

INFERENCE BASED INTERPRETATION OF KEYWORD QUERIES FOR OWL ONTOLOGY

Noman Hasany and Batool Alwatban

Department of Computer Science, College of Computer, Qassim University, KSA

ABSTRACT

Most of the systems presented to date deals with RDF format so they are limited in actually addressing the knowledge base features from the ontology based on OWL semantics. Now, there is a need that actual OWL features i.e. rules and axioms must be addressed to give precise answers to the user queries. This paper presents an interface to OWL ontology which also considers axioms and restrictions that can result in inferring results in understanding user queries and in selecting appropriate SPARQL queries for getting better interpretation and answers.

KEYWORDS

Keyword query interfaces, OWL axioms, Ontology

1. INTRODUCTION

Natural language interfaces to structured and semi-structured information are not new in computer science literature. They are basically used to facilitate the users in getting to the information quickly and easily through human languages without dealing with how the information is stored. One of the problems faced in these type of interfaces is their regular upgradation, which is almost always needed when the structure on which they are based introduces some new features or if it is scaled to a new format. In this paper, we have presented a model that can address the new features of OWL in order to support OWL axioms and rules. Here, we have discussed a model to deal with OWL axioms and restrictions which can lead to more effective answers of user queries through implied inference support in the reasoner.

2. REVIEW OF NATURAL LANGUAGE INTERFACES TO ONTOLOGIES

Natural language interfaces to ontologies can be categorized based on the user query format which can be either syntactic [1] [2] or keyword based [3] [4] [5] [6]. Users using keyword based query input has an advantage that the query can be posed quickly ignoring the grammatical mistakes. Some interfaces help users in constructing the query [8].

In keyword based interfaces to ontologies, some interfaces are too restrict to accept noun and verbs only but some interfaces gives the flexibility to the user to write even a grammatical question. In the later type of keyword interfaces, some extra information is also extracted from the query that can help in determining various parameters that can help in disambiguate the user query in order to get the precise and effective answer to the user question.

The problem with the current systems is that they discuss the solutions with respect to RDF and not with respect to OWL axioms and restrictions. So, the conversion and ranking problems discussed with respect to the systems mentioned above do not use the full capability of OWL in

terms of disjointness, symmetry, transitivity and inverse relation [9] [10]. However, these are the features that present ontology as a real knowledge base. Our major concern in this paper is to incorporate OWL features that can lead to inferencing and hence better understanding of user query. Some research has been done on searching for keywords in the OWL ontology and the result is shown as a graph [11] but it does not presented the work with respect to OWL restrictions and axioms. Work has been presented for SPARQL query answering [12] [16] on OWL ontologies but not with respect to user queries mapping to SPARQL queries.

Some surveys has suggested inferencing as the future in semantic search systems [7]. OWL-QL did some work with inferencing but requires heavy processing. Most of the systems still solving the keyword search problem using graph, and the capability of OWL with its support for description logic is still missing [7].

3. OWL AXIOMS AND RESTRICTIONS

An ontology for the university domain is constructed using Protégé [13]. The university ontology contains five ontological resources i.e. classes, individuals, datatype property, object property and literals. For the inference and reasoner purposes, the ontology is defined containing various restrictions on individuals and properties to give more meaningful interpretation of queries. Expression builder in Protégé is used to put restrictions on ontology which are used by the reasoner while running SPARQL queries. These restrictions are associated to the resources using the Protégé expression builder as shown in Figure 1.

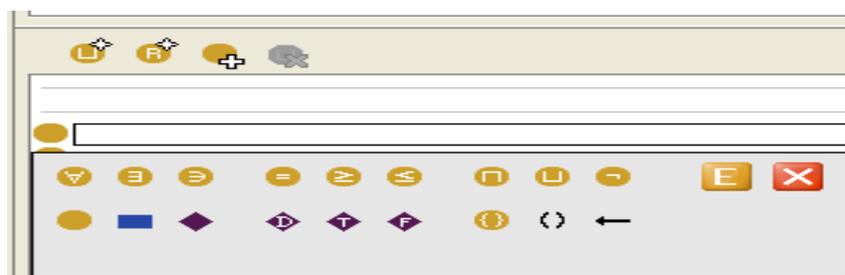


Figure 1: Protégé Expression Builder

The description of the symbols used in the expression builder are given in Table 1.

Table 1. Symbols used in the Expression Builder

Symbol	Meaning
	Insert all Values From
	Insert some Values From
	Insert has Values
	Insert min Cardinality
	Insert Cardinality
	Insert max Cardinality
	Insert intersection of
	Insert complement of

	Insert union of
	Show error message
	Cancel editing
	Insert class
	Insert property
	Insert individual
	Insert data type
	Insert true
	Insert false

For example, in order to create a restriction of `OWL:allValuesFrom` on ‘CSCtheses’ class that acts along with the property ‘`examinewith`’ a filler of ‘`thesis_examiner`’ by using create new expression.

1. Make sure that ‘CSCtheses’ is selected, click on the “Necessary” header in the ‘Conditions Widget’.
2. Press the ‘Create new expression’ button from the expression builder. This will display the ‘Create Restriction’ dialog as shown in Figure 1, which is used to create a restriction.
3. Insert ‘OWL: allValuesFrom’ as the type of restriction.
4. In the left of restriction, by using the Insert properties button to display the properties dialog, select examine as the property to be restricted.
5. In the right of restriction is for the filler of ‘thesis_examiner’.
6. Press the ‘OK’ button on the dialog to close the dialog and create the restriction. If there are any errors (due to typing errors etc.) the dialog will not close and an error message will be displayed at the bottom of the expression builder panel.

The code generated for `OWL:allValuesFrom`.

```
<Owl: Restriction>
<owl:onPropertyrdf: resource="# examine "/>
<OWL:AllValuesForm>
<Owl:Class>
<owl:Classrdf: about="# thesis_examiner "/>
</owl:Class>
</OWL:AllValuesForm>
</owl:Restriction>
```

This example describes an anonymous OWL class of all individuals for which examine property only has values from the class `thesis_examiner` shown in Figure 2.

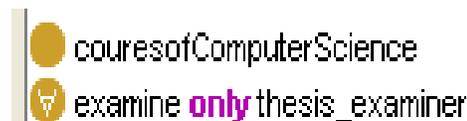


Figure 2: Restrictions related to CSCtheses

This means that if something is a member of the class CSCtheses, it is necessary for it to be a kind of coursesofComputerScience, and it is necessary for it to only have member from 'thesis_examiner' because of universal quantifier.

4. QUERY INTERPRETATION MODEL FOR OWL

In this section, we present our query interpretation model which can deal with various OWL features and restrictions, as shown in Figure 3. The idea of the research is based on formulating SPARQL queries for the user posed questions. Our approach was based on most of the design that first identify the keywords from the user questions [4] [6]. This first phase also involves the removal of noise words from the use query [5]. In our case, we have not eliminated the noise words but taken them into account for the identification of formal query templates. This approach remained to be successful as it helped in the identification of particular templates. For example, phrases like 'is it possible' or 'can be' with two classes at hand from the user query can be a target of query template with disjoint. If the two classes are defined with disjoint, it means there is no possibility for the said case. The identified keywords are then mapped to ontological resources i.e. the classes, individuals and property elements [1] [6]. Based on the ontological resources obtained, particular templates for the formal queries are chosen so the identified resources are mapped to the templates to form the final query which is ready for execution. The axioms in the query perform the inference implied in the SPARQL query execution. We have opted the approach presented in ONLI [15] for making some gold standard queries. Then our system tried to understand user queries and mapped them to possible templates. If one of the queries is mapped to the gold standard query previously determined then we are successful in getting our target. The obtained query is then executed to get the results which serve as the answers to the user questions [1] [5].

We discussed the creation of ontology in section 3 by considering the OWL axioms and restrictions defined with ontological resources. The model commences from accepting and understanding the user queries and ends up with the generation and execution of equivalent SPARQL queries in order to get the results. For understanding the user queries, keyword query processing techniques are implied even if the queries are in syntactic form.

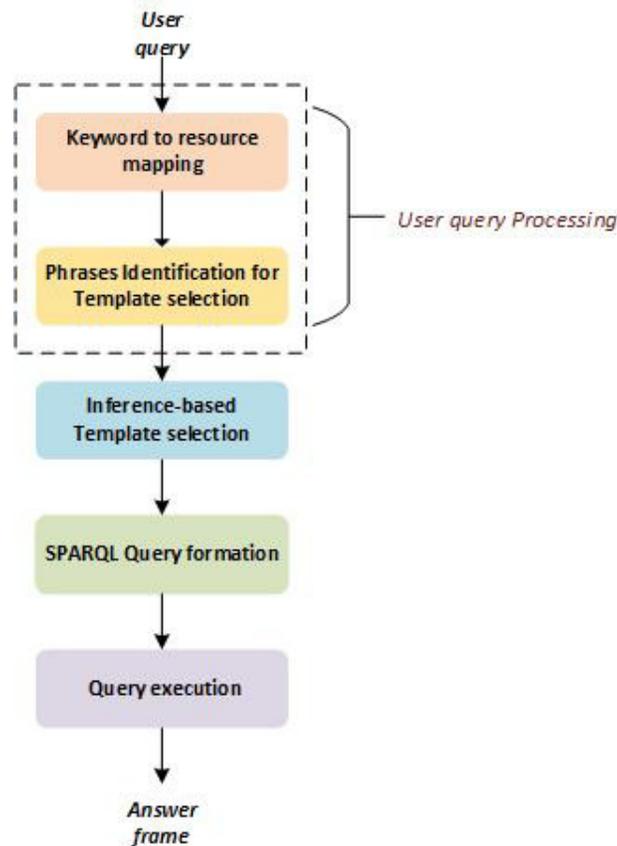


Figure 3: Query Interpretation Model

The query interpretation model consists of five phases as shown in Figure 3. The phases are discussed as follows:

4.1. User Query Processing

In the user query processing phase, we have suggested two major steps: the user query keywords to ontological element/resource mapping step and then identification of user words that can help in the selection of proper SPARQL template.

4.1.1. Keyword to Resource Mapping

The mapping step is responsible of providing a match of the user query terms to the ontological resource names used in the ontology. This can involve tools like stemmer or morphological analyser to find out the root or the base word for the user query words. In this step, mainly four type of ontological resources are identified i.e. the classes, individuals/instances, object properties and datatype properties.

4.1.2. Phrases Identification for Template selection

The user query is analysed in this step for noisy words like a, the etc. to be removed from the user query. But there is a major difference in our model as compare to other keyword interfaces. Some keyword interfaces just retrieve the nouns and verbs and sometime adjectives but our keyword

interface do not delete words like ‘can be’ or ‘only’ etc. as they may be helpful in formulating the appropriate OWL restriction for the user query.

4.2. Inference-based Template Selection

A process OWL feature phase is the one in which the OWL features and restrictions are selected based on the user query features extracted that address them. It is important in our model as this is the point of difference in our proposed model from the earlier ones. A list of SPARQL templates are provided in the discussion of algorithm.

In order to serve an example for a template, we consider the phrase ‘can be’ taken from the user query in the phase identification phrase. If the resources are identified as classes, the template chosen has the best target for *:disjointWith*.

By using disjoint we can find the disjoint members between classes. It provides an easy way to find the possibility for a class to share properties from another class [14].

These queries can be solved by the following SPARQL template:

```
SELECT ?var1 ? var2
WHERE {
?var1 OWL:disjointWith ?var2.
filter(?var1 = :class1Name) .
filter(?var2 = :class2Name)}
```

For the example SPARQL template construction, consider the user query: “Is Assistant Professor can be a thesis examiner?”

In the keyword to resource mapping phase, Assistant professor and thesis examiner are mapped to *:CSCAssistantProf* and *:thesis_examiner* ontological class resources.

Finding the disjoint property between the classes can easily provide the answer for the phrase ‘can be’, in the presence of class resources extracted from the ontology. After the substitution of resources in the template, following SPARQL query is obtained.

```
SELECT ?p ?o WHERE { ?p owl:disjointWith ?o.
filter(?o =:thesis_examiner).
filter(?p = :CSCAssistantProf) }
```

Results	
p	o
CSCAssistantProf	thesis_examiner

Figure 4: Result of query with disjoint

Due to the implied inference mechanism, it can be observed from the result in Figure 4 that the two classes *CSCAssistantProf* and *thesis_examiner* are disjoint to one another. The non-emptied result of the query identify that for this particular university, an assistant professor cannot be a thesis supervisor. In order to get this inference work here, it is important that the ontology must be defined by taking care of OWL axioms and restrictions at the ontology construction time.

4.3. SPARQL Query Formation

In this phase, the output of the last two stages is group together to form the final SPARQL query which is ready to run to get the final results for the user. The responsibility of this phase is to determine the variable for the SELECT clause, triple pattern and filter for the WHERE clause.

We have used SPARQL as a formal query for our model as it is standardized by W3C. A SPARQL Basic graph patterns (BGP) is a set of triple patterns written as conjunction of triple patterns, mainly consists of two clauses: the SELECT clause and the WHERE clause. The conversion algorithm working on a pattern decides what is variable query for the SELECT clause and what is the WHERE clause. There is set of primitives for the WHERE clause of the SPARQL that can be chosen based on the resources found and their connectivity information:

Relation between classes (?i1? p1 ?i2)

Relation to represent variable as individual of class (?i1 rdf:type ?c1)

Relation to represent property (?s :isSupervisorOf ?y) means ?s is a supervisor of ?y.

Where ?s is a domain of ?y and ?y is range for supervisors.

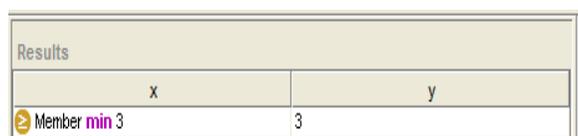
Relation to restrict the class individual to a specific individual by using filter. For example, filter (?x1 = :Batool_Rasheed)

The restriction phrases are defined by the expression builder panel and implemented by placing a filter expression as restriction on individual or on a data type value.

An example filter is (?salary>7000).

4.4. Query Execution

The query obtained is then compared with the manually constructed gold standard queries. The SPARQL query obtained from query formation phase is then executed using the Protégé platform to get the final results, as shown in the Figure 5.



Results	
x	y
Member min 3	3

Figure 5: Result from SPARQL execution

5. CONCLUSIONS

RDF interfaces model for querying ontologies provides assistance for users to restore the required information from ontologies based on ontological resources. These type of interfaces have a drawback as they handle limited with knowledge base logic features which are supported in OWL in the form of features and restrictions. To overcome problem this work presents Keyword query Model for OWL based ontologies addressing OWL features and restrictions in addition to the basic ontological resources. The proposed model suggests extraction of keywords and key phrases that can be mapped to the proposed templates. The proposed templates are constructed by considering the OWL features and restrictions but are not limited to deal with the OWL features and also incorporate the necessary RDF features. To answer the user query in the model, formal templates of target queries like SPARQL are proposed. Templates provide efficient and effective conversion for user queries but at the same time can have a limited coverage of input queries. Between the query understanding and template construction phase, the process is guided with the help of flowchart that can extract phrases or words from the user query and find out the appropriate SPARQL query template suitable to answer the user query.

6. FUTURE WORK

The proposed model discussed here is based on one-to-one mapping of user query terms to ontological resources. The interface can be extended to automatically interconnecting the results of subsequent queries in getting answer for user queries. The effect of different semantic measures like incorporating Word Net etc. can also be discussed. The query conversion mechanism is based on predefined templates. As a future work construction of graph from the identified resources can also be studied. This mechanism will be useful for complex queries in which resources are far located and joined with the help of the graph. The model presented is limited to some OWL axioms and restrictions chosen from the expression builder. It can be extended to all supported axioms and restrictions whereas it can be scaled to OWL 2 features. We describe the query OWL model up to the axiom level support. To extend the model to the rule level is the future work. This involves study of query decomposition to answer the information embodied in the form of SWRL rules.

ACKNOWLEDGEMENTS

We would like to thank Scientific Research Deanship (SRD) of Qassim University, for the grant number 2185 under which we completed this project.

REFERENCES

- [1] Kaufmann, Esther, Abraham Bernstein, and Renato Zumstein. "Querix: A natural language interface to query ontologies based on clarification dialogs." In 5th International Semantic Web Conference (ISWC 2006), pp. 980-981. 2006.
- [2] Wang, Chong, Miao Xiong, Qi Zhou, and Yong Yu. "Panto: A portable natural language interface to ontologies." In European Semantic Web Conference, pp. 473-487. Springer Berlin Heidelberg, 2007.
- [3] Kaufmann, Esther, Abraham Bernstein, and Lorenz Fischer. "Nlp-reduce: A "naive" but domain-independent natural language interface for querying ontologies." 4th ESWC, Innsbruck, A (2007).
- [4] Tran, Thanh, Haofen Wang, Sebastian Rudolph, and Philipp Cimiano. "Top-k exploration of query candidates for efficient keyword search on graph-shaped (rdf) data." In 2009 IEEE 25th International Conference on Data Engineering, pp. 405-416. IEEE, 2009.
- [5] Hasany, Noman, A. B. Jantan, M. H. B. Selamat, and Mohd Iqbal Saripan. "Querying ontology using keywords and quantitative restriction phrases." *Inform. Technol. J* 9 (2010): 67-78.
- [6] Zhou, Qi, Chong Wang, Miao Xiong, Haofen Wang, and Yong Yu. "SPARK: adapting keyword query to semantic search." In *The Semantic Web*, pp. 694-707. Springer Berlin Heidelberg, 2007.
- [7] Mäkelä, Eetu. "Survey of semantic search research." In *Proceedings of the seminar on knowledge management on the semantic web*. Department of Computer Science, University of Helsinki, Helsinki, 2005.
- [8] Zenz, Gideon, Xuan Zhou, Enrico Minack, Wolf Siberski, and Wolfgang Nejdl. "Interactive Query Construction for Keyword Search on the Semantic Web." In *Semantic Search over the Web*, pp. 109-130. Springer Berlin Heidelberg, 2012.
- [9] Karanastasi, Anastasia, Alexandros Zotos, and Stavros Christodoulakis. "The OntoNL framework for natural language interface generation and a domain-specific application." In *Digital Libraries: Research and Development*, pp. 228-237. Springer Berlin Heidelberg, 2007.
- [10] KOSSEIM, Leila, Reda SIBLINI, Christopher JO BAKER, and Sabine BERGLER. "Using Selectional Restrictions to Query an OWL Ontology."
- [11] Kim, S. I., & Kim, H. S. (2013, January). Ontology modeling for provision of semantic based open API information. In *Advanced Communication Technology (ICACT), 2013 15th International Conference on* (pp. 664-667). IEEE.
- [12] Kollia, I., Glimm, B., & Horrocks, I. (2011, May). SPARQL query answering over OWL ontologies. In *Extended Semantic Web Conference* (pp. 382-396). Springer Berlin Heidelberg.
- [13] Protégé: <http://protege.stanford.edu/>
- [14] Protégé OWL tutorial.
http://mowl-power.cs.man.ac.uk/protegeowltutorial/resources/ProtegeOWLTutorialP3_v1_0.pdf

- [15] Mithun, S., Kosseim, L., & Haarslev, V. (2007, October). Resolving quantifier and number restriction to question owl ontologies. In *Semantics, Knowledge and Grid, Third International Conference on* (pp. 218-223). IEEE.
- [16] Schneider, M. (2010, June). SPARQLAS–Implementing SPARQL Queries with OWL Syntax. In *Proceedings of the 3rd Workshop on Transforming and Weaving Ontologies in Model Driven Engineering*. CEUR Workshop Proceedings (Vol. 604).

Authors

Dr. Noman Hasany is working as an Assistant Professor in College of Computer, Qassim University, KSA. He did PhD in Computer Systems Engineering from Universiti Putra Malaysia; and M.E. and B.E. in Computer Systems Engineering from NED University, Karachi. His research interests include Semantic Web, Natural Language Processing, Information retrieval and Knowledge based systems.



Ms. Batool Alwatban has completed her Masters degree in Computer Science from College of Computer, Qassim University, KSA.