



Concordia University

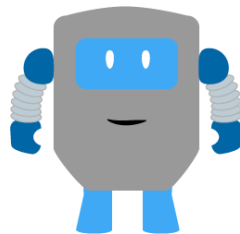
# Engineering and Computer Science

**COMP 6741**

**Intelligent  
Systems**

Project-1 Report

**Unibot**



Submitted To: Prof.  
Dr. René Witte

Date : 13 April, 2021

Team FL\_G\_03

First Name	Last Name	Student ID	Email ID
Chirag	Patel	40160656	<a href="mailto:chiragpatel2965@gmail.com">chiragpatel2965@gmail.com</a>
Harshil	Patel	40163431	<a href="mailto:harshilpatel1903@gmail.com">harshilpatel1903@gmail.com</a>

## Table of Contents

<b>1. Abstract .....</b>	<b>3.</b>
<b>2. Competency Questions .....</b>	<b>3.</b>
<b>3. Vocabulary .....</b>	<b>3.</b>
<b>4. Schema .....</b>	<b>7.</b>
<b>5. Knowledge base Construction .....</b>	<b>9.</b>
<b>6. Queries .....</b>	<b>10.</b>
<b>7. Project Part 2 .....</b>	<b>14.</b>
<b>8. References .....</b>	<b>21.</b>

## ABSTRACT:

The overall goal of this project is to build Unibot, an intelligent agent that can answer university course- and student-related questions, using a knowledge graph and natural language processing.

The first phase of our project was to gather information on all Concordia University courses. This information was used to generate a Knowledge Graph using existed and newly created vocabularies. A series of competency questions are written and transformed into queries to test the functionalities of generated Knowledge Base.

## Competency Questions

1. How many courses are offered at Concordia?
2. Which topics are covered in COMP 6741 lectures?
3. Which topics is Arihant is competent in?
4. Which courses at Concordia teaches deep learning?
5. Where can I get more information about Comp6721?
6. What are grades of Brendon Rihards?
7. Which COMP courses are taught at Concordia?
8. Which lecture of Comp6741 covers knowledge graph?
9. Does ACCO230 and ACCO350 cover similar topic?
10. How many students have taken COMP6721 courses?

## Vocabulary:

*Description how you modeled the schema for your knowledge base, including the vocabularies you reused, any vocabulary extensions you developed, etc. Give brief justifications where appropriate (e.g., choice of existing vocabulary).*

Table 1 : Vocabulary

RDF	Type	For defining classes and properties
	Property	
RDFS	Class	For defining classes
	subClassOf	For extending class definitions
	domain	For defining properties
	range	

FOAF	Person	A student is a person
	topic	Property used to relate a topic to a university course
	mbox	Mail info of the student
DBO	document	For defining our transcripts document seemed a fitting concept.
	type	For use of the Public_university type
DBR	Course_(education)	Suitable for broadly defining the courses offered at universities
	Public_university	Suitable definition for Universities
XSD	string	Literal strings serve as the object of several properties we have generated.
DCTERMS	relation	A related resource.
DCMITYPE	event	A non-persistent, time-based occurrence.

We used the focu schema <<http://focu.io/schema#>> and the ex schema <<http://example.org/>> to contain instances of the remaining classes and properties. Their definitions and details concerning their usage are given below. Each class and property created for our vocabulary contains an `rdfs:label` and `rdfs:comment` property as per project requirement.

1) **focu:Student**

```
a rdfs:Class;
rdfs:subClassOf foaf:Person;
```

Class to describe a student at the university

2) **focu:Lecture**

```
a rdfs:Class;
rdfs:subClassOf dcmitype:Event;
```

Class to describe lecture as a Dublin Core Event

3) **focu:Course**

```
a rdfs:Class;
rdfs:subClassOf dbr:Course_(education);
```

Class to describe Course at the university in the combined form of Subject + Catalog

4) **focu:courseTaken**

```
a rdf:property ;
rdfs:domain <Student> ;
rdfs:range <Course>;
```

Property to describe what course has been taken by a Student

5) **focu:hasContent**

```
a rdf:property ;
```

```
rdfs:subClassOf dct:relation ;  
rdfs:domain <Lecture> ;  
rdfs:range xsd:string;
```

Property to describe content for a specific lecture

6) **focu:competencies**

```
a rdf:Property;  
rdfs:domain <Student>;  
rdfs:range xsd:string;
```

Property to describe topics in which a Student is competent in

7) **focu:Record**

```
a dbo:document;
```

Property to describe information regarding course and grade achieved by a Student

8) **focu:hasRecord**

```
a rdf:Property;  
rdfs:domain <Student>;  
rdfs:range <Record>;
```

Property to describe what records a Student has

9) **focu:subject**

```
a rdf:Property;  
rdfs:domain <course>;  
rdfs:range xsd:string;
```

Property to describe course subject (“COMP”, “SOEN”)

10) **focu:catalog**

```
a rdf:Property;  
rdfs:domain <course>;  
rdfs:range xsd:string;
```

Property to describe course catalog number (“474”, “6741”)

11) **focu:credits**

```
a rdf:Property;  
rdfs:domain <course>;  
rdfs:range xsd:int;
```

Property to describe number of credits a course contains

12) **focu:grade**

```
a rdf:Property;  
rdfs:domain <Record>;  
rdfs:range xsd:string;
```

Property to describe what grade is scored by a student in particular course

13) **focu:provenance**

```
a rdf:Property;  
rdfs:domain foaf:topic;  
rdfs:range xsd:string;
```

Property to describe the source of topic identified for the course

14) **focu:firstName**

```
a rdf:Property;  
rdfs:domain <Student>;  
rdfs:range xsd:string;
```

Property to describe first name of the student

15) **focu:lastName**

```
a rdf:Property;  
rdfs:domain <Student>;  
rdfs:range xsd:string;
```

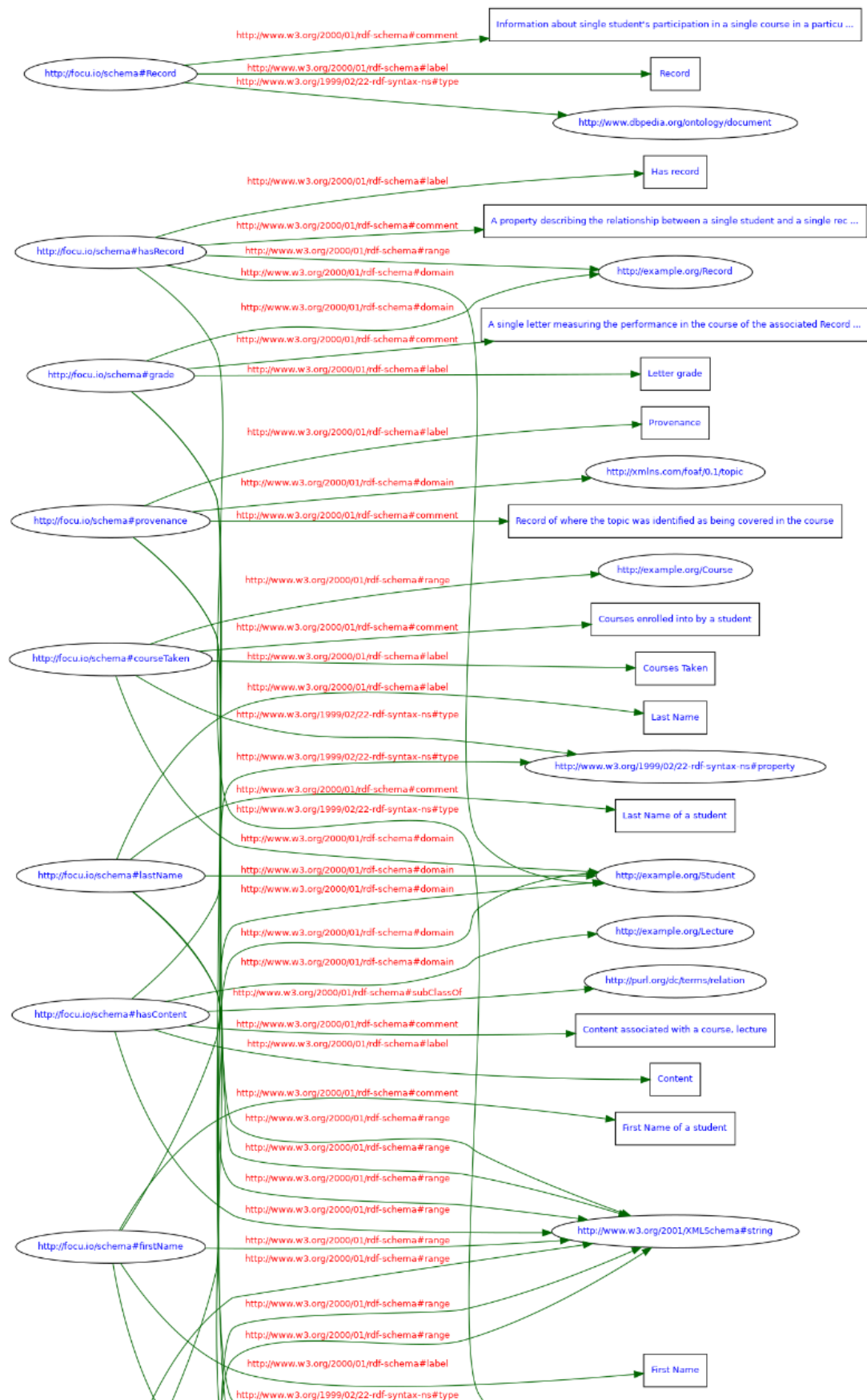
Property to describe last name of the student

16) **focu:offeredAt**

```
a rdf:Property;  
rdfs:domain focu:Course;  
rdfs:range dbr:Public_university;
```

Property to describe what course is offered at what university

## Schema Design :







## Knowledge Base Construction:

*Describe (a) your dataset and (b) your process and developed tools for populating the knowledge base from the dataset.*

*Describe how to run the tools to create the knowledge base. Explain your process for linking entities to DBpedia*

Knowledge Base was build using Python and following libraries:

**Pandas** - for reading and pre-processing csv files

**Spacy** - to perform DBpedia Spotlight annotation on course descriptions

**Rdflib** – To create knowledge graph using **Graph** and use existing namespaces provided by rdflib

### Database description :

Database was generated using files obtains from **Concordia opendata** datasets, namely course catalog and course description files were used. Using pandas dataframe, both the files were merged into one :

```
df = pd.merge(a, b, on = "Course ID", how="inner")
```

Unwanted columns were then dropped. The final output was saved to “**dataset.csv**”.

The file contains 7050 datapoints each of which contains the following values :

Course ID, Subject, Number, Name, Credits, Description

There are 2 additional files “**lecture\_data.csv**” and “**content\_data.csv**” which contains following values respectively:

CourseId, Identifier, Title, seeAlso, Topic

CourseId, Identifier, ContentType, Content

Additionally, a “**students.csv**” file contains information about randomly generated students using

**Students.py** :

studentID, firstName, lastName, Email, Subjects, SubjectsGrades, Competencies

### Generating Knowledge Graph :

Knowledge Graph was generated using **main.py**.

All the csv files from dataset folder were loaded using pandas. For each data point in the dataset.csv, following values were added to the graph :

course\_id, course\_subject, course\_number, course\_title, course\_credit,  
course\_description, course\_topics, course\_link

Using Spacy and it’s connection to DBpedia Spotlight, the course title and course descriptions were annotated with a confidence score or 0.4 and were stored in course topics list.

```
course_topics = list(set([(ent.text, ent.kb_id_) for ent in topic_ents]))
```

Each topic was then added to the graph along with it's course. Similarly, lecture data and content of lecture matching to the course were split into list and added as triples to the graph.

After this, randomly generated information of 100 students were created and appropriate triples for them were added to the graph.

Finally the graph was stored in a serialize format of both N-triple and Turtle namely **knowledge\_base\_n3.nt** and **knowledge\_base\_ttl.ttl**

### Generated Knowledge Base:

**Triples – 111284**  
**Course – 7025**  
**Lectures – 26**  
**Contents – 118**  
**Students – 100**

### Queries :

⇒ The following queries have been used to test the graph. The queries can be found in the accompanying queries.sparql file. In the report, we will limit to 10 queries, but more are defined in the queries.sparql file

- Question 1  
How many courses are offered at Concordia?

**Query**  
SELECT (count(?courseId) as ?CourseCount)  
WHERE {  
    ?courseId a focu:Course.}

**Result**  
"7009"^^<http://www.w3.org/2001/XMLSchema#integer>

- Question 2  
Which topics are covered in COMP 6741 lectures?

**Query**  
SELECT ?topics WHERE {  
    ?sub focu:subject "COMP" .  
    ?sub focu:catalog "6741".  
    ?sub rdfs:comment ?topics .  
}

**Result**  
Knowledge representation and reasoning. Uncertainty and conflict resolution. Design of intelligent systems. Grammar-based,

rule-based, and blackboard architectures.

- Question 3  
Which topics is Arihant competent in?

**Query**

```
SELECT ?Competencies WHERE {  
    ?student a focu:Student.  
    ?student focu:firstName "Arihant".  
    ?student focu:competencies ?Competencies.  
}
```

**Result**

```
1 automated reasoning  
2 audit  
3 A.I.  
4 CGA  
5 heuristic  
6 recurrent neural networks  
7 knowledge representation  
8 Artificial Intelligence  
9 ACCO  
10 assurance services
```

- Question 4  
Which courses at Concordia teaches deep learning?

**Query**

```
SELECT ?subjects WHERE {  
    ?subjects foaf:topic dbr:deep_learning.  
}
```

**Result**

```
1 <http://example.org/CEPS1114E0>  
2 <http://example.org/COEN432>  
3 <http://example.org/COMP432>
```

- Question 5  
Where can I get more information about Comp6721?

**Query**

```
SELECT ?information  
WHERE {  
    ?subject a focu:Course.  
    ?subject focu:subject "COMP".  
    ?subject focu:catalog "6721".  
    ?subject rdfs:seeAlso ?information.  
}
```

**Result**

```
1 ('A.I.', 'http://dbpedia.org/resource/Artificial_intelligence')
```

- Question 6  
What are grades of Brendon Rihards?

#### Query

```
SELECT ?course ?grade WHERE {
  ?student a focu:Student.
  ?student focu:firstName "Brendon".
  ?student focu:lastName "Rihards".
  ?student focu:hasRecord ?record.
  ?record focu:courseTaken ?course.
  ?record focu:grade ?grade
}
```

#### Result

```
1 http://example.org/SOEN6441 A-
2 http://example.org/ACCO465 B+
3 http://example.org/COMP6741 B
```

- Question 7  
Which COMP courses are taught at Concordia?

#### Query

```
SELECT ?subjects
WHERE {
  ?subjects a focu:Course.
  ?subjects focu:subject "COMP".
} LIMIT 5
```

#### Result

```
1 <http://example.org/COMP6651>
2 <http://example.org/COMP371>
3 <http://example.org/COMP6661>
4 <http://example.org/COMP6521>
5 <http://example.org/COMP492>
```

- Question 8  
Which lecture of Comp6741 covers knowledge graph?

#### Query

```
SELECT ?lecture
WHERE {
  ?subject a focu:Course.
  ?subject focu:subject "COMP".
  ?subject focu:catalog "6741".
  ?lecture dcterms:isPartOf ?subject.
  ?lecture foaf:topic "Knowledge_Graph".
}
```

#### Result

```
1 <http://example.org/COMP6741\_Lec2>
```

- Question 9  
Does ACCO230 and ACCO350 cover similar topic?

#### Query

```

SELECT ?topics
WHERE {
    ?subject1 a focu:Course.
    ?subject1 focu:subject "ACCO".
    ?subject1 focu:catalog "230".
    ?subject1 foaf:topic ?topics.
    ?subject2 a focu:Course.
    ?subject2 focu:subject "ACCO".
    ?subject2 focu:catalog "350".
    ?subject2 foaf:topic ?topics.

}

```

### Result

1 <<http://www.dbpedia.org/resource/Accounting>>

- Question 10  
How many students have taken COMP6721 courses?

### Query

```

SELECT ?firstName ?lastName WHERE {
    ?student a focu:Student.
    ?student focu:firstName ?firstName .
    ?student focu:lastName ?lastName.
    ?student focu:courseTaken ?course.
    ?course a focu:Course.
    ?course focu:subject "COMP".
    ?course focu:catalog "6721".
} LIMIT 5

```

### Result

```

1 Dante Walter
2 Bartlomiej Zubair
3 Bruin Thomas-Jay
4 Aaryn Struan
5 Ardeshir Madison

```

### Updates after Project part 1:

**Modified KB to remove redundant course name information from student data and only kept the info in the records of each student. No other changes were asked.**

## PROJECT PART 2 :

### New Queries listed in project description

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.

```
SELECT ?topic ?uri ?source ?lectureNum
WHERE {
  ?sub a focu:Course.
  ?sub focu:subject "COMP" .
  ?sub focu:catalog "6741".
  ?lecture dcterms:isPartOf ?sub.
  ?lecture dcmitype:identifier ?lectureNum.
  ?lecture foaf:topic ?topicid.
  ?topicid rdfs:label ?topic.
  ?topicid rdfs:seeAlso ?uri.
  ?topicid focu:topicSource ?source
}
```

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

```
SELECT ?cname (count (?cname) as ?count)
WHERE {
  ?sub a focu:Course.
  ?sub focu:courseName ?cname.
  ?lecture dcterms:isPartOf ?sub.
  ?lecture foaf:topic ?topicid.
  ?topicid rdfs:label "AI".
}
GROUP BY ?cname ORDER BY DESC(?count)
```

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)

```
SELECT ?cname ?uri ?lectureNum ?source
WHERE {
  ?sub a focu:Course.
  ?sub focu:courseName ?cname.
  ?lecture dcterms:isPartOf ?sub.
  ?lecture dcmitype:identifier ?lectureNum.
  ?lecture foaf:topic ?topicid.
  ?topicid rdfs:label "AI".
  ?topicid rdfs:seeAlso ?uri.
  ?topicid focu:topicSource ?source
}
```

## Chatbot Using Rasa:

Rasa is an open source machine learning framework for automated text and voice-based conversations. Understand messages, hold conversations, and connect to messaging channels and APIs.

For this project, the chatbot will be implemented with Rasa. New intents will be created, and each intent will be linked to a query. New entities will also be added to identify the values the chatbot can query on.

### Intents:

The nlu.yml file has to be modified to include the intents. For each intents, a question will be asked in with different ways of asking the same questions, so that the model can be trained to identify the question intent.

The domain.yml file has to be updated to identify the intents

```
intents:
- greet
- goodbye
- affirm
- deny
- mood_great
- mood_unhappy
- bot_challenge
- about_number_of_courses_offered
- about_course_teaches_topic
- about_person_competency
- about_information
- about_grade
- about_department_courses
- about_lecture_course_topic
- about_similar_topic_courses
- about_students_taken_course
- about_topic_covered_course
- about_course
- about_event
```

## Entities

The domain.yml file has to be modified to identify the different entities to be extracted from the questions, namely (course) which is the course, (event) which the event type e.g. LAB, and (topic)

```
entities:
  - person
  - course
  - topic
  - university
  - grade
  - department
  - lecture
  - event
```

```
slots:
  course:
    type: any
    initial_value: ""
    mappings:
      - type: from_entity
        entity: course
  topic:
    type: any
    initial_value: "initial"
    mappings:
      - type: from_entity
        entity: topic
  university:
    type: any
    initial_value: ""
    mappings:
      - type: from_entity
        entity: university
  person:
    type: any
    initial_value: ""
    mappings:
      - type: from_entity
        entity: person
  grade:
```



## Stories

A single story will be used to basically ask for the questions. The storie.yml file has to be modified to process the different possible intents

```
version: "3.1"

stories:

  - story: get num uni courses
    steps:
      - intent: about_number_of_courses_offered
      - action: action_number_of_courses_offered

  - story: get num uni topics
    steps:
      - intent: about_course_teaches_topic
      - action: action_course_teach_topic

  - story: get comp person
    steps:
      - intent: about_person_competency
      - action: action_person_competency

  - story: get info
    steps:
      - intent: about_information
      - action: action_information

  - story: get department courses
    steps:
      - intent: about_department_courses
      - action: action_department_courses
```

## Actions

The domain.yml has to be modified to include the action. Finally, the actions.py file has to be modified to process the intent.

```

PS C:\Users\cp291\PycharmProjects\COMP-6741\rasa> rasa run actions
2022-04-13 16:54:10 INFO     rasa_sdk.endpoint - Starting action endpoint server...
2022-04-13 16:54:10 INFO     rasa_sdk.executor - Registered function for 'action_number_of_courses_offered'.
2022-04-13 16:54:10 INFO     rasa_sdk.executor - Registered function for 'action_topic_covered_course'.
2022-04-13 16:54:10 INFO     rasa_sdk.executor - Registered function for 'action_person_competency'.
2022-04-13 16:54:10 INFO     rasa_sdk.executor - Registered function for 'action_course_teach_topic'.
2022-04-13 16:54:10 INFO     rasa_sdk.executor - Registered function for 'action_information'.
2022-04-13 16:54:10 INFO     rasa_sdk.executor - Registered function for 'action_grade'.
2022-04-13 16:54:10 INFO     rasa_sdk.executor - Registered function for 'action_department_courses'.
2022-04-13 16:54:10 INFO     rasa_sdk.executor - Registered function for 'action_lecture_course_topic'.
2022-04-13 16:54:10 INFO     rasa_sdk.executor - Registered function for 'action_similar_topic_courses'.
2022-04-13 16:54:10 INFO     rasa_sdk.executor - Registered function for 'action_students_taken_course'.
2022-04-13 16:54:10 INFO     rasa_sdk.executor - Registered function for 'action_course'.
2022-04-13 16:54:10 INFO     rasa_sdk.executor - Registered function for 'action_event'.
2022-04-13 16:54:11 INFO     rasa_sdk.endpoint - Action endpoint is up and running on http://0.0.0.0:5055

```

## Rasa outputs for each competency questions :

Q1)

```

Your input -> How many courses are offered at Concordia?
Concordia University offers a total of 7009 courses

```

Q2)

```

Your input -> Which topics are covered in COMP6741 lectures?
Here are topics covered in COMP6741
Knowledge representation and reasoning. Uncertainty and conflict resolution. Design of intelligent systems.

```

Q3)

```

Your input -> Which topics is Arihant competent in?
Arihant is Competent in
automated reasoning
audit
A.I.
CGA
heuristic
recurrent neural networks
knowledge representation
Artificial Intelligence
ACCO
assurance services

```

Q4)

```

Your input -> Which courses at Concordia teaches deep learning?
Courses which covers deep_learning are
COMP6721 With topic: deep_learning Appearing 10 times
COMP6741 With topic: deep_learning Appearing 5 times

```

Q5)

```
Your input -> Where can I get more information about Comp6721?
Here are some useful links
('A.I.', 'http://dbpedia.org/resource/Artificial_intelligence')
```

Q6)

```
Your input -> What are grades of Brendon?
Here are grades of Brendon
SOEN6441 A-
ACC0465 B+
COMP6741 B
```

Q7)

```
Your input -> Which COMP courses are taught at Concordia?
Here are courses offered by COMP
COMP6651
COMP371
COMP6661
COMP6521
COMP492
```

Q8)

```
Your input -> Which lecture of Comp6741 covers Knowledge_Graph?
Knowledge_Graph is covered in Comp6741 Lecture: Lec2
```

Q9)

```
Your input -> Does ACC0230&COMP6741 cover similar topic?
ACC0230 and COMP6741 covers following similar
No topics for this course
```

Q10)

```
Your input -> How many students have taken COMP6721 courses?
Here is List of students who have taken COMP6721
Blaike
Damian
Finnan
Bohbi
Corran
```

### New Query Q1)

```
Your input -> what is COMP 6741 about?  
COMP6741 is about  
Knowledge representation and reasoning. Uncertainty and conflict resolution. Design of intelligent systems. Grammar-based.
```

### New Query Q2)

```
Your input -> which topics are covered in Lab#2 of COMP 6741?  
Following topics are covered in LAB#2 of COMP6741  
Python  
RDF  
DBpedia  
g1g1  
FOAF  
XOR  
Apache  
Concordia  
Java  
URI  
SPARQL  
University  
JSON LD  
Schema RDFS  
Jena  
XML  
rdfs
```

### New Query Q3)

```
Your input -> Which courses cover AI?  
Courses which covers AI are  
COMP6721 With topic: AI Appearing 10 times  
COMP6741 With topic: AI Appearing 5 times
```

### Running Program :

- 1) Ensure all dependencies are installed in python environment
- 2) Run the main.py
- 3) Run the Fuseki server and create a new dataset unibot and upload knowledge\_base\_ttl.ttl file
- 4) From the queries folder, copy the query that should be executed and run it

## Generated Knowledge Base:

**Triples – 106296**

**Course – 7025**

**Lectures – 26**

**Contents – 118**

**Students – 100**

**Number of unique topics (COMP 6741 combining lab and lecture) – 613**

**Number of unique topics (COMP 6721 combining lab and lecture) - 443**

## 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>
- [8] Remove blanks from text file  
<https://www.kite.com/python/answers/how-to-remove-empty-lines-from-a-string-in-python>
- [9] Spacy to annotate using DBPedia Spotlight  
<https://spacy.io/api>