



IN3067/INM713 SEMANTIC WEB TECHNOLOGIES AND
KNOWLEDGE GRAPHS
COURSEWORK PROJECT (PART2)

Authors: Ali Momenzadeh Kholejani & Pawel Staszynski

Contents

Introduction	3
2.2 Tabular Data to Knowledge Graph (Task RDF).....	3
Subtask RDF.0.....	3
Subtask RDF.3.....	3
2.3 SPARQL and Reasoning (Task SPARQL)	3
Subtask SPARQL.1.....	4
Subtask SPARQL.2.....	4
Subtask SPARQL.3.....	5
Subtask SPARQL.4.....	5
Subtask SPARQL.5.....	6
2.4 Ontology Alignment (Task OA)	7
Subtask OA.2	7
Subtask OA.4	7
2.5 Ontology Embeddings (Task Vector)	7
Summary	7
References.....	9

Introduction

We have developed the source code for this entire coursework in a single notebook file called "Coursework_Solution.ipynb," which can be found in the "code/KG_Coursework" directory of the compressed file. Additionally, in the "readme.txt" file within the "code" directory, there are two links: one for accessing the notebook on Google Colab and the other for the entire directory on Google Drive, which operates in conjunction with the notebook code. It is worth mentioning that the structure of the Google Drive folders differs from that of the folders in the compressed file, as the latter is designed based on the coursework description. In the upcoming sections, we will only describe some of the necessary details, as other details regarding the code are already presented in the notebook; we do not repeat them.

2.2 Tabular Data to Knowledge Graph (Task RDF)

Subtask RDF.0

In this subtask we created the restaurants "Restaurant Ali 1", "Restaurant Ali 2", "Restaurant Pawel" in "cw_data", by adding 5 data entries to the end of the file.

Subtask RDF.3

Regrettably, we were unable to fully implement the Google KG and Wikidata KG services. However, we did comprehend the overall structure of their APIs. We have included our code as comments, as we were unable to integrate it into our coursework due to time constraints.

2.3 SPARQL and Reasoning (Task SPARQL)

The queries for this section can be located in either the "code/KG_Coursework/queries/" or "sparql" directories. Furthermore, the resulting CSV files are situated in either the "code/KG_Coursework/output/queries_output" or "sparql" directories.

Subtask SPARQL.1

This query returns the restaurants names and the pizzas they serve, only for the restaurants that their name starts with "the".

```
PREFIX cw: <http://www.semanticweb.org/city/in3067-inm713/2024/restaurants#>
```

```
SELECT ?restaurantName ?pizza
WHERE {
  ?pizza rdf:type cw:Pizza .
  ?pizza cw:servedIn ?restaurant .
  ?restaurant cw:restaurantName ?restaurantName .
  FILTER regex(?restaurantName, "^the")
}
```

Subtask SPARQL.2

This query returns the restaurants names, pizzas, and their ingredients only for the restaurants with names shorter than 10 characters and pizzas with ingredients either "Olives" or "Cheese".

```
PREFIX cw: <http://www.semanticweb.org/city/in3067-inm713/2024/restaurants#>
```

```
SELECT ?restaurantName ?pizza ?ingredient
WHERE {
  ?pizza rdf:type cw:Pizza .
  ?pizza cw:servedIn ?restaurant .
  ?restaurant cw:restaurantName ?restaurantName .
  ?pizza cw:hasIngredient ?ingredient .
  FILTER (STRLEN(?restaurantName) < 10
    && (?ingredient = cw:Olives || ?ingredient = cw:Cheese))
}
```

Subtask SPARQL.3

This query returns restaurantName, pizzas, and their ingredient for pizzas containing either "Fig" or "Pineapple", together with pizzas containing "Olives".

PREFIX cw: <http://www.semanticweb.org/city/in3067-inm713/2024/restaurants#>

```
SELECT ?restaurantName ?pizza ?ingredient
WHERE {
  {
    ?pizza rdf:type cw:Pizza .
    ?pizza cw:servedIn ?restaurant .
    ?restaurant cw:restaurantName ?restaurantName .
    ?pizza cw:hasIngredient ?ingredient .
    FILTER (?ingredient = cw:Fig || ?ingredient = cw:Pineapple)
  }
  UNION
  {
    ?pizza rdf:type cw:Pizza .
    ?pizza cw:servedIn ?restaurant .
    ?restaurant cw:restaurantName ?restaurantName .
    ?pizza cw:hasIngredient ?ingredient .
    FILTER (?ingredient = cw:Olives)
  }
}
```

Subtask SPARQL.4

This query returns the count of different pizzas served in each restaurant and filters out the restaurants that serve more than "8" types of pizza.

PREFIX cw: <http://www.semanticweb.org/city/in3067-inm713/2024/restaurants#>

```

SELECT ?restaurantName (COUNT(?pizza) as ?numPizzas)
WHERE {
    ?pizza rdf:type cw:Pizza .
    ?pizza cw:servedIn ?restaurant .
    ?restaurant cw:restaurantName ?restaurantName .
}
GROUP BY ?restaurantName
HAVING (COUNT(?pizza) > 8)

```

Subtask SPARQL.5

This query returns restaurant names alongside counts of distinct pizzas and ingredients served by each of them. It then filters restaurants to get the ones which sells more than 5 pizzas and orders the results by pizza count in descending order and by ingredient count in ascending order.

PREFIX cw: <<http://www.semanticweb.org/city/in3067-inm713/2024/restaurants#>>

```

SELECT ?restaurantName
    (COUNT(DISTINCT ?pizza) AS ?numPizzas)
    (COUNT(DISTINCT ?ingredient) AS ?numIngredients)
WHERE {
    ?pizza rdf:type cw:Pizza .
    ?pizza cw:servedIn ?restaurant .
    ?restaurant cw:restaurantName ?restaurantName .
    ?pizza cw:hasIngredient ?ingredient .
}
GROUP BY ?restaurantName
HAVING (COUNT(DISTINCT ?pizza) > 5)
ORDER BY DESC(?numPizzas) ASC(?numIngredients)

```

2.4 Ontology Alignment (Task OA)

Subtask OA.2

In this task, which involved calculating the Precision and Recall, we encountered the necessity to modify the initial function provided in lab sessions and Wikipedia. This modification was essential to map the triples of (subject, predicate, object) and (object, predicate, subject) as our generated file and the given files differed in the order of subjects and objects. We faced the option of either rewriting our file to match the order of the given file or adapting the function to accommodate these variations. We chose the latter option as it proved to be more feasible for meeting our requirements.

Subtask OA.4

In this task we have created the query below that returns the pizza name, ingredients, and type of pizza for pizzas with the "VegetarianPizza" type.

PREFIX pizza: <http://www.co-ode.org/ontologies/pizza/pizza.owl#>

```
SELECT ?pizza ?ingredient ?type
WHERE {
    ?pizza a pizza:Pizza ;
        a ?type ;
        pizza:hasIngredient ?ingredient .
    FILTER (?type = pizza:VegetarianPizza)
}
```

2.5 Ontology Embeddings (Task Vector)

Due to the heavy workload of the current semester, we were unable to proceed with this task, as we unfortunately ran out of time.

Summary

In this coursework, we had the opportunity to delve into the internals of ontologies, gaining a comprehensive understanding of the underlying mechanics of ttl and owl formats. Additionally, we explored the intricacies of reasoning and querying ontologies in a deeper manner, which facilitated a clearer comprehension of their operational dynamics and the types of tasks for which they are most

adeptly suited. Furthermore, we delved into the inner workings of Protégé software, elucidating its capabilities in reasoning and detecting inconsistencies, while also recognizing the importance of a well-designed ontology in mitigating such issues.

References

1. Lab Solutions by Ernesto Jiménez-Ruiz
2. https://en.wikipedia.org/wiki/Precision_and_recall