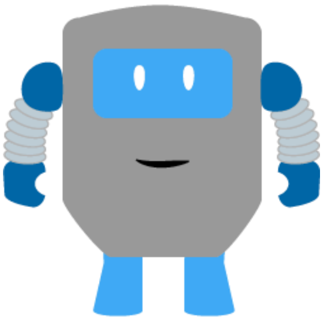
COMP 6741

Intelligent Systems

(Project Report)

Unibot



Submitted To: Prof. Dr. René Witte

Date: 21 March 2021

Team: FL_G_7

Vickykumar Patel (40185238)

Kavleen Kour Sidhu (40205476)

Part I

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.
```

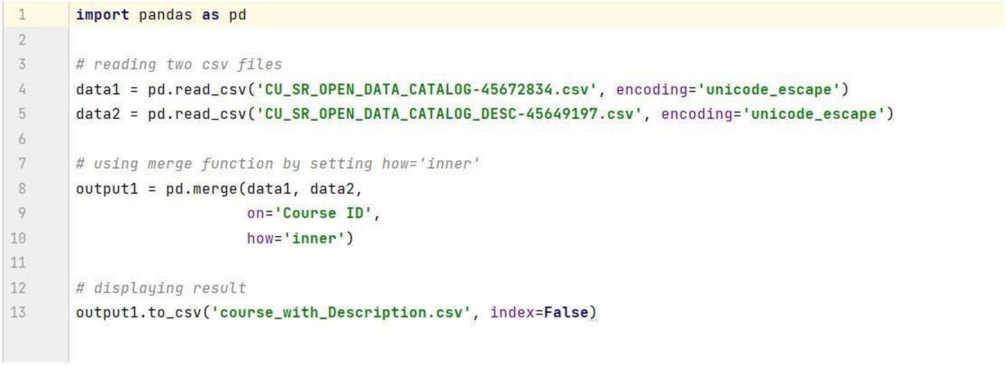
hasCredit: Property-Each course whether graduate/ Undergraduate is targeted with some grades. For Graduate courses its 4 credits, while for undergraduate its 3.

```
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

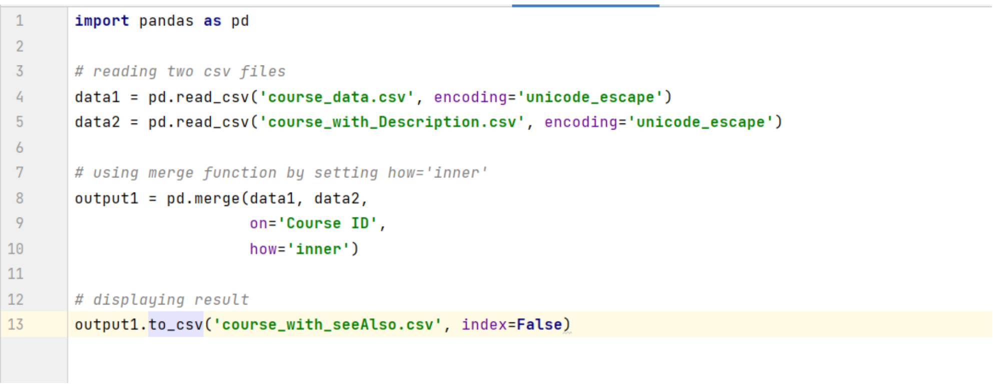
The database for building knowledge base was extracted from thehttps://opendata.concordia.ca/datasets/. The following files were used to create the .csv files we require for the project with the given attributes:

- **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'.



```python
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



```python
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.

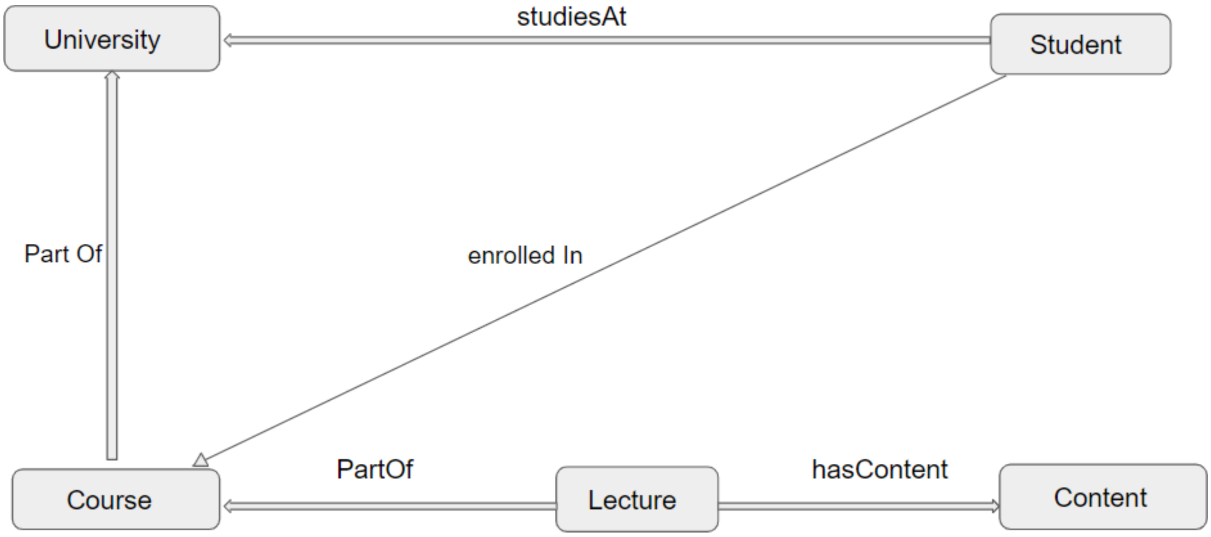
  course_with_seeAlso.csv: contains all the courses, their ID, and their reference link to graduate/undergraduate calendars.

  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:

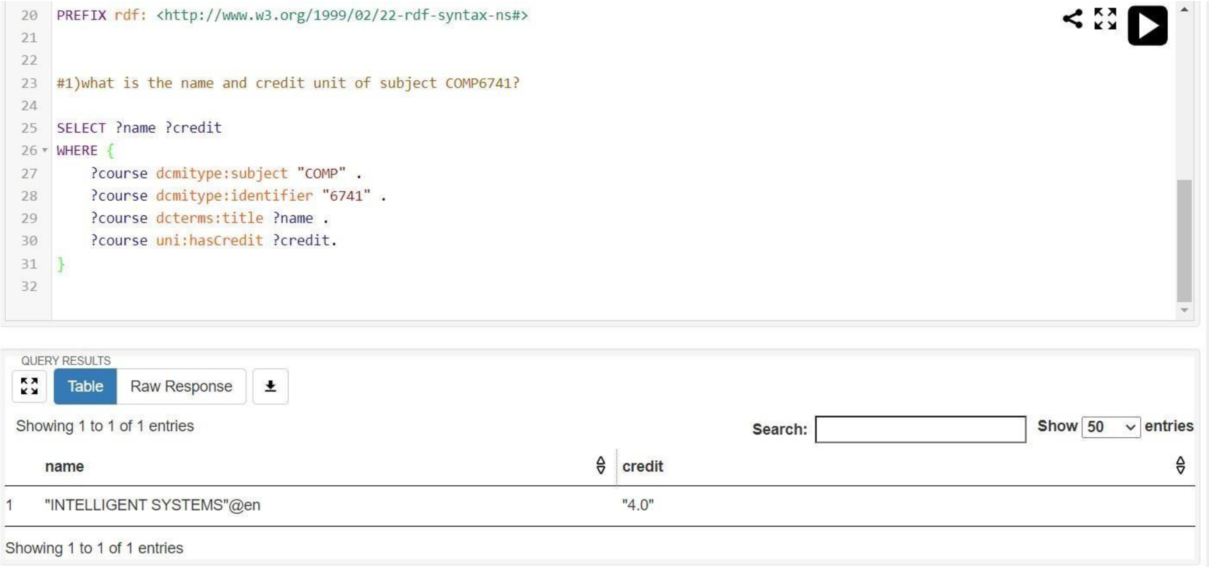
1. **Pandas:** Pandas is a very powerful framework to assist several computations like its ability to read and write various CSV files. We manipulated the CSV files in pandas using functions like read_csv().
2. **UUID:** A universally unique identifier (UUID) is used to identify each resource uniquely 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?



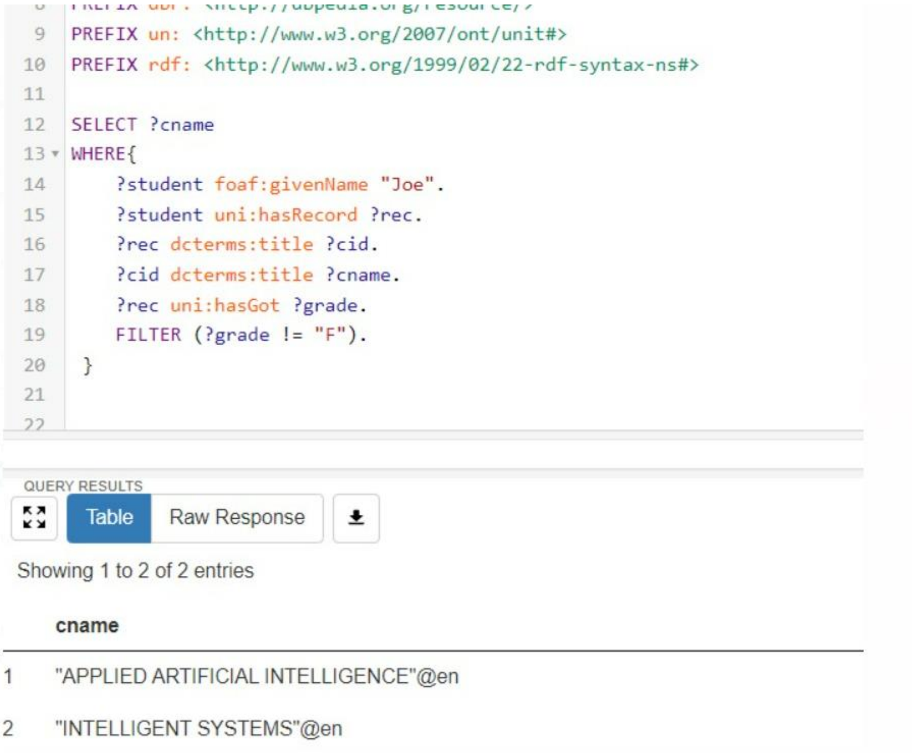
```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>


#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.
}
```

Showing 1 to 1 of 1 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?

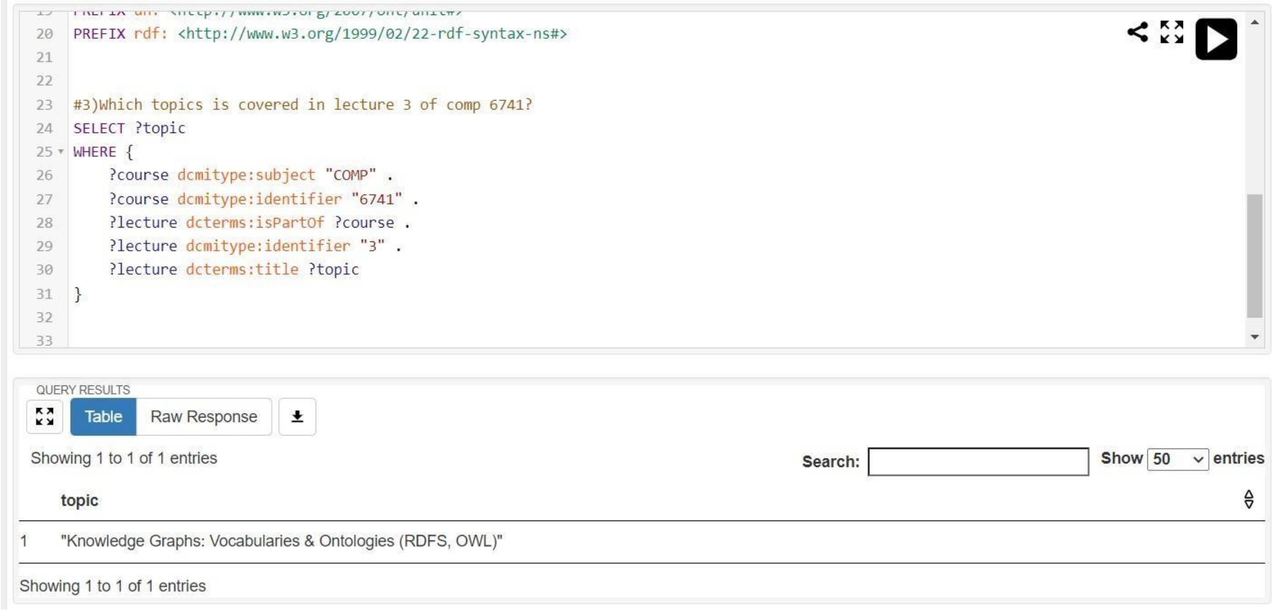


```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX un: <http://www.w3.org/2007/ont/unit#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?cname
WHERE{
    ?student foaf:givenName "Joe".
    ?student uni:hasRecord ?rec.
    ?rec dcterms:title ?cid.
    ?cid dcterms:title ?cname.
    ?rec uni:hasGot ?grade.
    FILTER (?grade != "F").
 }
```

Showing 1 to 2 of 2 entries

	cname
1	"APPLIED ARTIFICIAL INTELLIGENCE"@en
2	"INTELLIGENT SYSTEMS"@en

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



```
PREFIX uni: <http://www.w3.org/2007/ont/unit#>
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
}
```

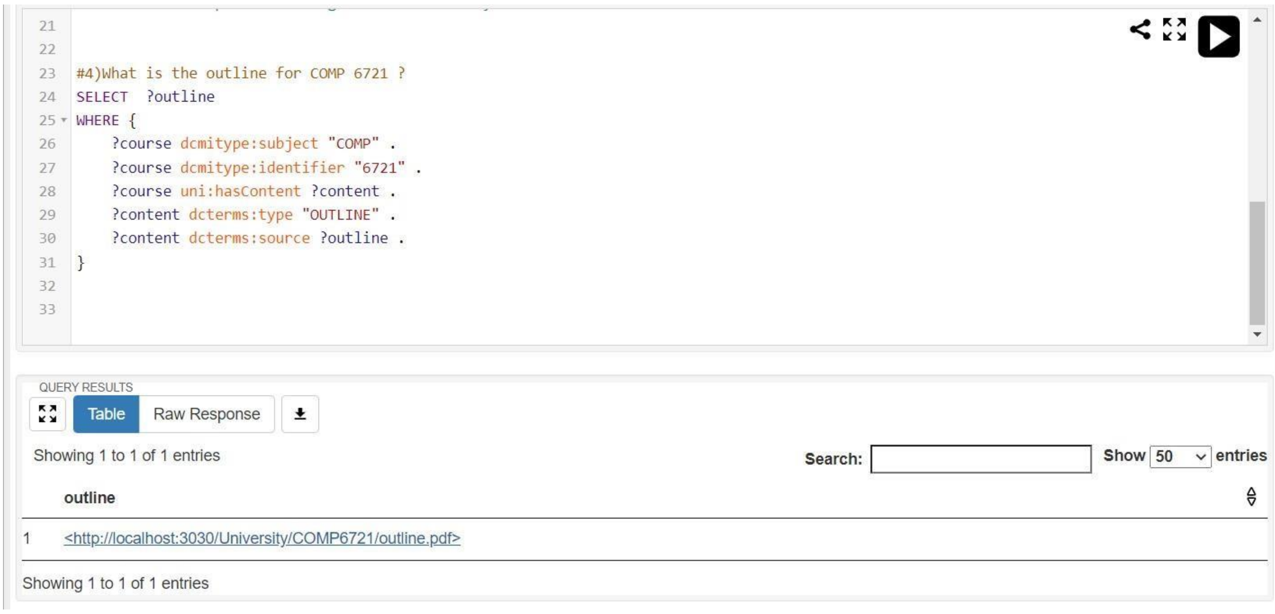
Showing 1 to 1 of 1 entries

	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.



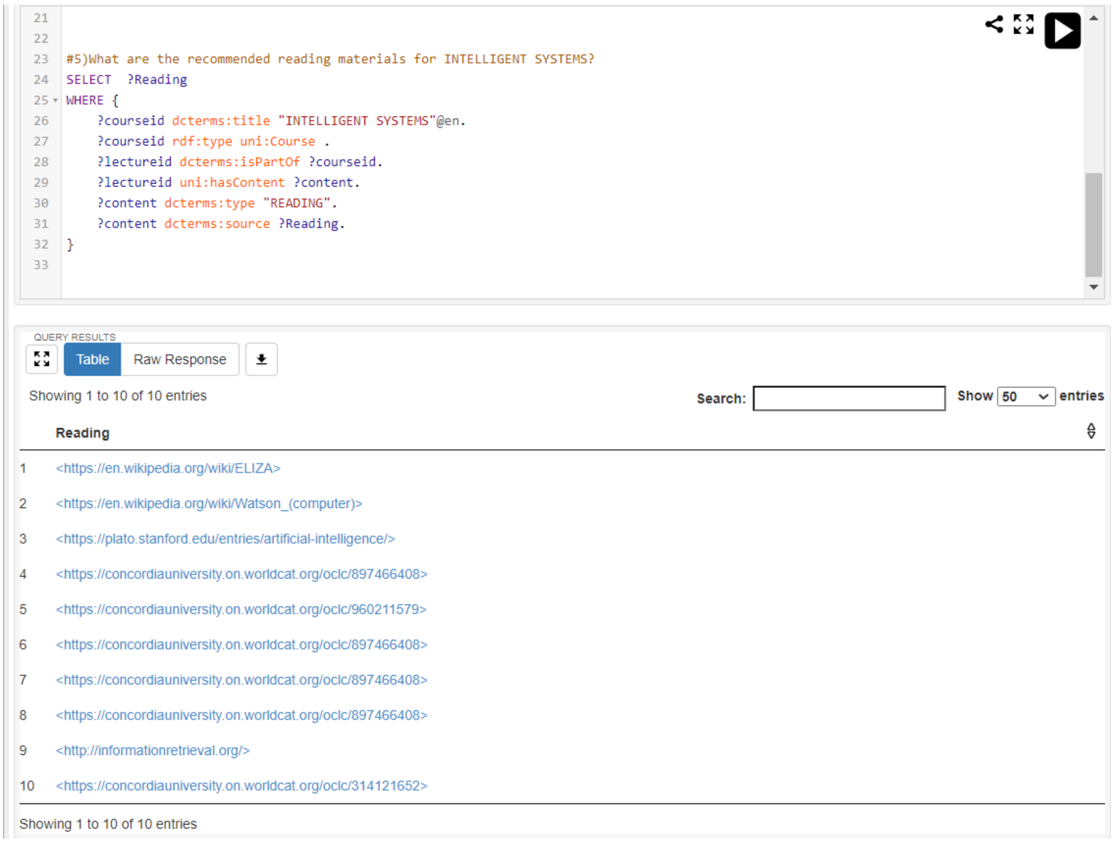
```


#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 .
}
```

Showing 1 to 1 of 1 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]?



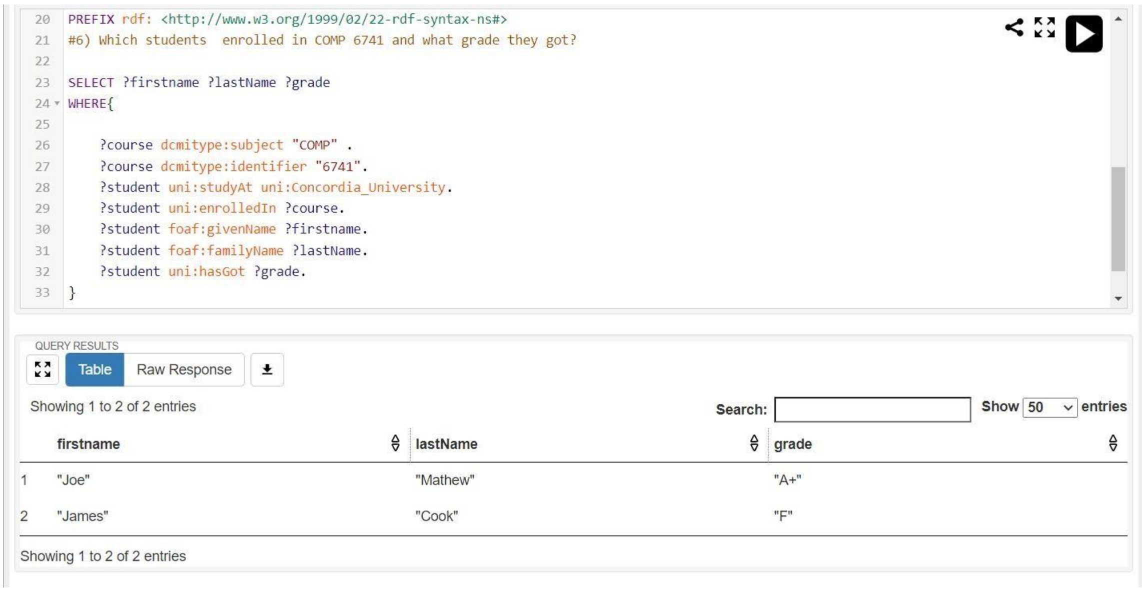
```


#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.
}

```

Showing 1 to 10 of 10 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?



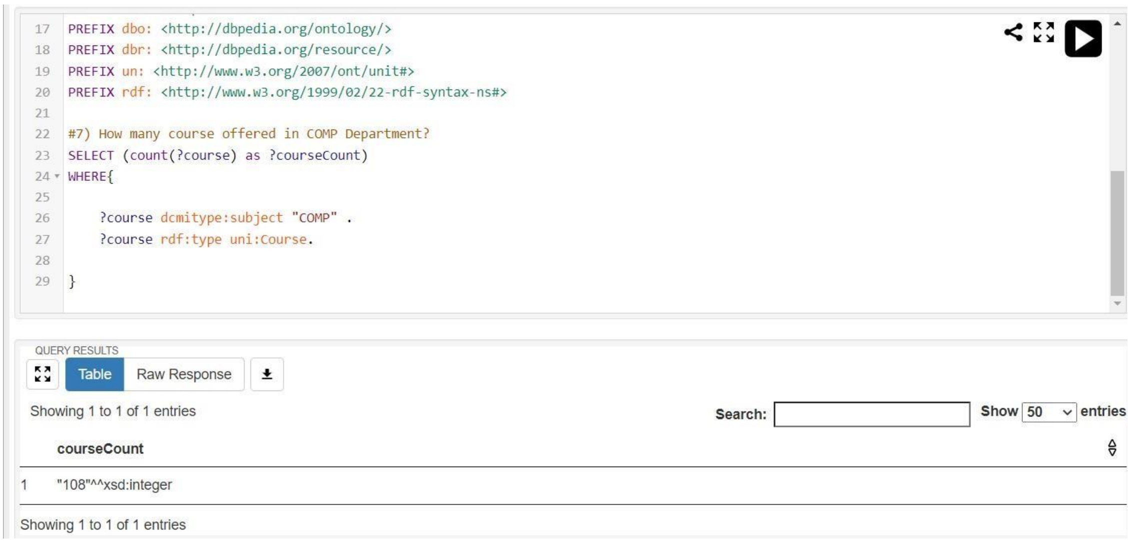
```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
#6) Which students  enrolled in COMP 6741 and what grade they got?

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.
}
```

Showing 1 to 2 of 2 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.

}
```

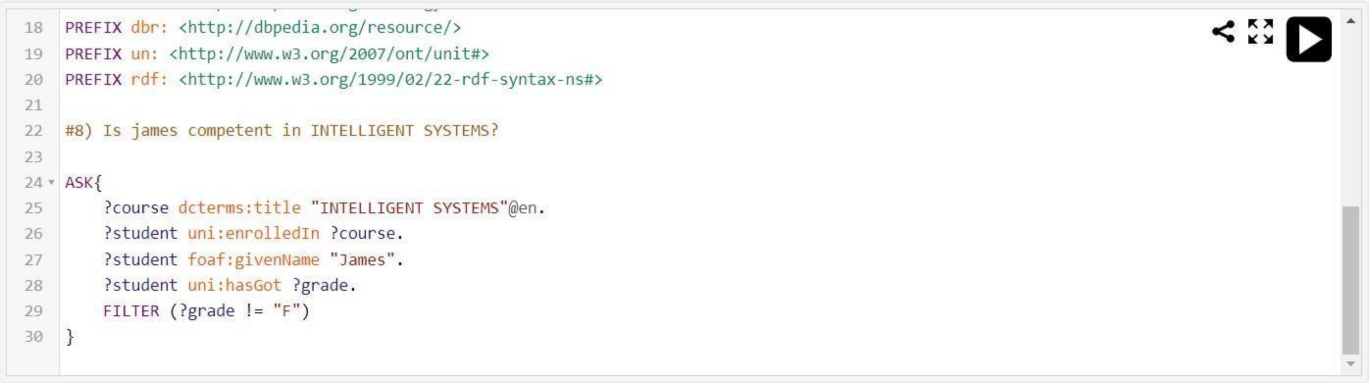
Showing 1 to 1 of 1 entries

	courseCount
1	"108"^^xsd:integer

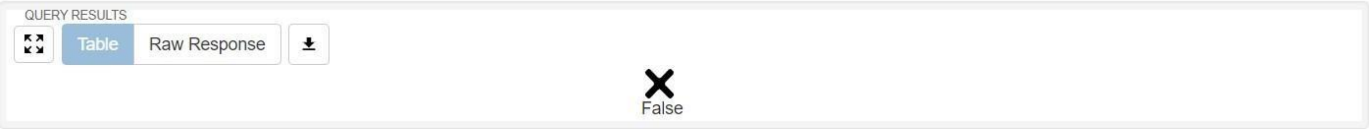
Showing 1 to 1 of 1 entries

#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”)

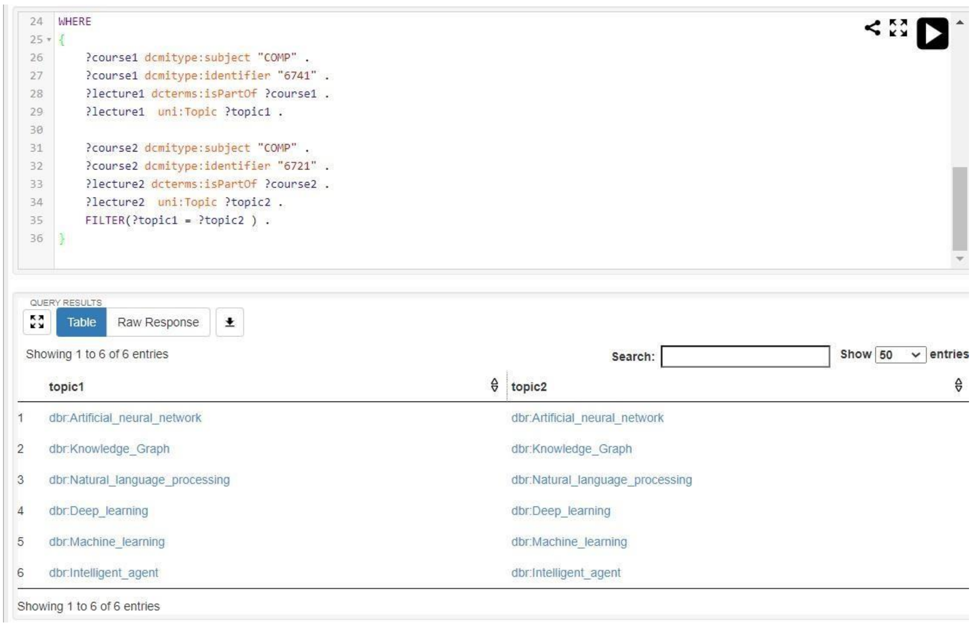


```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX un: <http://www.w3.org/2007/ont/unit#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

#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")
}
```



✗
False

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



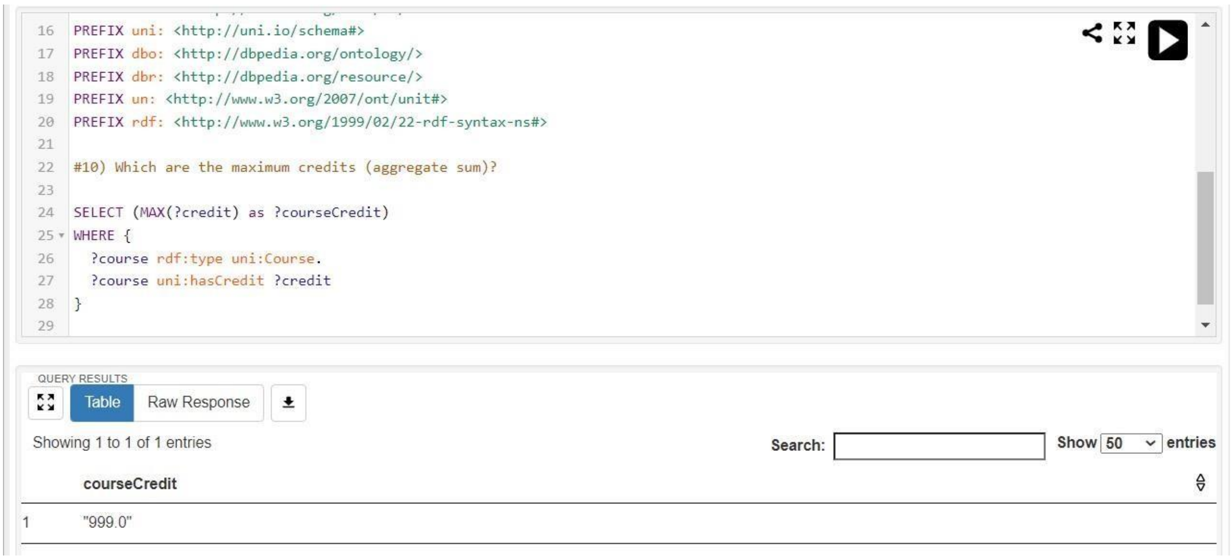
```
WHERE
{
    ?course1 dcmitype:subject "COMP" .
    ?course1 dcmitype:identifier "6741" .
    ?lecture1 dcterms:isPartOf ?course1 .
    ?lecture1  uni:Topic ?topic1 .

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

Showing 1 to 6 of 6 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
}

```

Showing 1 to 1 of 1 entries

	courseCredit
1	"999.0"

6. Statistics

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

Part II

7. Data Description

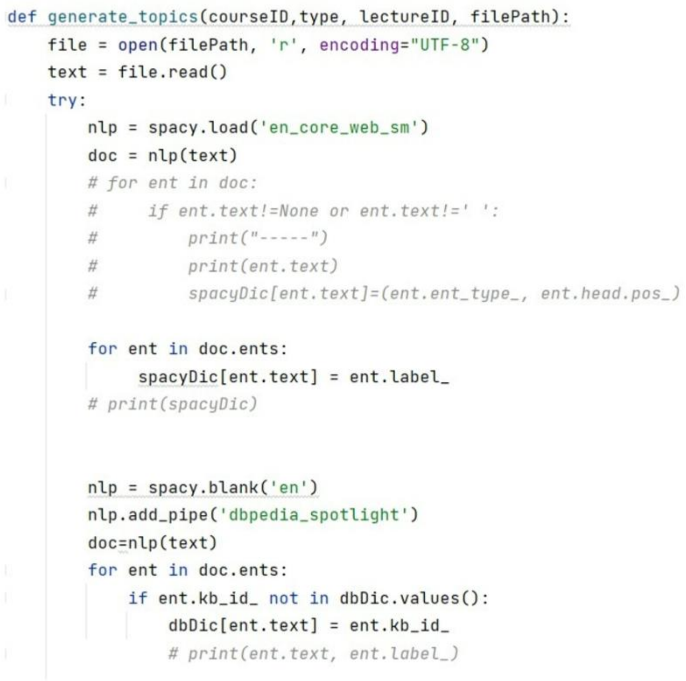
The following files are generated and used for creating Knowledge Graph:

- **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
- **Topics.csv:** Contains name each topics with its corresponding DBpedia link extracted using Tika.
- **Term.csv**: Contains data as to in which course and what grade was achieved on completion of respective course.
- **TopicGeneration.csv:** Contains the DBpedia entities with link filtered through POS technique, so as to keep only the meaningful items and discard those that do not carry semantic value.

8.Topic Generation

To establish topics for two courses we follow the steps as below:

- Accumulate all the lecture materials for the 2 courses including Lecture slides, worksheets, lab pdfs and others and set them into 2 separate folders COMP 6721(Applied Artificial Intelligence) and COMP 6741 (Intelligent Systems).
- Convert the collected materials in distinguished plain text files using Apache Tika. It is being implemented in ‘pdftotxt.py’. The duplicate materials as well as duplicate lines from the text file for each lab, lecture slides and worksheets were removed. This helps to avoid the extra requests made onto DBpedia spotlight for similar lines.
- Now, we give these plain text files as input to Dbpedia Spotlights and Spacy to extract the Dbpedia links and entity labels. This process is performed using TopicGeneration.py file.
- These are now saved in csv files. Lab Materials: {lab_topics.csv, labContent.csv}, Lecture Materials : {lectureData.csv, lectureContent.csv}. TopicGeneration.py file generates the Topics.csv file which contains all the Dbpedia link and entity label for extracted topics of the course.
- Next, knowledge builder file is updated to generate knowledge graph



```
def generate_topics(courseID,type, lectureID, filePath):
    file = open(filePath, 'r', encoding="UTF-8")
    text = file.read()
    try:
        nlp = spacy.load('en_core_web_sm')
        doc = nlp(text)
        # for ent in doc:
        #     if ent.text!=None or ent.text!=' ':
        #         print("-----")
        #         print(ent.text)
        #         spacyDic[ent.text]=(ent.ent_type_, ent.head.pos_)

        for ent in doc.ents:
            spacyDic[ent.text] = ent.label_
        # print(spacyDic)


        nlp = spacy.blank('en')
        nlp.add_pipe('dbpedia_spotlight')
        doc=nlp(text)
        for ent in doc.ents:
            if ent.kb_id_ not in dbDic.values():
                dbDic[ent.text] = ent.kb_id_
                # print(ent.text, ent.label_)
```

Figure: Generating Topics using Spacy and DBpedia Spotlight

Query	Result
Generated Triples	64942
Courses	7164
Events	52
Topics	382

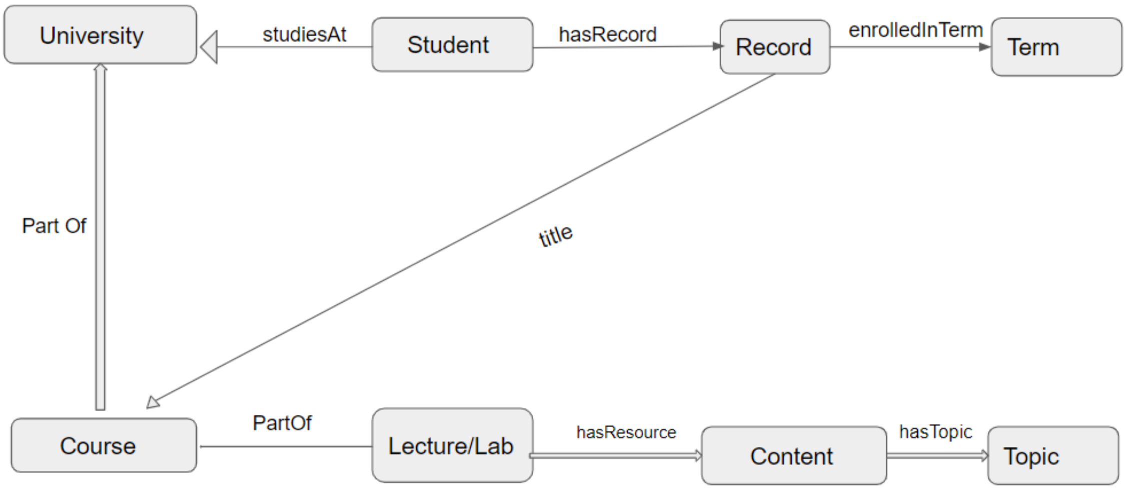


Figure: Relation between different classes

9.University Chatbot Design

The second part of project demanded on natural language processing interface to enable a conversation between an end user and the chatbot. Users can interact with Rasa Chatbot by querying into the knowledge base created above. Queries are mapped corresponding to SPARQL queries and are fed into FUSEKI server, upon which the response generated in form of triples is displayed in human-readable natural language using rasa framework.

For Part-2: New intents will be created, connecting each intent to a query.

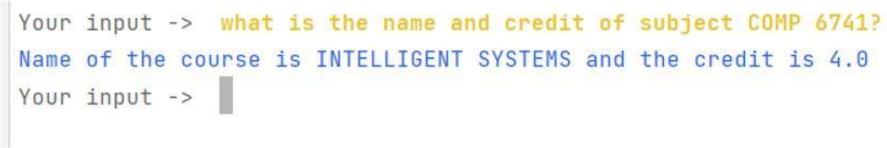
The Rasa stack composes of 2 Major components – (1) Rasa NLU (2) Rasa Core

- Domain.yml: It specifies the intents that chatbot is expecting to give a response over, the slots to track and the actions that chatbot performs.
- Stories.yml: A training sample undertaken for a dialogue system is called a story.
- Nlu.yml: It consists of intents. Each intent maps to a query on Fuseki. For each intent, question can be asked in different way for asking the same question, which can be taken into account using [oov__] representing out of vocabulary.
- Actions.py: Actions .py lists down all the actions which the chatbot needs to perform

10.Rasa- Framework Queries

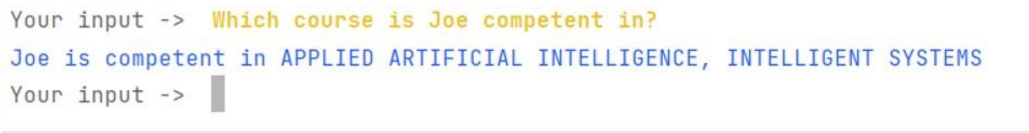
(a) All the Competency questions from Part #1

# Question 1: What is the name and credit of [COMP] (course) [6741] (course_number)?



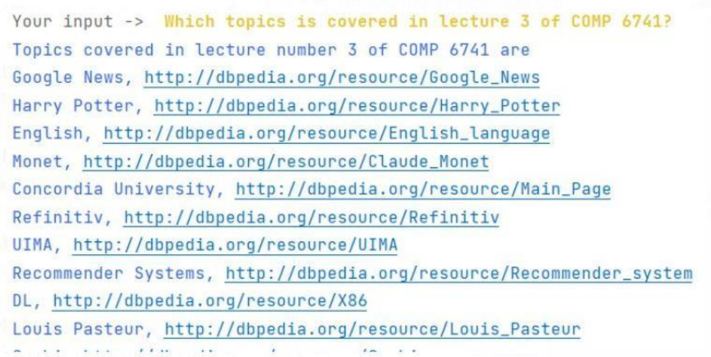
```
Your input ->  what is the name and credit of subject COMP 6741?
Name of the course is INTELLIGENT SYSTEMS and the credit is 4.0
Your input ->
```

# Question 2: Which course and topics is [Joe](person) [_oov] competent in?



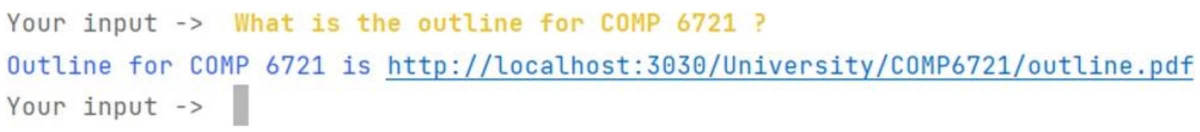
```
Your input ->  Which course is Joe competent in?
Joe is competent in APPLIED ARTIFICIAL INTELLIGENCE, INTELLIGENT SYSTEMS
Your input ->
```

# Question 3: Which topics is covered in lecture [3](lecture_number) of [COMP](course) [6741](course_number)?



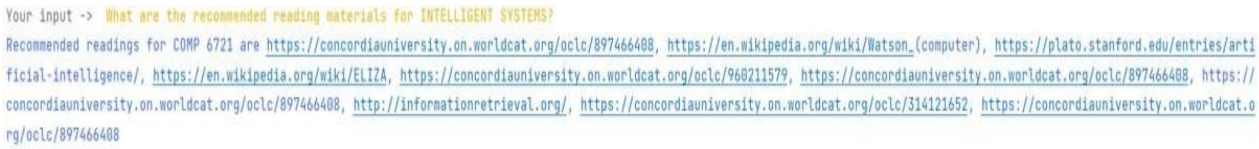
```
Your input ->  Which topics is covered in lecture 3 of COMP 6741?
Topics covered in lecture number 3 of COMP 6741 are
Google News, http://dbpedia.org/resource/Google_News
Harry Potter, http://dbpedia.org/resource/Harry_Potter
English, http://dbpedia.org/resource/English_language
Monet, http://dbpedia.org/resource/Claude_Monet
Concordia University, http://dbpedia.org/resource/Main_Page
Refinitiv, http://dbpedia.org/resource/Refinitiv
UIMA, http://dbpedia.org/resource/UIMA
Recommender Systems, http://dbpedia.org/resource/Recommender_system
DL, http://dbpedia.org/resource/X86
Louis Pasteur, http://dbpedia.org/resource/Louis_Pasteur
```

# Question 4: What is the outline for [COMP](course) [6741](course_number)??



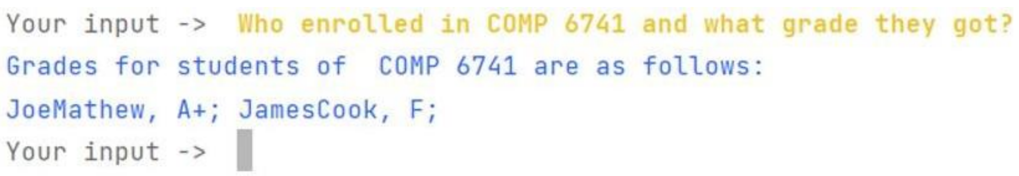
```
Your input ->  What is the outline for COMP 6721 ?
Outline for COMP 6721 is http://localhost:3030/University/COMP6721/outline.pdf
Your input ->
```

# Question 5: What are the recommended reading materials for [INTELLIGENT SYSTEMS](course_name)?



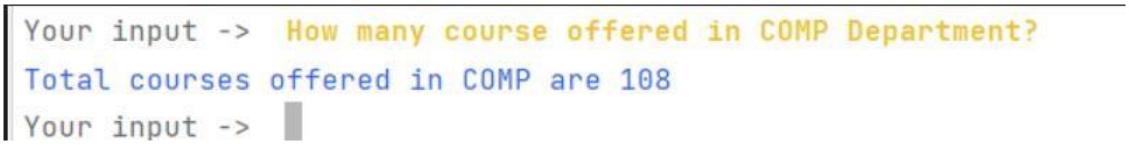
```
Your input ->  What are the recommended reading materials for INTELLIGENT SYSTEMS?
Recommended readings for COMP 6721 are https://concordiauniversity.on.worldcat.org/oclc/897466408, https://en.wikipedia.org/wiki/Watson_(computer), https://plato.stanford.edu/entries/arti
ficial-intelligence/, https://en.wikipedia.org/wiki/ELIZA, https://concordiauniversity.on.worldcat.org/oclc/960211579, https://concordiauniversity.on.worldcat.org/oclc/897466408, https://
concordiauniversity.on.worldcat.org/oclc/897466408, http://informationretrieval.org/, https://concordiauniversity.on.worldcat.org/oclc/314121652, https://concordiauniversity.on.worldcat.o
rg/oclc/897466408
```

# Question 6: Who enrolled in [COMP](course) [6741](course_number) and what grade they got?



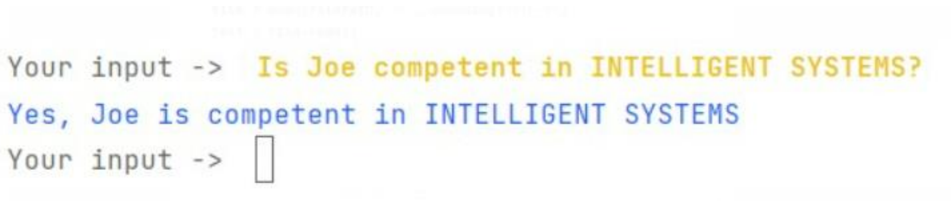
```
Your input ->  Who enrolled in COMP 6741 and what grade they got?
Grades for students of  COMP 6741 are as follows:
JoeMathew, A+; JamesCook, F;
Your input ->
```

# Question 7: How many course offered in [COMP](course) Department?



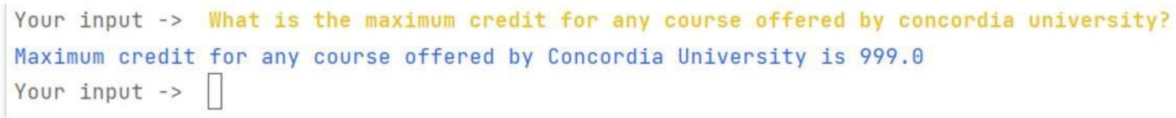
```
Your input ->  How many course offered in COMP Department?
Total courses offered in COMP are 108
Your input ->
```

#Question 8: Is [Joe](person) competent in [INTELLIGENT SYSTEMS](course_name)?



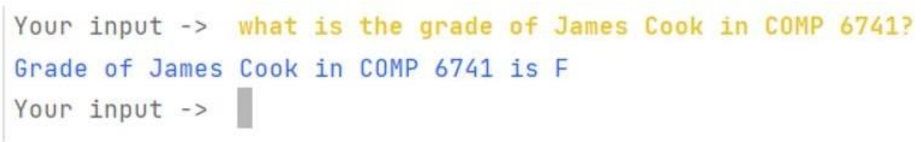
```
Your input ->  Is Joe competent in INTELLIGENT SYSTEMS?
Yes, Joe is competent in INTELLIGENT SYSTEMS
Your input ->
```

# Question 9: What is the maximum credit for any course offered by concordia university?



```
Your input ->  What is the maximum credit for any course offered by concordia university?
Maximum credit for any course offered by Concordia University is 999.0
Your input ->
```

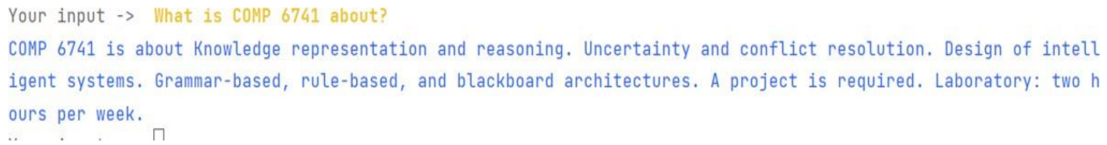
# Question 10: What is the grade of [James](givenName) [Cook](familyName) in [COMP](course) [6741](course_number)?



```
Your input ->  what is the grade of James Cook in COMP 6741?
Grade of James Cook in COMP 6741 is F
Your input ->
```

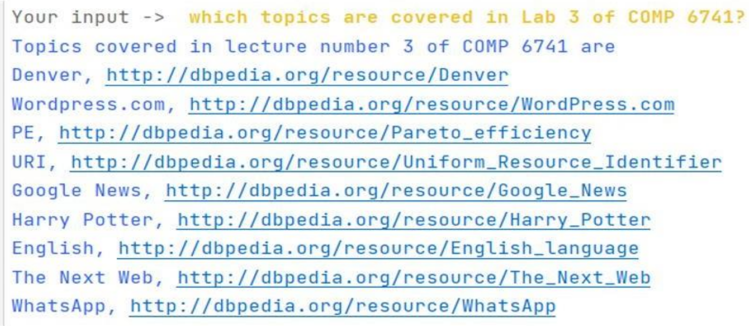
(b) Additional Queries:

# Question 1: What is [COMP](course) [474](course_number) about?



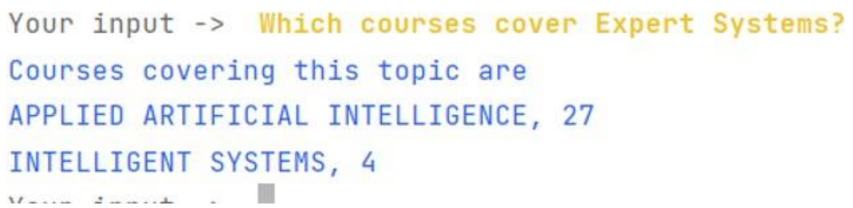
```
Your input -> What is COMP 6741 about?
COMP 6741 is about Knowledge representation and reasoning. Uncertainty and conflict resolution. Design of intell
igent systems. Grammar-based, rule-based, and blackboard architectures. A project is required. Laboratory: two h
ours per week.
```

# Question 2: Which topics are covered in Lab [2] (lab_number) of [COMP](course) [6741](course_number)?



```
Your input -> which topics are covered in Lab 3 of COMP 6741?
Topics covered in lecture number 3 of COMP 6741 are
Denver, http://dbpedia.org/resource/Denver
Wordpress.com, http://dbpedia.org/resource/WordPress.com
PE, http://dbpedia.org/resource/Pareto_efficiency
URI, http://dbpedia.org/resource/Uniform_Resource_Identifier
Google News, http://dbpedia.org/resource/Google_News
Harry Potter, http://dbpedia.org/resource/Harry_Potter
English, http://dbpedia.org/resource/English_language
The Next Web, http://dbpedia.org/resource/The_Next_Web
WhatsApp, http://dbpedia.org/resource/WhatsApp
```

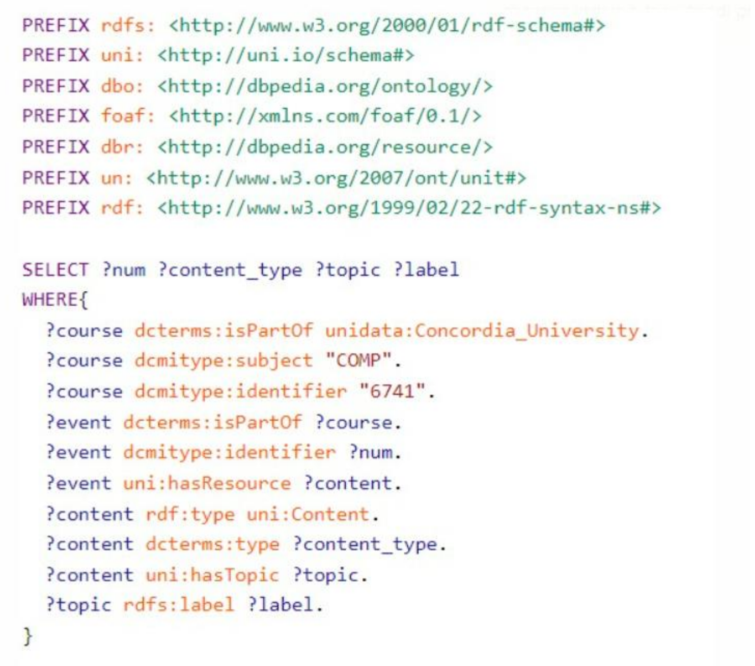
# Question 3: Which course cover [Subject]? :- Which course cover Expert Systems?



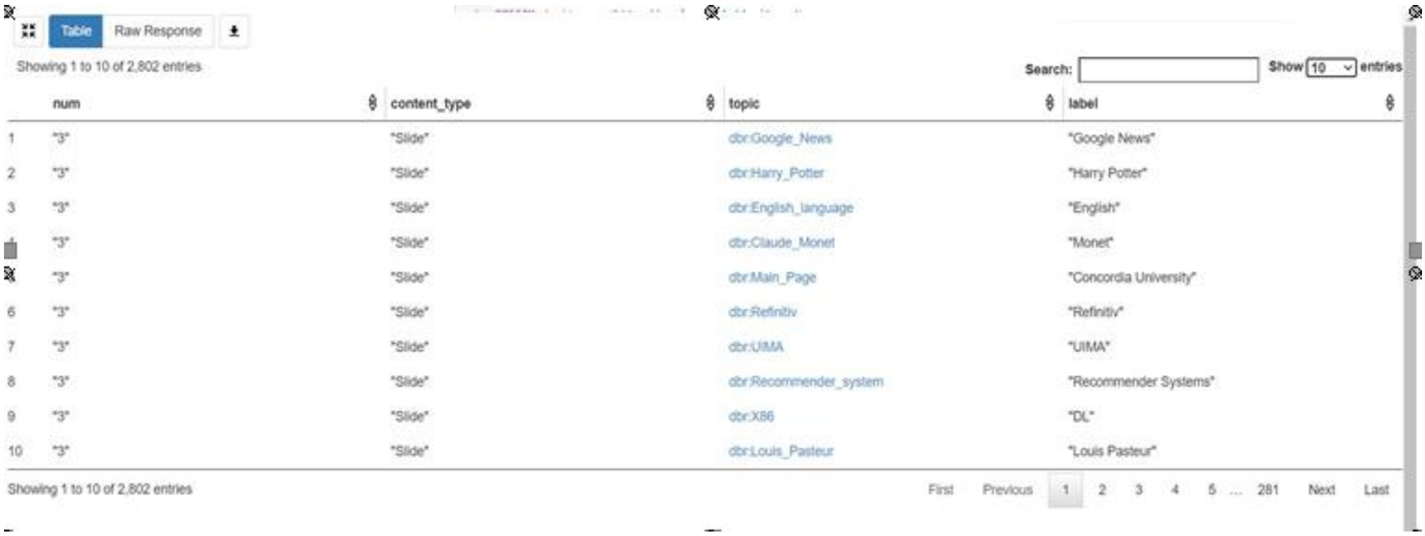
```
Your input -> Which courses cover Expert Systems?
Courses covering this topic are
APPLIED ARTIFICIAL INTELLIGENCE, 27
INTELLIGENT SYSTEMS, 4
```

(c) Knowledge Base Population- Triplication (SPARQL Queries)

# **Query 1:** For a course c, list all covered topics t, printing out their English labels and their DBpedia URI, together with the course event URI (e.g., ’lab3’) and resource URI (e.g., ’slides10’) where they appeared.



```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX uni: <http://uni.io/schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
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#>

SELECT ?num ?content_type ?topic ?label
WHERE{
  ?course dcterms:isPartOf unidata:Concordia_University.
  ?course dcmitype:subject "COMP".
  ?course dcmitype:identifier "6741".
  ?event dcterms:isPartOf ?course.
  ?event dcmitype:identifier ?num.
  ?event uni:hasResource ?content.
  ?content rdf:type uni:Content.
  ?content dcterms:type ?content_type.
  ?content uni:hasTopic ?topic.
  ?topic rdfs:label ?label.
}
```



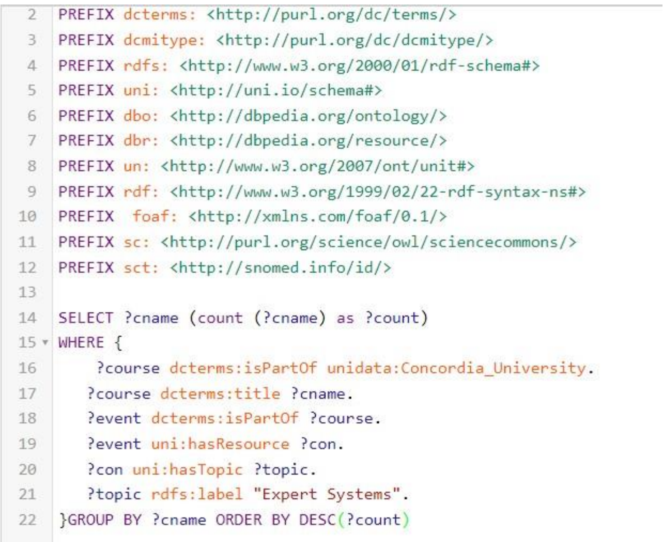
Showing 1 to 10 of 2,802 entries

Search: Show 10 entries

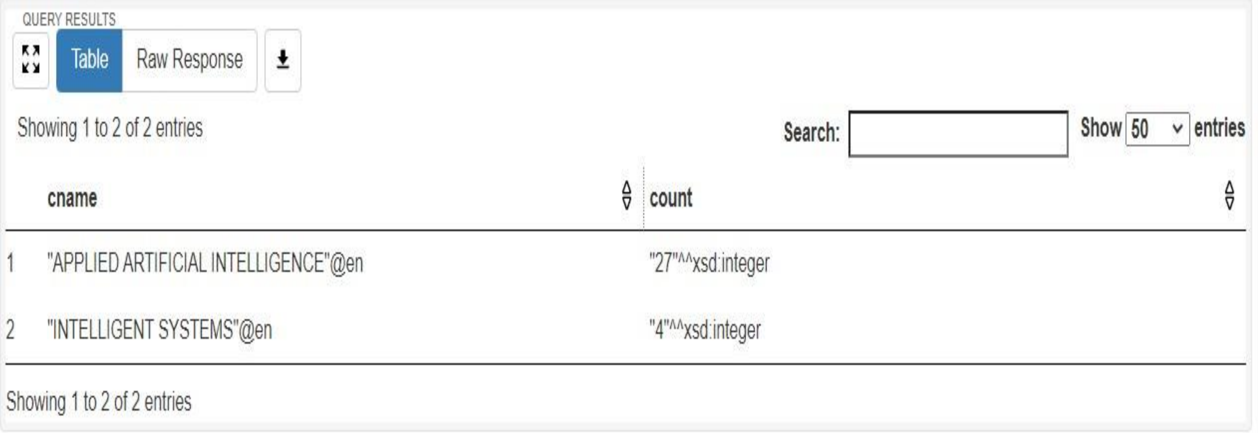
	num	content_type	topic	label
1	"3"	"Slide"	dbr:Google_News	"Google News"
2	"3"	"Slide"	dbr:Harry_Potter	"Harry Potter"
3	"3"	"Slide"	dbr:English_language	"English"
4	"3"	"Slide"	dbr:Claude_Monet	"Monet"
5	"3"	"Slide"	dbr:Main_Page	"Concordia University"
6	"3"	"Slide"	dbr:Refinitiv	"Refinitiv"
7	"3"	"Slide"	dbr:UIMA	"UIMA"
8	"3"	"Slide"	dbr:Recommender_system	"Recommender Systems"
9	"3"	"Slide"	dbr:X86	"DL"
10	"3"	"Slide"	dbr:Louis_Pasteur	"Louis Pasteur"

Showing 1 to 10 of 2,802 entries

# Query 2: For a given topic t (DBpedia URI), list all courses where they appear, together with a count, sorted by frequency.



```
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX dcmitype: <http://purl.org/dc/dcmitype/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
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#>
PREFIX  foaf: <http://xmlns.com/foaf/0.1/>
PREFIX sc: <http://purl.org/science/owl/sciencecommons/>
PREFIX sct: <http://snomed.info/id/>

SELECT ?cname (count (?cname) as ?count)
WHERE {
    ?course dcterms:isPartOf unidata:Concordia_University.
   ?course dcterms:title ?cname.
   ?event dcterms:isPartOf ?course.
   ?event uni:hasResource ?con.
   ?con uni:hasTopic ?topic.
   ?topic rdfs:label "Expert Systems".
}GROUP BY ?cname ORDER BY DESC(?count)
```



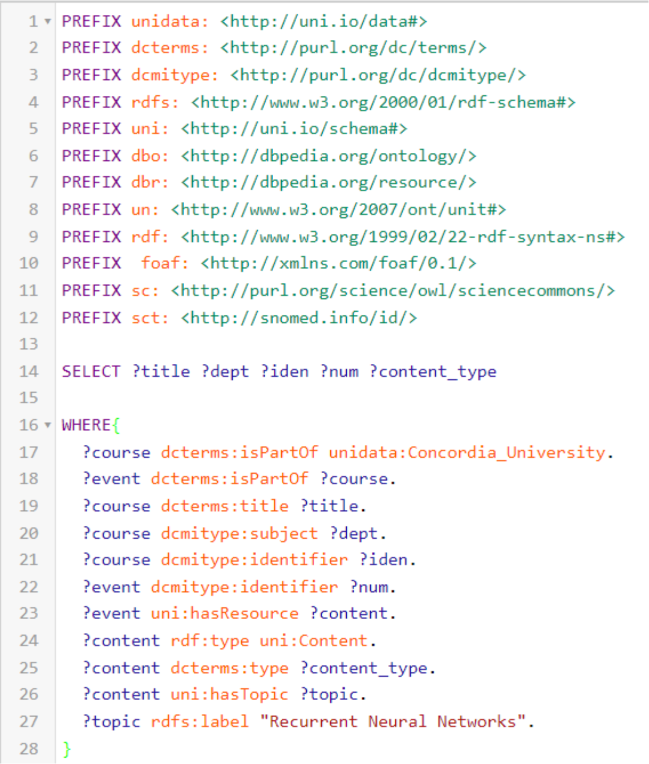
Showing 1 to 2 of 2 entries

Search: Show 50 entries

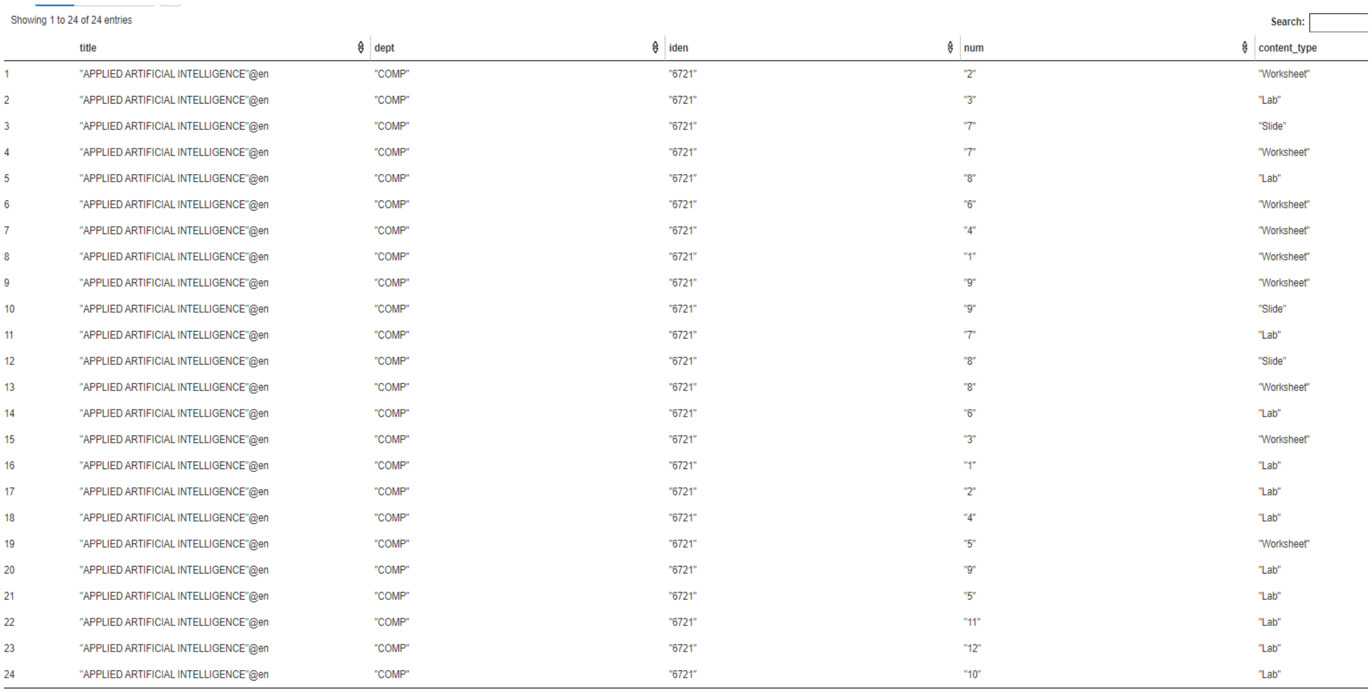
	cname	count
1	"APPLIED ARTIFICIAL INTELLIGENCE"@en	"27"^^xsd:integer
2	"INTELLIGENT SYSTEMS"@en	"4"^^xsd:integer

Showing 1 to 2 of 2 entries

# **Query 3:** For a given topic t, list the precise course URI, course event URI and corresponding resource URI where the topic is covered (e.g., “NLP” is covered in COMP474 → Lecture 10→ Lab 10 → Lab Notes).



```
PREFIX unidata: <http://uni.io/data#>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX dcmitype: <http://purl.org/dc/dcmitype/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
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#>
PREFIX  foaf: <http://xmlns.com/foaf/0.1/>
PREFIX sc: <http://purl.org/science/owl/sciencecommons/>
PREFIX sct: <http://snomed.info/id/>

SELECT ?title ?dept ?iden ?num ?content_type

WHERE{
  ?course dcterms:isPartOf unidata:Concordia_University.
  ?event dcterms:isPartOf ?course.
  ?course dcterms:title ?title.
  ?course dcmitype:subject ?dept.
  ?course dcmitype:identifier ?iden.
  ?event dcmitype:identifier ?num.
  ?event uni:hasResource ?content.
  ?content rdf:type uni:Content.
  ?content dcterms:type ?content_type.
  ?content uni:hasTopic ?topic.
  ?topic rdfs:label "Recurrent Neural Networks".
}
```

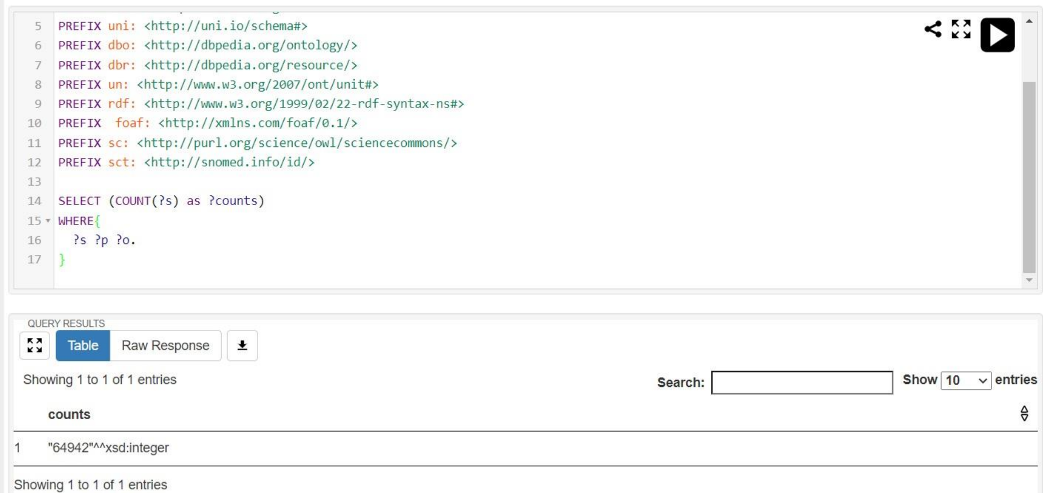


Showing 1 to 24 of 24 entries

Search:

	title	dept	iden	num	content_type
1	"APPLIED ARTIFICIAL INTELLIGENCE"@en	"COMP"	"6721"	"2"	"Worksheet"
2	"APPLIED ARTIFICIAL INTELLIGENCE"@en	"COMP"	"6721"	"3"	"Lab"
3	"APPLIED ARTIFICIAL INTELLIGENCE"@en	"COMP"	"6721"	"7"	"Slide"
4	"APPLIED ARTIFICIAL INTELLIGENCE"@en	"COMP"	"6721"	"7"	"Worksheet"
5	"APPLIED ARTIFICIAL INTELLIGENCE"@en	"COMP"	"6721"	"8"	"Lab"
6	"APPLIED ARTIFICIAL INTELLIGENCE"@en	"COMP"	"6721"	"6"	"Worksheet"
7	"APPLIED ARTIFICIAL INTELLIGENCE"@en	"COMP"	"6721"	"4"	"Worksheet"
8	"APPLIED ARTIFICIAL INTELLIGENCE"@en	"COMP"	"6721"	"1"	"Worksheet"
9	"APPLIED ARTIFICIAL INTELLIGENCE"@en	"COMP"	"6721"	"9"	"Worksheet"
10	"APPLIED ARTIFICIAL INTELLIGENCE"@en	"COMP"	"6721"	"9"	"Slide"
11	"APPLIED ARTIFICIAL INTELLIGENCE"@en	"COMP"	"6721"	"7"	"Lab"
12	"APPLIED ARTIFICIAL INTELLIGENCE"@en	"COMP"	"6721"	"8"	"Slide"
13	"APPLIED ARTIFICIAL INTELLIGENCE"@en	"COMP"	"6721"	"8"	"Worksheet"
14	"APPLIED ARTIFICIAL INTELLIGENCE"@en	"COMP"	"6721"	"6"	"Lab"
15	"APPLIED ARTIFICIAL INTELLIGENCE"@en	"COMP"	"6721"	"3"	"Worksheet"
16	"APPLIED ARTIFICIAL INTELLIGENCE"@en	"COMP"	"6721"	"1"	"Lab"
17	"APPLIED ARTIFICIAL INTELLIGENCE"@en	"COMP"	"6721"	"2"	"Lab"
18	"APPLIED ARTIFICIAL INTELLIGENCE"@en	"COMP"	"6721"	"4"	"Lab"
19	"APPLIED ARTIFICIAL INTELLIGENCE"@en	"COMP"	"6721"	"5"	"Worksheet"
20	"APPLIED ARTIFICIAL INTELLIGENCE"@en	"COMP"	"6721"	"9"	"Lab"
21	"APPLIED ARTIFICIAL INTELLIGENCE"@en	"COMP"	"6721"	"5"	"Lab"
22	"APPLIED ARTIFICIAL INTELLIGENCE"@en	"COMP"	"6721"	"11"	"Lab"
23	"APPLIED ARTIFICIAL INTELLIGENCE"@en	"COMP"	"6721"	"12"	"Lab"
24	"APPLIED ARTIFICIAL INTELLIGENCE"@en	"COMP"	"6721"	"10"	"Lab"

(d) Statistical Query :



```
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#>
PREFIX  foaf: <http://xmlns.com/foaf/0.1/>
PREFIX sc: <http://purl.org/science/owl/sciencecommons/>
PREFIX sct: <http://snomed.info/id/>

SELECT (COUNT(?s) as ?counts)
WHERE{
  ?s ?p ?o.
}
```

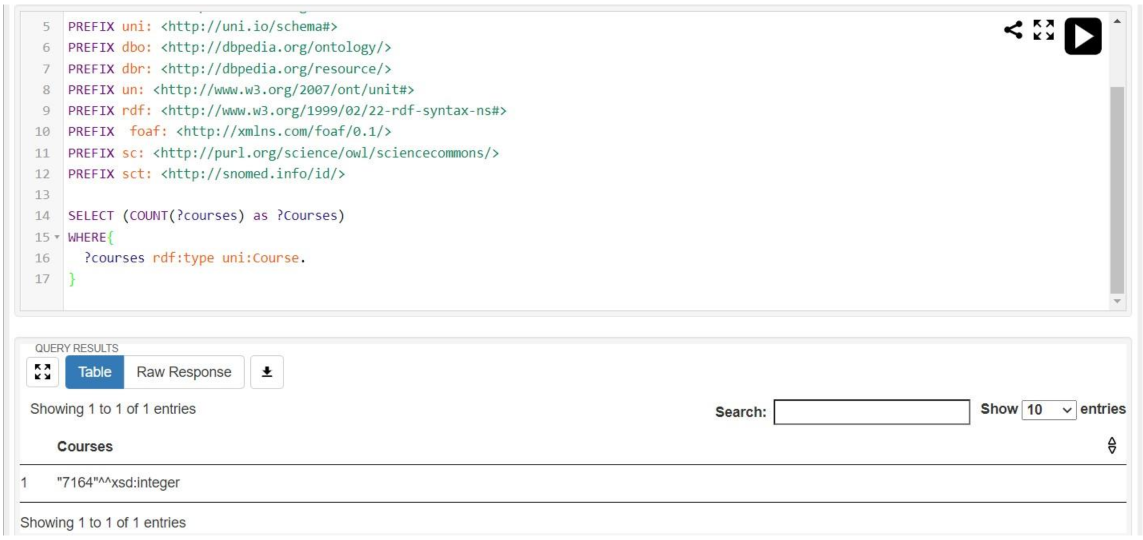
QUERY RESULTS

Showing 1 to 1 of 1 entries

	counts
1	"64942"^^xsd:integer

Showing 1 to 1 of 1 entries

Figure: Description of statistics in context to triples used



```
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#>
PREFIX  foaf: <http://xmlns.com/foaf/0.1/>
PREFIX sc: <http://purl.org/science/owl/sciencecommons/>
PREFIX sct: <http://snomed.info/id/>

SELECT (COUNT(?courses) as ?Courses)
WHERE{
  ?courses rdf:type uni:Course.
}
```

QUERY RESULTS

Showing 1 to 1 of 1 entries

	Courses
1	"7164"^^xsd:integer

Showing 1 to 1 of 1 entries

Figure : Description of total number of courses offered

(e) 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 and then on 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.

Part III: 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/