Concordia University

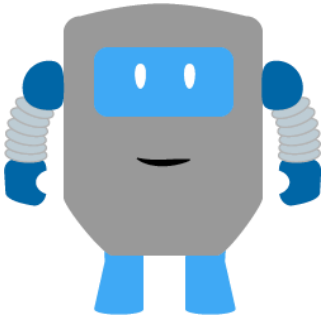
Engineering and Computer Science

COMP 6741

Intelligent Systems

(Project-1 Report)

Unibot



Submitted To: Prof. Dr. René Witte

Date: 21 March 2021

Team: FL_G_7

Vickykumar Patel (40185238)

Kavleen Kour Sidhu (40205476)

1. Abstract

Our project aims to build a system called- *"Unibot",* an agent that can intelligently answer queries of student studying in Concordia University using knowledge graphs and RDF techniques.

Students often have a lot of questions about courses, lectures, course materials and so on. To get answers to these questions, users often must jump from one webpage to another or try contacting their peers who have completed the same course previously. All this proves to be quite cumbersome for the user and eventually wasting their lot of time. This project proposes a new "technique" referred as Unibot, which acts as an intelligent agent to solve all the university-related questions using a knowledge graph and natural language processing.[8]

In the first part of the project, we are going to focus on building the knowledge graph and creating the database for courses, lectures, and their content. Therefore, we are going to explore the W3C technologies, particularly RDF and RDFS to represent our information in form of graphs. Thereafter, a series of competency questions will test the system against few SPARQL queries.[8]

2. Knowledge base

2.1 Pre-defined Vocabulary - For our project, we reused some of the public and state-of-art vocabularies, including:

Vocabulary	Association
RDF (Resource Description Framework)	It allows us to make statements about resources
FOAF (Friend-of-a- Friend)	An IRI to state an acquaintance relationship b/w people
OWL (Web Ontology Language)	These are designed for automated reasoning
Dublin Core	Set of predefined URIs representing different properties of a given document.

2.2 Developed Vocabulary- Besides using state of art vocabulary, we have defined a new vocabulary for stating relationship between university and student records.

2.2.1 Class Modelling

This section provides a description of the different terms used in building Student and University schema and for building the data components.

University: Class-University is defined as a Class and extends the Agent class from Dublin Core vocabulary. The University is the primary agent with the following schema:

```
# University
uni:University
   a rdfs:Class ;
   rdfs:subClassOf dcterms:Agent ;
   rdfs:label "University"@en ;
   rdfs:comment "University information" .
```

Course: Class- Course is defined as a Class and extends the Agent class from Dublin Core. The schema is as follows:

```
# Course
uni:Course
   a rdfs:Class ;
   rdfs:subClassOf dcterms:Agent ;
   rdfs:label "Course"@en ;
   rdfs:comment "Courses offered at University"@en .
```

Dublin core terms	**Association**
dcterms:title	Identifies the course name
dcmitype:subject	Identifies subject e.g. COMP, SOEN
dcmitype:identifier	Identifies the course number, e.g 6741
dcterms:description	Provides a description of the course
rdfs:seeAlso	Links the webpage with the course information
dcterms:isPartOf	Identifies the University which course belongs to ?? not clear
uni:hasContent	Identifies the course outline

Lecture: Class

Lecture belongs to course and extends the Event class from Dublin Core. The schema is as follows:

```
#  Lectures
uni:Lecture
   a rdfs:Class ;
   rdfs:subClassOf dcmitype:Event ;
   rdfs:label "Lecture"@en ;
   rdfs:comment "Information about lecture"@en .
```

Dublin core terms	Association
dcmitype:identifier	Identifies the lecture number
dcmitype:title	Identifies names of lecture
rdfs:seeAlso	Links the webpage with the lecture information
dcterms:isPartOf	Identifies the course to which lecture belongs.
uni:hasContent	Outlines the reading and reference materials for slides worksheets

2.2.2 Property Modelling

Topic: Property

A topic is a sub property of subject, which identifies a link to DBpedia:

```
uni:Topic
   a rdf:property ;
   rdfs:subPropertyOf dcterms:subject ;
   rdfs:label "Topic"@en ;
   rdfs:comment "URI"@en .
```

hasContent: Property

Identifies the contents that are associated with a course, lecture or event:

uni:hasContent

a rdf:property ;

rdfs:subClassOf dcterms:relation ;

rdfs:label "Content"@en ;

rdfs:range dcterms:BibliographicResource;

rdfs:comment "Content associated with a course, lecture"@en .

dcterms:type	Identifies the type of content, e.g., Readings, Slides, Worksheets.
dcterms:source	Locates the resource.

2.2.3 Student Modelling

Student: Class-Student is itself a Class and subclass of foaf:Person class from FOAF vocabulary.

#StudentInformation

uni:Student

ardf:Class;

rdfs:subClassOffoaf:Person;

rdfs:label"Student"@en;

rdfs:Comment "Student admitted to a University"@en.

studiesAt: Property-This property links the University to students by the relationship studiesAt.

```
uni:studiesAt
  a rdf:property;
  rdfs:subClassOf dcterms:relation ;
  rdfs:label "study at"@en ;
  rdfs:domain uni:Student ;
  rdfs:range uni:University;
  rdfs:comment "Relationship showing students enrolled at a university."@en.
```

enrolledIn: Property -Now, this shows the relationship between the students and the courses they are associated with.

```
uni:enrolledIn
  a rdf:property;
  rdfs:subClassOf dcterms:relation ;
  rdfs:label "Enrolled in"@en ;
  rdfs:domain uni:Student ;
  rdfs:range uni:Course;
  rdfs:comment "Relationship showing students enrolled in a Course."@en.
```

hasGot: Property – For each course, students receive grades. This property links the grades of students with their enrolled courses.

```
uni:hasGot
  a rdf:property;
  rdfs:subClassOf dcterms:relation ;
  rdfs:label "Got"@en ;
  rdfs:domain uni:Student ;
  rdfs:range uni:Course;
  rdfs:comment "Relationship showing students got grade in a Course."@en.
```

hasCredit: Property-Each course whether graduate/ Undergraduate is targeted with some grades. For Graduate courses its 4 credits, while for undergraduate its 3.

```
uni:hasCredit
  a rdf:property;
  rdfs:subClassOf dcterms:relation ;
  rdfs:label "Credit"@en ;
  rdfs:domain uni:Course ;
  rdfs:range xsd:String;
  rdfs:comment "Relationship showing course has credit."@en.
```

3. Data Creation

The database for building knowledge base was extracted from the https://opendata.concordia.ca/datasets/. The following files were used to create the .csv files we require for the project with the given attributes:

- **CU_SR_OPEN_DATA_CATALOG-45672834.csv** (To get Course ID, Subject, Credits, Title)
- **CU_SR_OPEN_DATA_CATALOG-45649197.csv** (To get Course ID and its respective description)
- **CATALOG.csv and Experiential Learning.csv** (To get Course Website/seeAlso also at DBpedia)
- The information contained in these csv files required some pre-processing before they could be used as Concordia knowledge base. Merging two csv files to get Course ID, Subject, Catalog, Long Title, Class Units, Description in a single file named: 'course_with_Description.csv'.

```
import pandas as pd

# reading two csv files
data1 = pd.read_csv('CU_SR_OPEN_DATA_CATALOG-45672834.csv', encoding='unicode_escape')
data2 = pd.read_csv('CU_SR_OPEN_DATA_CATALOG_DESC-45649197.csv', encoding='unicode_escape')

# using merge function by setting how='inner'
output1 = pd.merge(data1, data2,
                   on='Course ID',
                   how='inner')

# displaying result
output1.to_csv('course_with_Description.csv', index=False)
```

- Merging catalogue csv file and above formed 'course_with_Description.csv' to link the webpage from DBpedia to get the Outline

```
import pandas as pd

# reading two csv files
data1 = pd.read_csv('course_data.csv', encoding='unicode_escape')
data2 = pd.read_csv('course_with_Description.csv', encoding='unicode_escape')

# using merge function by setting how='inner'
output1 = pd.merge(data1, data2,
                   on='Course ID',
                   how='inner')

# displaying result
output1.to_csv('course_with_seeAlso.csv', index=False)
```

- The following files are generated and used for creating Knowledge Graph:

  course_data.csv: contains all the courses extracted from opendata. concordia.

  course_with_Description.csv: contains all the courses and their corresponding descriptions.

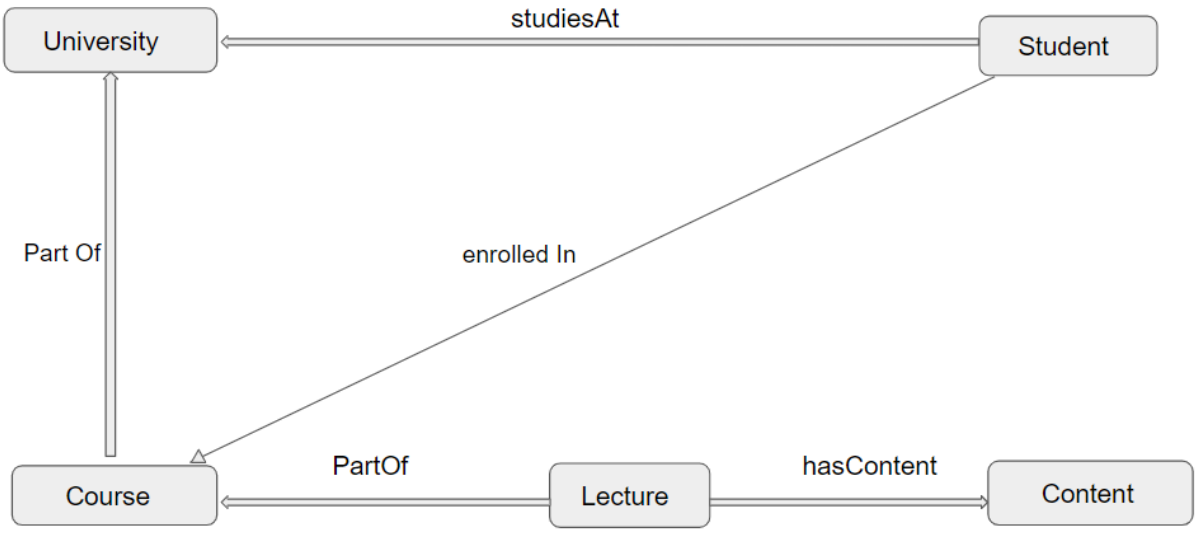
  course_with_seeAlso.csv: contains all the courses, their ID, and their reference link to graduate/undergraduate calendars.

  studentData.csv: Contains fake user data with Grades and Competencies.

4. Knowledge base

Knowledge Graphs captures entities, their attributes, and relationships. A standard triple can be represented as rdf = r: (s, p,o) → <rdf: subject, rdf: predicate, rdf: object>. The building of the knowledge base uses python programming language and the following libraries:

1. **Pandas:** Pandas is a very powerful framework to assist several computations like its ability to read and write various CSV files. We manipulated the CSV files in pandas using functions like read_csv().
2. **UUID:** A universally unique identifier (UUID) is used to identify each resource uniquely and make them distinct from each other.
3. **RDFLib Graph** helps to represent the knowledge information in form of graphs where nodes are URI references.
4. **Namespace:** helps to create a shorter version of URI by splitting them into prefix and suffix surrounded by < >. E.g.: RDFS, RDF DC, DCTERMS are provided as namespaces in rdflib.
5. **Literal** – These represent object values and can be connected with data types such as- xsd:string, xsd:number. xsd: date. Datatypes are used with rdf literals to represent values such as strings, numbers and dates.
6. The following diagram give a general flow of the code structure:



5. Queries

The following queries have been used to test the graph. The queries can be found in the accompanying Queries.txt file. In the dissertation we will search up to 10 queries and their results are as shown below. Further queries can be generated using the given turtle schema and vocabulary.

Question 1: What is the name [subject] and credit units of subject COMP6741?

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>


#1)what is the name and credit unit of subject COMP6741?

SELECT ?name ?credit
WHERE {
    ?course dcmitype:subject "COMP" .
    ?course dcmitype:identifier "6741" .
    ?course dcterms:title ?name .
    ?course uni:hasCredit ?credit.
}
```

QUERY RESULTS

Table | Raw Response

Showing 1 to 1 of 1 entries

Search: Show 50 entries

	name	credit
1	"INTELLIGENT SYSTEMS"@en	"4.0"

Showing 1 to 1 of 1 entries

Question 2: Which course and topics is Robert (ref. StudentData.csv) competent in?

```
PREFIX un: <http://www.w3.org/2007/ont/unit#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>


#2)Which course and topics is Robert competent in?
SELECT ?title ?competancy
WHERE {
  ?firstname foaf:givenName "Robert".
  ?firstname uni:enrolledIn ?cid.
  ?cid dcterms:title ?title.
  ?firstname dcterms:description ?competancy.
}
```

QUERY RESULTS

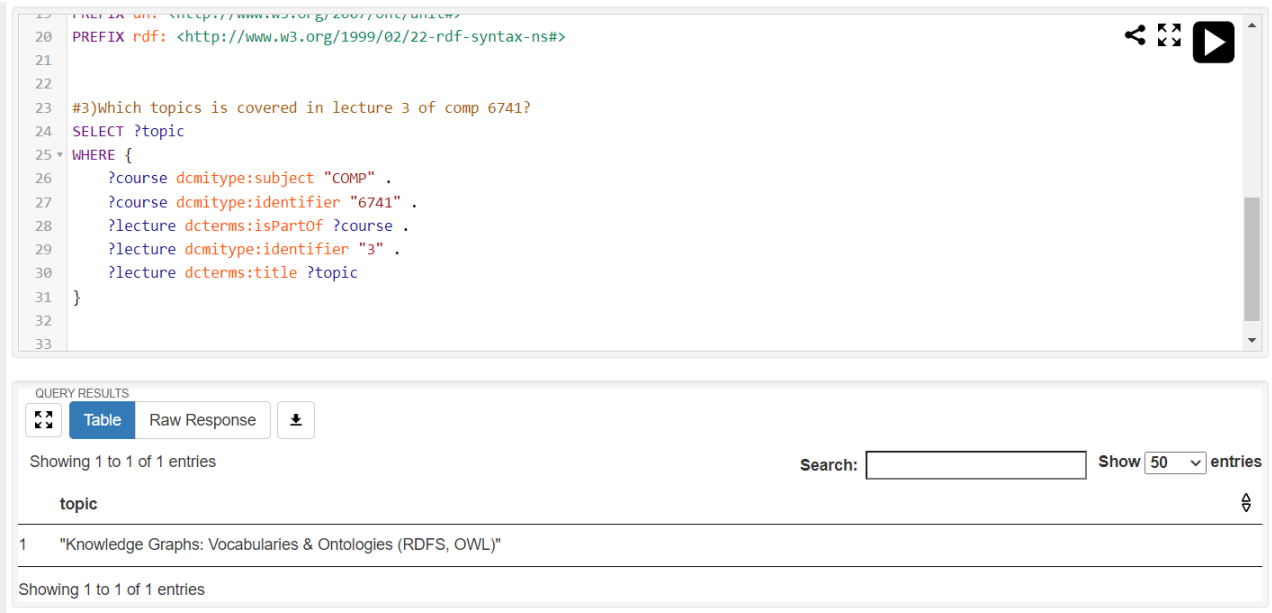
Table | Raw Response

Showing 1 to 1 of 1 entries

Search: Show 50 entries

	title	competancy
1	"ALGORITHM DESIGN TECHNIQUES"@en	"Mathematical preliminaries; Empirical and theoretical measures of algorithm efficiencies;greedy algorithms, dynamic programming, and graph algorithms; Amortized analysis; String matching algorithms and tries; Unidimensional and multidimensional search trees; NP-complete problems and approximate solutions; "

Question 3: Which topics is covered in lecture 3 of [subject][number] COMP 6741?



```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>


#3)Which topics is covered in lecture 3 of comp 6741?
SELECT ?topic
WHERE {
    ?course dcmitype:subject "COMP" .
    ?course dcmitype:identifier "6741" .
    ?lecture dcterms:isPartOf ?course .
    ?lecture dcmitype:identifier "3" .
    ?lecture dcterms:title ?topic
}
```

QUERY RESULTS

Table | Raw Response

Showing 1 to 1 of 1 entries

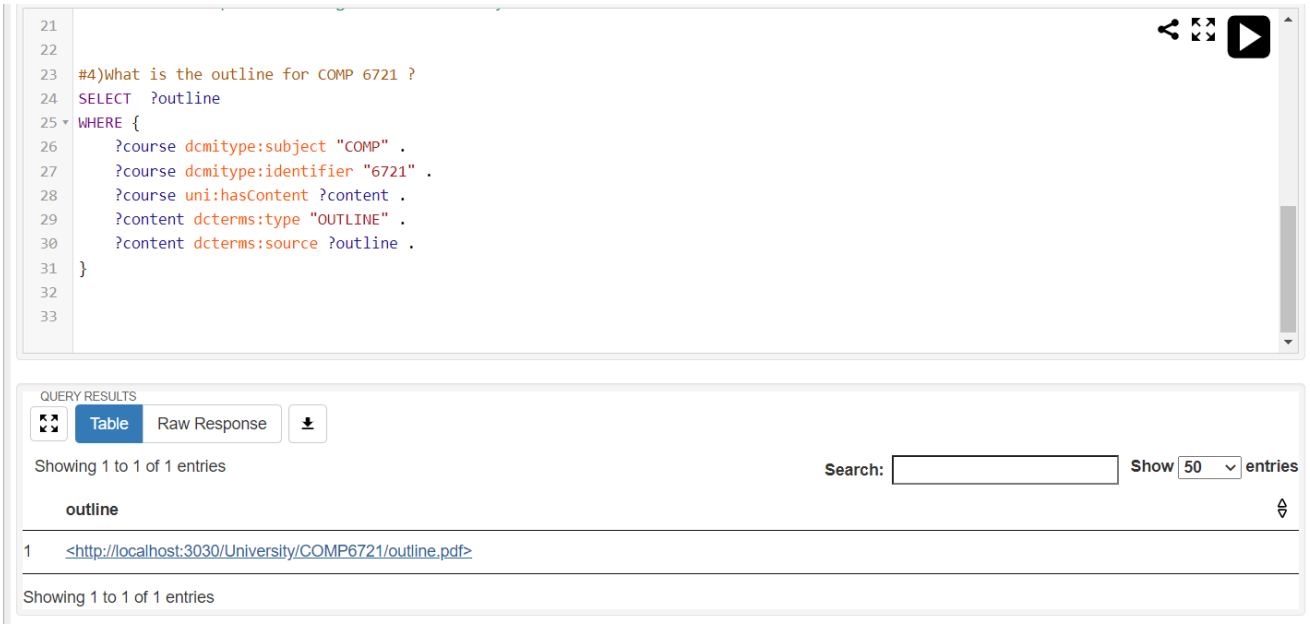
Search: Show 50 entries

	topic
1	"Knowledge Graphs: Vocabularies & Ontologies (RDFS, OWL)"

Showing 1 to 1 of 1 entries

Question 4: What is the course outline for COMP 6721?

The link further opens- http://localhost:3030/University/COMP6721/outline.pdf as the pdf file.



```


#4)What is the outline for COMP 6721 ?
SELECT  ?outline
WHERE {
    ?course dcmitype:subject "COMP" .
    ?course dcmitype:identifier "6721" .
    ?course uni:hasContent ?content .
    ?content dcterms:type "OUTLINE" .
    ?content dcterms:source ?outline .
}
```

QUERY RESULTS

Table | Raw Response

Showing 1 to 1 of 1 entries

Search: Show 50 entries

	outline
1	<http://localhost:3030/University/COMP6721/outline.pdf>

Showing 1 to 1 of 1 entries

Question 5: What are the recommended reading materials for INTELLIGENT SYSTEMS [subject][name]?

```
#5)What are the recommended reading materials for INTELLIGENT SYSTEMS?
SELECT  ?Reading
WHERE {
    ?courseid dcterms:title "INTELLIGENT SYSTEMS"@en.
    ?courseid rdf:type uni:Course .
    ?lectureid dcterms:isPartOf ?courseid.
    ?lectureid uni:hasContent ?content.
    ?content dcterms:type "READING".
    ?content dcterms:source ?Reading.
}
```

QUERY RESULTS

Table | Raw Response

Showing 1 to 10 of 10 entries

Search: Show 50 entries

	Reading
1	<https://en.wikipedia.org/wiki/ELIZA>
2	<https://en.wikipedia.org/wiki/Watson_(computer)>
3	<https://plato.stanford.edu/entries/artificial-intelligence/>
4	<https://concordiauniversity.on.worldcat.org/oclc/897466408>
5	<https://concordiauniversity.on.worldcat.org/oclc/960211579>
6	<https://concordiauniversity.on.worldcat.org/oclc/897466408>
7	<https://concordiauniversity.on.worldcat.org/oclc/897466408>
8	<https://concordiauniversity.on.worldcat.org/oclc/897466408>
9	<http://informationretrieval.org/>
10	<https://concordiauniversity.on.worldcat.org/oclc/314121652>

Showing 1 to 10 of 10 entries

Question 6: Which students enrolled in COMP 6741 and what grade they got?

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
#6) Which students  enrolled in COMP 6741 and what grade they got?

SELECT ?firstname ?lastName ?grade
WHERE{

    ?course dcmitype:subject "COMP" .
    ?course dcmitype:identifier "6741".
    ?student uni:studyAt uni:Concordia_University.
    ?student uni:enrolledIn ?course.
    ?student foaf:givenName ?firstname.
    ?student foaf:familyName ?lastName.
    ?student uni:hasGot ?grade.
}
```

QUERY RESULTS

Table | Raw Response

Showing 1 to 2 of 2 entries

Search: Show 50 entries

	firstname	lastName	grade
1	"Joe"	"Mathew"	"A+"
2	"James"	"Cook"	"F"

Showing 1 to 2 of 2 entries

Question 7: How many courses are offered in COMP department?

```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX un: <http://www.w3.org/2007/ont/unit#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

#7) How many course offered in COMP Department?
SELECT (count(?course) as ?courseCount)
WHERE{

    ?course dcmitype:subject "COMP" .
    ?course rdf:type uni:Course.

}
```

QUERY RESULTS

Table | Raw Response

Showing 1 to 1 of 1 entries

Search: Show 50 entries

	courseCount
1	"108"^^xsd:integer

Showing 1 to 1 of 1 entries

#Question 8: Is James (ref. StudentData.csv) competent in INTELLIGENT SYSTEMS?

Checking student's competencies - It is defined as a set of topics, based on the courses a student successfully passed (e.g., if a student passes COMP474, a course that includes the topic "Knowledge Graphs")

```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX un: <http://www.w3.org/2007/ont/unit#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

#8) Is james competent in INTELLIGENT SYSTEMS?

ASK{
    ?course dcterms:title "INTELLIGENT SYSTEMS"@en.
    ?student uni:enrolledIn ?course.
    ?student foaf:givenName "James".
    ?student uni:hasGot ?grade.
    FILTER (?grade != "F")
}
```

QUERY RESULTS

Table | Raw Response

✗ False

Question 9: List down the topics common for subjects COMP 6741 and COMP 6721?

```
WHERE
{
    ?course1 dcmitype:subject "COMP" .
    ?course1 dcmitype:identifier "6741" .
    ?lecture1 dcterms:isPartOf ?course1 .
    ?lecture1  uni:Topic ?topic1 .

    ?course2 dcmitype:subject "COMP" .
    ?course2 dcmitype:identifier "6721" .
    ?lecture2 dcterms:isPartOf ?course2 .
    ?lecture2  uni:Topic ?topic2 .
    FILTER(?topic1 = ?topic2 ) .
}
```

QUERY RESULTS

Table | Raw Response

Showing 1 to 6 of 6 entries

Search: Show 50 entries

	topic1	topic2
1	dbr:Artificial_neural_network	dbr:Artificial_neural_network
2	dbr:Knowledge_Graph	dbr:Knowledge_Graph
3	dbr:Natural_language_processing	dbr:Natural_language_processing
4	dbr:Deep_learning	dbr:Deep_learning
5	dbr:Machine_learning	dbr:Machine_learning
6	dbr:Intelligent_agent	dbr:Intelligent_agent

Showing 1 to 6 of 6 entries

Question 10: Which course has the maximum number of credits?

```
PREFIX uni: <http://uni.io/schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX un: <http://www.w3.org/2007/ont/unit#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

#10) Which are the maximum credits (aggregate sum)?

SELECT (MAX(?credit) as ?courseCredit)
WHERE {
  ?course rdf:type uni:Course.
  ?course uni:hasCredit ?credit
}
```

QUERY RESULTS

Table | Raw Response

Showing 1 to 1 of 1 entries

Search: Show 50 entries

	courseCredit
1	"999.0"

6. Statistics

Statistics	Count
Knowledge_base	51072
University_Schema.ttl	49
Number of lectures	23

7. Running program

1) Ensure RDFLib, and pandas are installed in python environment.
2) Place the university folder containing subject content and reference readings on the ApacheFuseki server →webapp folder.
3) Run the knowledgebuilder.py
4) Run the ApacheFuseki server and create a new dataset test and upload Knowledge_base (i.e. the n triples format).
5) From the Queries.txt file, copy with PREFIX section, copy the query that should be executed.

8. References

[1] RDF Schema 1.1
https://www.w3.org/TR/rdf-schema/

[2] FOAF Vocabulary
http://xmlns.com/foaf/spec/

[3] Dublin Core Metadata Initiative
https://www.dublincore.org/specifications/dublin-core/dcmi-terms/#http://purl.org/dc/dcmitype/Event

[4] Vivo Core Ontology
https://lov.linkeddata.es/dataset/lov/vocabs/vivo

[5] Vivo Tutorial by Shanshan Chen, Yuyin Sun, Ying Ding
https://info.sice.indiana.edu/~dingying/Teaching/S604/VIVO-tutorial-v1.pdf

[6] Merge 2 CSV files.
https://www.geeksforgeeks.org/how-to-merge-two-csv-files-by-specific-column-using-pandas-in-python/

[7] Write SPARQL query in Fuseki-Server
https://www.youtube.com/watch?v=5-UfFV5XmTI