem of Von-QBE, however, they are instance-based approaches, which can be unfeasible in big data scenarios where the RDF data is huge and demands lots of computational resources to keep the knowledge base in memory. Moreover, most of these solutions need the knowledge base materialized into some RDF format(triplified), which can be a hard task for legacy bases.

[5] describes Von-QBE in details. Von-QBE derives from the term virtual ontology, since it is not instance-based and it can use a virtualized ontology by tools like Ontop[2] instead of materialized RDF stores like Virtuoso[1]. All in all, Von-QBE is an open-source [2] schema-based approach to query over RDF data without any previous knowledge about the ontology or RDF technical skills. It lets the user queries using natural language questions or keyword search and translates the search into a SPARQL formal query. Furthermore, Von-QBE assists the user to construct his/her query search interactively, providing examples.

A screencast of Von-QBE is available at YouTube [3]. To the best of the authors' knowledge, this work is the first that addresses the problem of question answering over RDF using only the schema. In the next section, we present the main components of Von-QBE. In Section 3, we present the demonstration scenarios. Section 4 draws the final conclusions.

2 Von-QBE

Von-QBE addresses the problem of question answering over RDF. Beyond that, it also helps the user to construct its search interactively, providing examples. Figure 1 shows Von-QBE architecture which comprises three main components: *Fragment Extractor*, *Fragment Expansor* and *Query Builder*. All these components handle the ontology schema as an RDF graph[3], allowing the usage of

[1] https://virtuoso.openlinksw.com/
[2] http://github.com/insightlab/von-qbe
[3] https://youtu.be/ScXgGzhbx50

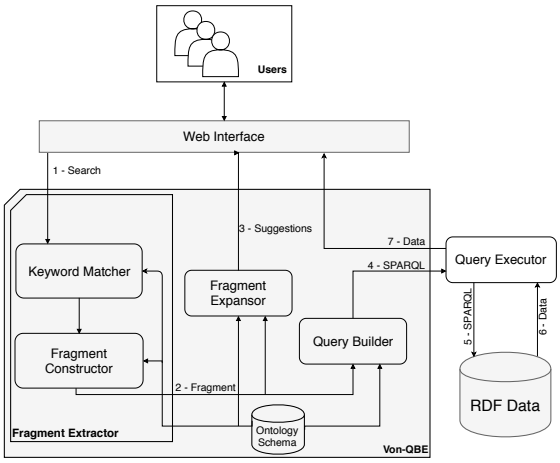

Fig. 1: Von-QBE's architecture

well-known graph algorithms from the literature. Von-QBE is implemented using Scala, Java (the web service) and ReactJS (the web interface). In what follows, we provide a brief idea of each Von-QBE main components.

Fragment Extraction is responsible for identifying, from a natural language question Q_N, the ontology subset (here we call as a *fragment*) that corresponds to the classes and properties mentioned on Q_N. The *Fragment Extraction* comprises two sub-components: 1) *Keyword Matcher* that identifies the ontology concepts mentioned on Q_N by using similarity metrics[4], and 2) *Fragment Constructor* that discovers how these concepts are related on the ontology schema by using well-known graph algorithms: Dijkstra shortest path and Prim Minimum spanning tree.

Fragment Expansor. Von-QBE provides examples to the user by expands Q_N using the ontology classes and properties that are directly connected to the fragment retrieved from the previous module. Considering our ontology O represented as an RDF graph, and the fragment nodes are ontology concepts, the *Fragment Expansor* expands the fragment with all edges (of course, the ones that are not already in the fragment) that come in (or out) from the fragment nodes.

Query Builder works as follows: each edge in the fragment (outputted by *Fragment Constructor* or expanded with the suggestions outputted by *Fragment Expansor* and accepted by the user) is added as a triple pattern in the query, and the source and the target nodes are named as variables. Since the schema might have properties with multiple domains and ranges, *Query Builder* also adds a clause to inform the instance type (class) for each variable. All the triple patterns are given as input to Apache Jena library[4] which generates Q_S according to the SPARQL syntax.

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

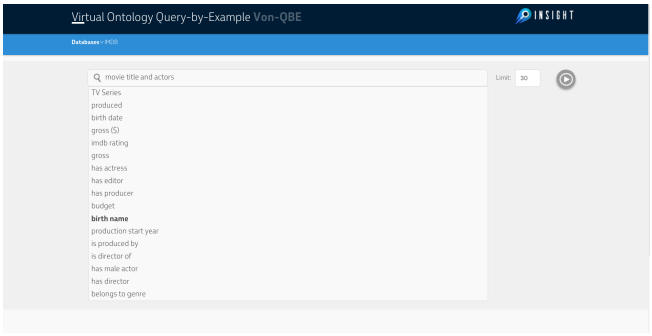

Fig. 2: Von-QBE suggests other concepts from the search query: *movie title and actors*

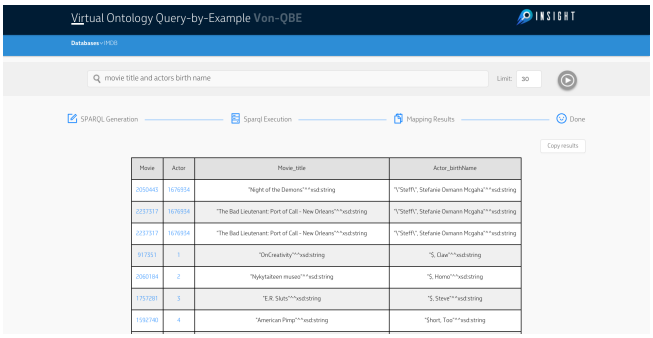

Fig. 3: Results for the text *movie title and actors birth name*

3 Demonstration Scenario: IMDB Movie Ontology

Von-QBE works with any RDF ontology that presents a schema and a SPARQL endpoint. The current version of Von-QBE supports Virtuoso SPARQL endpoint and Ontop[2] mapping to a non-RDF database. It is worth to mention that Von-QBE effectiveness depends on the RDF schema quality to work properly. In this section, we present some demonstration scenarios using the IMDB Movie Ontology[5].

Figure 2 shows Von-QBE's interface. First, the user should write the keyword search or the query at the text field. Once the user has written a term, Von-QBE suggests other concepts or relations from the ontology schema based on the user-written term(s). Figure 2 shows the suggestions given by Von-QBE to the query: *movie title and actors*. The property *birth name* (an attribute from the *Actor* class) appears as a suggestion. The user can choose it and keep constructing the search query.

Once the user finishes writing the search query, he/she can limit the number of triples returned based on a limit value and run the query over the database. Von-QBE will then retrieve the results and return them to the user, as in Figure 3. Von-QBE also provides the generated SPARQL from the user query search,

[5] https://sites.google.com/site/ontopiswc13/home/imdb-mo

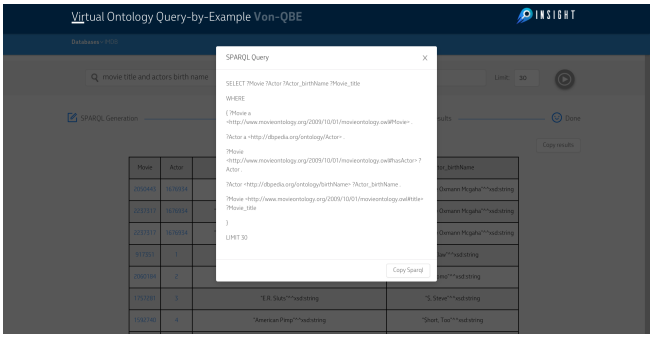

Fig. 4: SPARQL generated from the user-search query: *movie title and actors birth name*

as stated in Figure 4. The user has also the option to copy the results table to the clipboard as a .tsv(*tab-separated values*) and paste anywhere.

4 Conclusion

In this paper, we present Von-QBE to address the problem of querying over RDF databases using natural language question or a keyword search. Moreover, Von-QBE also helps the user to construct his/her query and translates the user-query search into SPARQL query. From the authors' knowledge, Von-QBE is the first tool to address such a problem using only the ontology schema. As future work, we aim at using natural language processing tools to detect entities described in the query and find its corresponding concept over the ontology schema.

References

1. Arnaout, H., and Elbassuoni, S. Effective searching of rdf knowledge graphs. *Journal of Web Semantics 48* (2018), 66 – 84.
2. Calvanese, D., Cogrel, B., Komla-Ebri, S., Kontchakov, R., Lanti, D., Rezk, M., Rodriguez-Muro, M., and Xiao, G. Ontop: Answering sparql queries over relational databases. *Semantic Web 8*, 3 (2017), 471–487.
3. Consortium, W. W. W., et al. Rdf 1.1 concepts and abstract syntax.
4. Gomaa, W. H., and Fahmy, A. A. A survey of text similarity approaches. *IJCA 68*, 13 (2013), 13–18.
5. Peres, L., Silva, T. L. C. d., Macedo, J., and Araujo, D. Ontology based query by example. ER 2019.
6. Usbeck, R., Ngomo, A.-C. N., Bühmann, L., and Unger, C. Hawk–hybrid question answering using linked data. In *European Semantic Web Conference* (2015), Springer, pp. 353–368.
7. Xu, K., Zhang, S., Feng, Y., and Zhao, D. Answering natural language questions via phrasal semantic parsing. In *Natural Language Processing and Chinese Computing*. Springer, 2014, pp. 333–344.
8. Yahya, M., Berberich, K., Elbassuoni, S., Ramanath, M., Tresp, V., and Weikum, G. Natural language questions for the web of data. In *Proceedings of the*

2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (2012), Association for Computational Linguistics, pp. 379–390.

9. Yih, S. W.-t., Chang, M.-W., He, X., and Gao, J. Semantic parsing via staged query graph generation: Question answering with knowledge base.