COMP 6591 - Introduction to Knowledge-Base Systems

Domain specific knowledge using Prolog and NLP

Report by Team T107:
Ananya Varsha (40197012)
Saghana Mahesh Sarma (40198979)
Sivakumaran Malli Janardhanan (40155790)
Vishanth Surresh (40181942)

August 13, 2022

Abstract

Knowledge representation is predominantly used in the field of Artificial Intelligence. But, there is a lack of domain specific data knowledge. In this paper, we intend to implement a knowledge base system for the periodic table in the Prolog language. This will aid the Artificial Intelligence systems to deduce insights from the information housed in the knowledge base. This knowledge-based system also includes an interface through which users can query the system and interact with you. Several facts related to the periodic table will be implemented using Prolog as a knowledge base. It will also include a Python Prolog interface implemented for friendly user interaction.

1 Introduction

The knowledge-based configuration has been used in numerous applications, with natural language processing being just one of them (NLP). It can be challenging to apply one's skills and expertise to choose the optimal model from among the dataset that are often created and useful for machine learning applications. Making sense of unstructured data sets using natural language processing (NLP) enables the automation of crucial decision-making procedures that would otherwise demand time and labour to complete manually. The task of natural language inference (NLI), which has attracted a lot of interest in the field of natural language processing, is to determine if one statement logically implies another. Making software that can comprehend natural language is difficult.[6] The reasoning program should be knowledgeable about both general knowledge and the viewpoints, goals, and objectives of its users. The declarative nature of Prolog makes it potentially simple to write grammar based on the user's objectives. As some of the capabilities of unification and backtracking processes are more suited to NLP, Prolog is appropriate for the construction of natural language interfaces for software applications. Using examples from the domain, such as a small vocabulary, few relationships, and the majority of users making simple queries, we demonstrate how creating NLI for domain-specific knowledge bases and databases is relatively easier.

2 Why we have selected this Journal

The publication of massive, difficult datasets has significantly increased awareness of the NLI problem. Although several open knowledge bases contain different kinds of reasoning data, their use in NLI has not been thoroughly studied. The utilization of the Prolog structure for knowledge base systems and the significance of natural language processing for text and data analytics encouraged us to choose the paper since they provide insight into the significance of these in machine learning.

3 Literature Review

In the last couple of years, numerous researchers have focused on developing NLP for Business Analysis, and these NLP's have been developed both for relational databases and knowledge bases. NLI for complex aggregates was proposed by Gupta[2]. Setlur[7] developed an NLI for visual analysis. His paper employs a probabilistic grammar-based approach with predefined rules that are dynamically updated based on the data visualisations. Some major problems while developing NLI and NLP are due to the natural language phenomenon, such as lack of concise definitions, ambiguity, influence and change of context, etc.

4 Knowledge Base Systems addressed by the paper

Several knowledge base systems such as Prolog and its abstractions have been utilised in this project. We have implemented a large knowledge base containing several facts regarding the periodic table which can be seen in the below sections. Certain concepts learnt in the course, such as lists, dynamic fast, complex structures, etc have been utilised. The process of creating a seamless connection between Prolog instance Python have also been studied and implemented in this project.

5 Implementation

5.1 Architecture

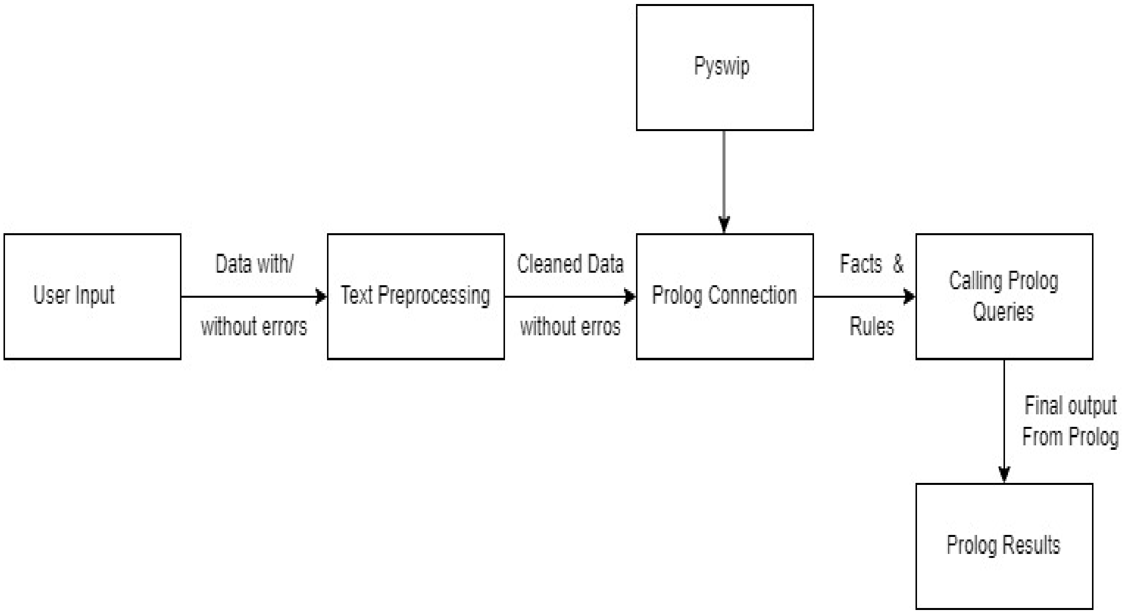


Figure 1: Overall Architecture

To develop a domain-specific interface, we are going to use adequate knowledge representations of entities in the system.[6] First order logic programming construct will be used in developing this knowledge base. Our system will implement the schema for the domain-specific language known as ”entity network”. Every schema entry is represented in the form of ”Entity Association Entity” conveying that, the two entities are confined together by the given ”Association”.[1] An example of the above statement would be ”magnetism of a symbol.” The knowledge base developed will have an organised and structured Prolog predicates, i.e facts and rules. The facts consist of elements and several properties of the periodic table. The rules about the periodic table will be represented using the schema of relations. A Python-based console application[4] acting as an interface between the Prolog knowledge base and the user, has been implemented as shown in the sections below.

5.2 Python

Artificial Intelligence projects are different from traditional software projects. The difference lies in the technology stack, the skills required for AI-based projects, and the need for in-depth research. For implementing AI projects we need to use a programming language that is stable, flexible, and easy to integrate with other software. Python provides all of these, as it is a high-level, interpreted, and general-purpose programming language. It is easy to learn, understand and implement when compared to other programming languages. It is also dynamically typed and garbage collected.[5] Python contains so many built-in libraries for Machine Learning and Natural Language Processing. Currently, in all the IT industries, Machine Learning Engineers are mostly using Python for AI and NLP-related projects as it is very flexible and easy to implement.

5.3 Natural Language Processing

The ability of a machine to read, comprehend and derive a possible meaning from human languages is known as Natural Language Processing. NLP tends to process large volumes of textual data at ease. It also allows for Structuring a highly unstructured data source.[3] At present, there are numerous open source libraries for NLP which help us to perform operations easily. We have also leveraged the use of NLP libraries for our implementation in Python. In this project, we have used the NLTK library in python for Natural Language Processing.

5.4 Prolog

A simple logic programming language, widely used in the field of Artificial Intelligence. It was predominantly used as a declarative programming language. The core of Prolog lies in the logic that is being applied.[8] Logic is expressed as relations called Facts and Rules. It is easy to build a data representation and does not require much programming effort. In this project, we have created facts and rules for the Periodic Table. Execution of a Prolog program is initiated by the user's posting of a single goal, called the query.

Sample Prolog facts and rules are shown below.

Sample Facts for Periodic Table Group

```
period_group(1,’H’,hydrogen,1,1,cavendish).
period_group(2,’He’,helium,18,1,janssen).
period_group(3,’Li’,lithium,1,2,arfvedson).
period_group(4,’Be’,beryllium,2,2,vaulquelin).
period_group(5,’B’,boron,13,2,gay-lussac).
period_group(6,’C’,carbon,14,2,prehistoric).
period_group(7,’N’,nitrogen,15,2,rutherford).
```

Sample Facts for Physical Properties

```
physical(1,’H’,hydrogen,null,yes,0.0000899,14.175,20.28,0.79,2.2).
physical(2,'He',helium,null,yes,0.000179,null,4.22,0.49,null).
physical(3,'Li',lithium,yes,null,0.534,453.85,1615,2.1,0.98).
physical(4,'Be',beryllium,yes,null,1.85,1560.15,2742,1.4,1.57).
physical(5,'B',boron,null,null,2.34,2573.15,4200,1.2,2.04).
physical(6,'C',carbon,null,yes,2.27,3948.15,4300,0.91,2.55).
physical(7,'N',nitrogen,null,yes,0.00125,63.29,77.36,0.75,3.04).
```

Sample Facts for Magnetic Properties

```
type(1,'H',hydrogen,nonmetal).
type(2,'He',helium,noble_gas).
type(3,'Li',lithium,alkali_metal).
type(4,'Be',beryllium,alkaline_earth_metal).
type(5,'B',boron,metalloid).
type(6,'C',carbon,nonmetal).
type(7,'N',nitrogen,nonmetal).
```

Sample Facts for Element Types

```
type(1,'H',hydrogen,nonmetal).
type(2,'He',helium,noble_gas).
type(3,'Li',lithium,alkali_metal).
type(4,'Be',beryllium,alkaline_earth_metal).
type(5,'B',boron,metalloid).
type(6,'C',carbon,nonmetal).
type(7,'N',nitrogen,nonmetal).
```

Sample Facts for Element Phases

```
type(1,'H',hydrogen,nonmetal).
type(2,'He',helium,noble_gas).
type(3,'Li',lithium,alkali_metal).
type(4,'Be',beryllium,alkaline_earth_metal).
type(5,'B',boron,metalloid).
type(6,'C',carbon,nonmetal).
type(7,'N',nitrogen,nonmetal).
```

Sample Rules for Periodic Table Group

```
elements_in_a_group(Group,Symbol,Element):-period_group(_,Symbol,Element,Group,_,_).
atomic_number(Symbol,Element,Number):-period_group(Number,Symbol,Element,_,_,_).
name_of(Symbol,Element):-period_group(_,Symbol,Element,_,_,_).
discoverer(Element,Discoverer):-period_group(_,_,Element,_,_,Discoverer).
metal_or_nonmetal(Element,Y):-physical(_,_,Element,Y,_,_,_,_,_,_).
density(Element,D):-physical(_,_,Element,_,_,D,_,_,_,_).
```

5.5 Working

1. Obtain the user input as a string.

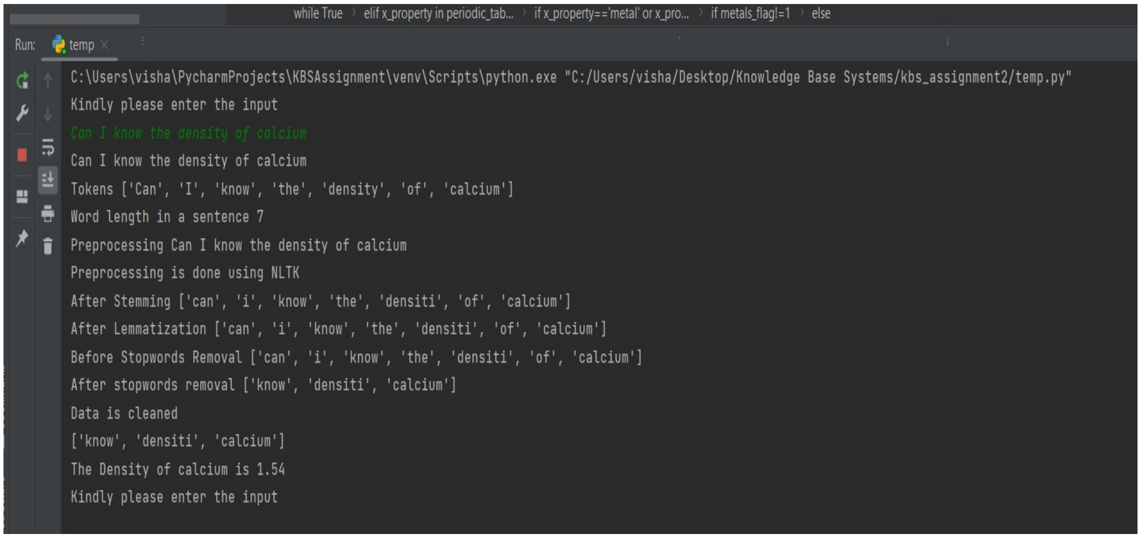
2. Once the input is read, the program uses the NLTK libraries to pre-process the text. Stemming, lemmatization, emoji conversion, and stop words removal are all included in pre-processing.

3. Cleaned, error-free data after pre-processing is received.

4. The Pyswip module in Python, which links Python with Prolog, is used to feed this data as input to Prolog.

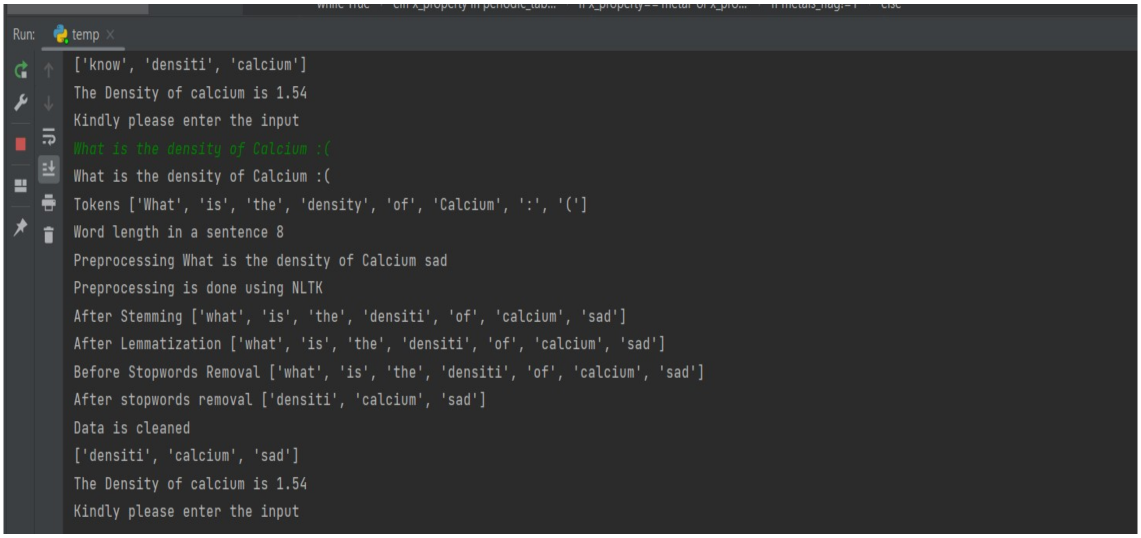
5. Based on the user's input its respective prolog rules and queries are called.

6. Prolog returns the findings.


while True
elif x_property in periodic_tab...
if x_property=='metal' or x_pro...
if metals_flag!=1
else
Run:
temp

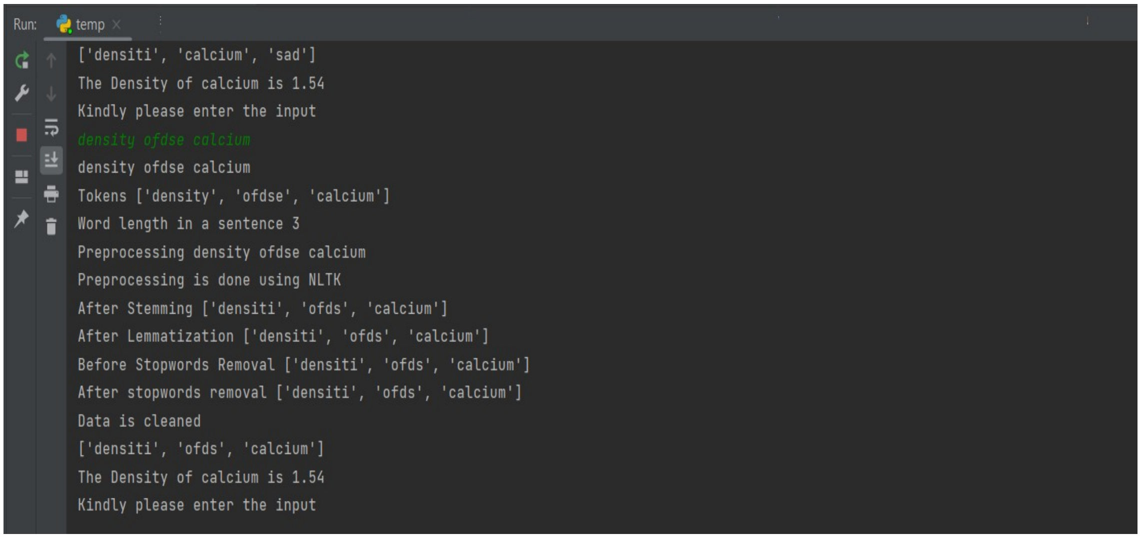
```
C:\Users\visha\PycharmProjects\KBSAssignment\venv\Scripts\python.exe "C:/Users/visha/Desktop/Knowledge Base Systems/kbs_assignment2/temp.py"
Kindly please enter the input
Can I know the density of calcium
Can I know the density of calcium
Tokens ['Can', 'I', 'know', 'the', 'density', 'of', 'calcium']
Word length in a sentence 7
Preprocessing Can I know the density of calcium
Preprocessing is done using NLTK
After Stemming ['can', 'i', 'know', 'the', 'densiti', 'of', 'calcium']
After Lemmatization ['can', 'i', 'know', 'the', 'densiti', 'of', 'calcium']
Before Stopwords Removal ['can', 'i', 'know', 'the', 'densiti', 'of', 'calcium']
After stopwords removal ['know', 'densiti', 'calcium']
Data is cleaned
['know', 'densiti', 'calcium']
The Density of calcium is 1.54
Kindly please enter the input
```

Figure 2: User input without any errors



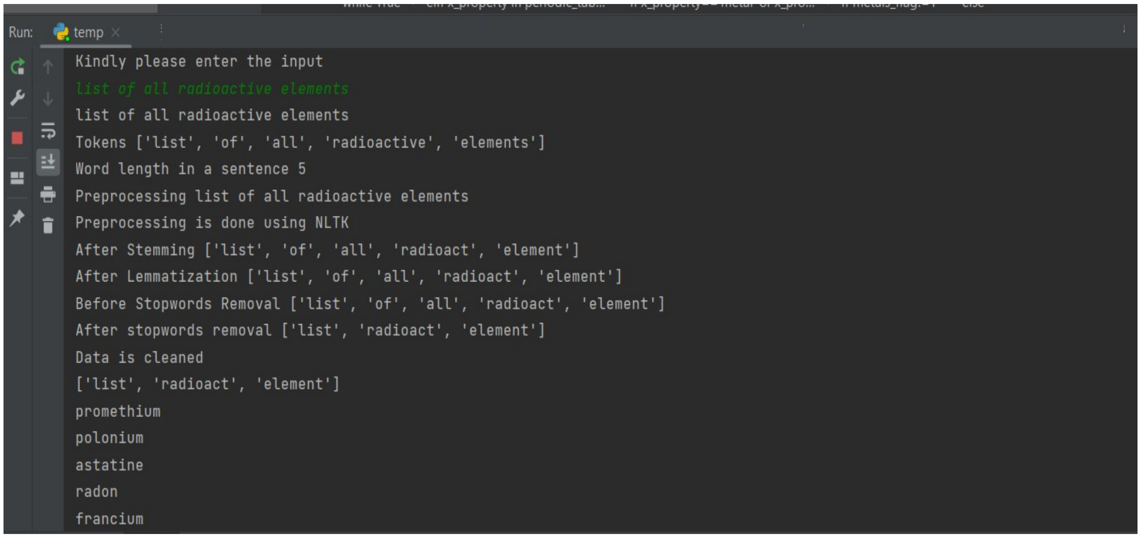
```
['know', 'densiti', 'calcium']
The Density of calcium is 1.54
Kindly please enter the input
What is the density of Calcium :(
What is the density of Calcium :(
Tokens ['What', 'is', 'the', 'density', 'of', 'Calcium', ':', '(']
Word length in a sentence 8
Preprocessing What is the density of Calcium sad
Preprocessing is done using NLTK
After Stemming ['what', 'is', 'the', 'densiti', 'of', 'calcium', 'sad']
After Lemmatization ['what', 'is', 'the', 'densiti', 'of', 'calcium', 'sad']
Before Stopwords Removal ['what', 'is', 'the', 'densiti', 'of', 'calcium', 'sad']
After stopwords removal ['densiti', 'calcium', 'sad']
Data is cleaned
['densiti', 'calcium', 'sad']
The Density of calcium is 1.54
Kindly please enter the input
```

Figure 3: User input with Emoji's



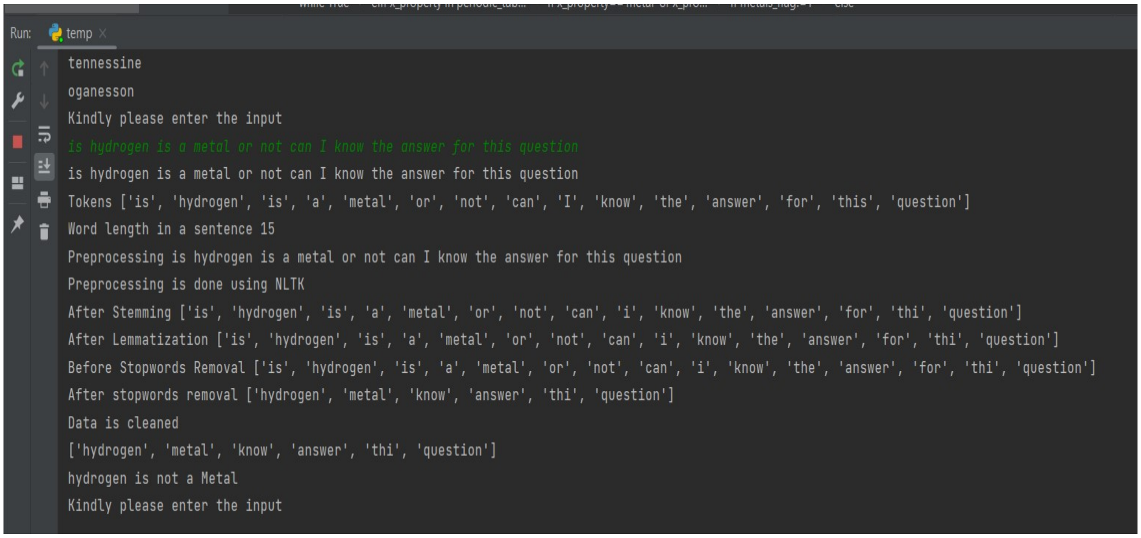
```
['densiti', 'calcium', 'sad']
The Density of calcium is 1.54
Kindly please enter the input
density ofdse calcium
density ofdse calcium
Tokens ['density', 'ofdse', 'calcium']
Word length in a sentence 3
Preprocessing density ofdse calcium
Preprocessing is done using NLTK
After Stemming ['densiti', 'ofds', 'calcium']
After Lemmatization ['densiti', 'ofds', 'calcium']
Before Stopwords Removal ['densiti', 'ofds', 'calcium']
After stopwords removal ['densiti', 'ofds', 'calcium']
Data is cleaned
['densiti', 'ofds', 'calcium']
The Density of calcium is 1.54
Kindly please enter the input
```

Figure 4: User input with some errors



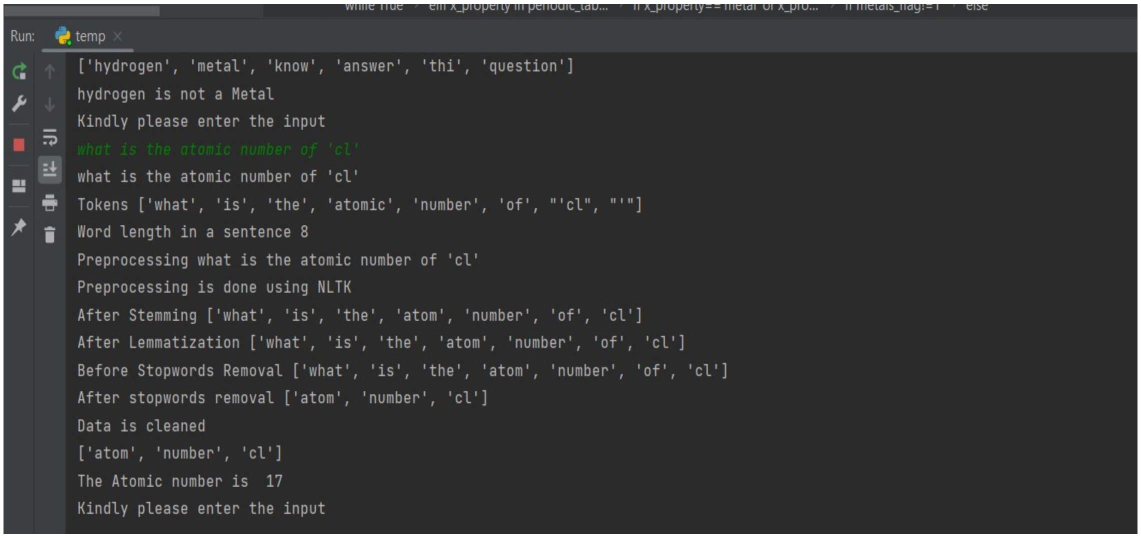
```
Kindly please enter the input
list of all radioactive elements
list of all radioactive elements
Tokens ['list', 'of', 'all', 'radioactive', 'elements']
Word length in a sentence 5
Preprocessing list of all radioactive elements
Preprocessing is done using NLTK
After Stemming ['list', 'of', 'all', 'radioact', 'element']
After Lemmatization ['list', 'of', 'all', 'radioact', 'element']
Before Stopwords Removal ['list', 'of', 'all', 'radioact', 'element']
After stopwords removal ['list', 'radioact', 'element']
Data is cleaned
['list', 'radioact', 'element']
promethium
polonium
astatine
radon
francium
```

Figure 5: Different type of Question and its response


while True > elif x_property in periodic_tab... > if x_property==metal or x_pro... > if metals_flag!=1 > else
Run: temp

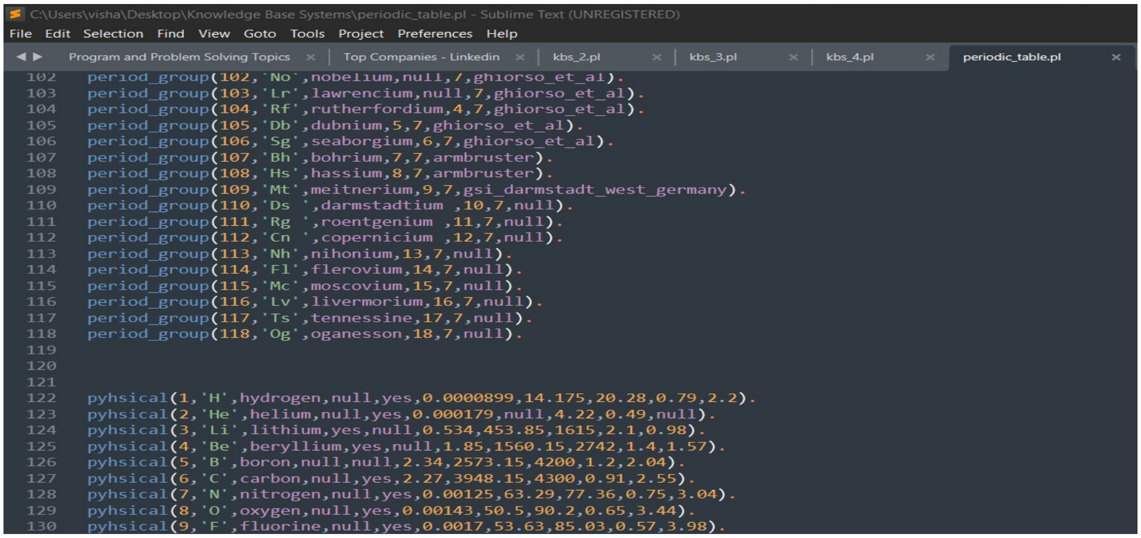
```
tennessine
oganesson
Kindly please enter the input
is hydrogen is a metal or not can I know the answer for this question
is hydrogen is a metal or not can I know the answer for this question
Tokens ['is', 'hydrogen', 'is', 'a', 'metal', 'or', 'not', 'can', 'I', 'know', 'the', 'answer', 'for', 'this', 'question']
Word length in a sentence 15
Preprocessing is hydrogen is a metal or not can I know the answer for this question
Preprocessing is done using NLTK
After Stemming ['is', 'hydrogen', 'is', 'a', 'metal', 'or', 'not', 'can', 'i', 'know', 'the', 'answer', 'for', 'thi', 'question']
After Lemmatization ['is', 'hydrogen', 'is', 'a', 'metal', 'or', 'not', 'can', 'i', 'know', 'the', 'answer', 'for', 'thi', 'question']
Before Stopwords Removal ['is', 'hydrogen', 'is', 'a', 'metal', 'or', 'not', 'can', 'i', 'know', 'the', 'answer', 'for', 'thi', 'question']
After stopwords removal ['hydrogen', 'metal', 'know', 'answer', 'thi', 'question']
Data is cleaned
['hydrogen', 'metal', 'know', 'answer', 'thi', 'question']
hydrogen is not a Metal
Kindly please enter the input
```

Figure 6: Ground queries


while True > elif x_property in periodic_tab... > if x_property==metal or x_pro... > if metals_flag!=1 > else
Run: temp

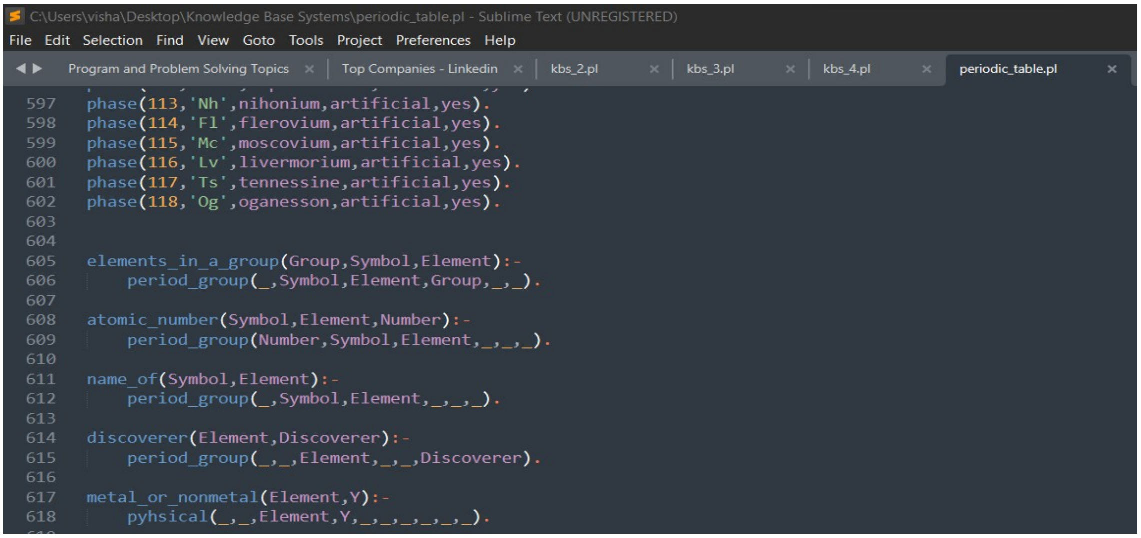
```
['hydrogen', 'metal', 'know', 'answer', 'thi', 'question']
hydrogen is not a Metal
Kindly please enter the input
what is the atomic number of 'cl'
what is the atomic number of 'cl'
Tokens ['what', 'is', 'the', 'atomic', 'number', 'of', "'cl", "'"]
Word length in a sentence 8
Preprocessing what is the atomic number of 'cl'
Preprocessing is done using NLTK
After Stemming ['what', 'is', 'the', 'atom', 'number', 'of', 'cl']
After Lemmatization ['what', 'is', 'the', 'atom', 'number', 'of', 'cl']
Before Stopwords Removal ['what', 'is', 'the', 'atom', 'number', 'of', 'cl']
After stopwords removal ['atom', 'number', 'cl']
Data is cleaned
['atom', 'number', 'cl']
The Atomic number is  17
Kindly please enter the input
```

Figure 7: Finding Atomic number of an Element


C:\Users\visha\Desktop\Knowledge Base Systems\periodic_table.pl - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Program and Problem Solving Topics
Top Companies - Linkedin
kbs_2.pl
kbs_3.pl
kbs_4.pl
periodic_table.pl

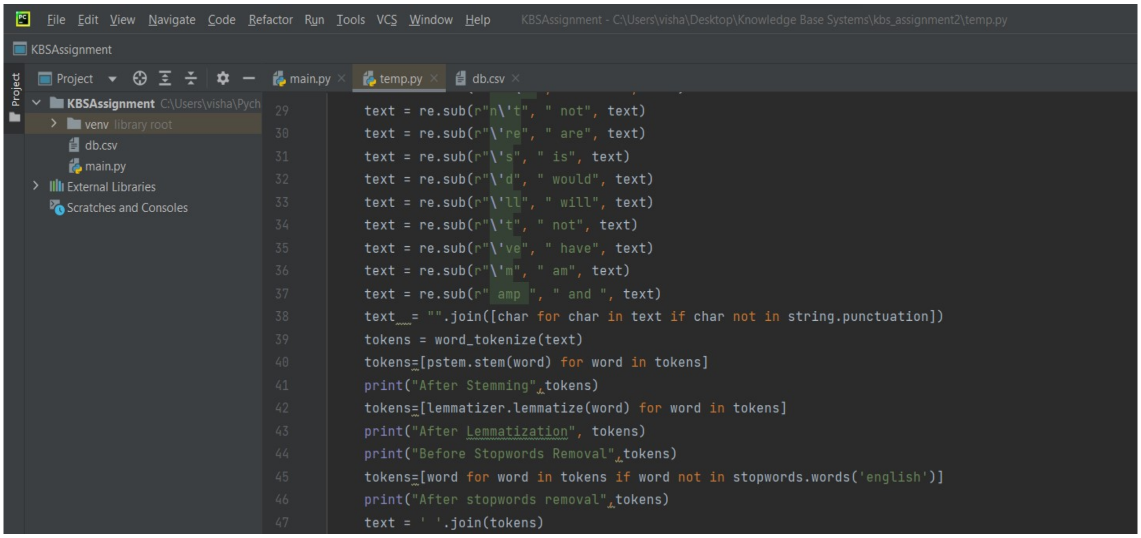
```
period_group(102,'No',nobelium,null,7,ghiorso_et_al).
period_group(103,'Lr',lawrencium,null,7,ghiorso_et_al).
period_group(104,'Rf',rutherfordium,4,7,ghiorso_et_al).
period_group(105,'Db',dubnium,5,7,ghiorso_et_al).
period_group(106,'Sg',seaborgium,6,7,ghiorso_et_al).
period_group(107,'Bh',bohrium,7,7,armbruster).
period_group(108,'Hs',hassium,8,7,armbruster).
period_group(109,'Mt',meitnerium,9,7,gsi_darmstadt_west_germany).
period_group(110,'Ds ',darmstadtium ,10,7,null).
period_group(111,'Rg ',roentgenium ,11,7,null).
period_group(112,'Cn ',copernicium ,12,7,null).
period_group(113,'Nh',nihonium,13,7,null).
period_group(114,'Fl',flerovium,14,7,null).
period_group(115,'Mc',moscovium,15,7,null).
period_group(116,'Lv',livermorium,16,7,null).
period_group(117,'Ts',tennessine,17,7,null).
period_group(118,'Og',oganesson,18,7,null).



pyhsical(1,'H',hydrogen,null,yes,0.0000899,14.175,20.28,0.79,2.2).
pyhsical(2,'He',helium,null,yes,0.000179,null,4.22,0.49,null).
pyhsical(3,'Li',lithium,yes,null,0.534,453.85,1615,2.1,0.98).
pyhsical(4,'Be',beryllium,yes,null,1.85,1560.15,2742,1.4,1.57).
pyhsical(5,'B',boron,null,null,2.34,2573.15,4200,1.2,2.04).
pyhsical(6,'C',carbon,null,yes,2.27,3948.15,4300,0.91,2.55).
pyhsical(7,'N',nitrogen,null,yes,0.00125,63.29,77.36,0.75,3.04).
pyhsical(8,'O',oxygen,null,yes,0.00143,50.5,90.2,0.65,3.44).
pyhsical(9,'F',fluorine,null,yes,0.0017,53.63,85.03,0.57,3.98).
```

Figure 8: Prolog Rules for Periodic Table


C:\Users\visha\Desktop\Knowledge Base Systems\periodic_table.pl - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Program and Problem Solving Topics
Top Companies - Linkedin
kbs_2.pl
kbs_3.pl
kbs_4.pl
periodic_table.pl

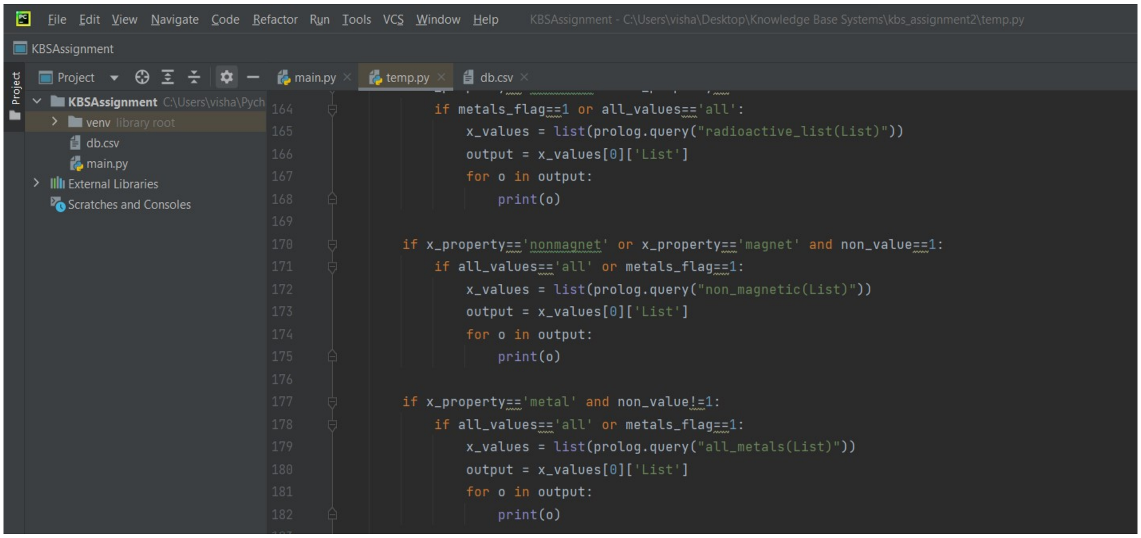
```
phase(113,'Nh',nihonium,artificial,yes).
phase(114,'Fl',flerovium,artificial,yes).
phase(115,'Mc',moscovium,artificial,yes).
phase(116,'Lv',livermorium,artificial,yes).
phase(117,'Ts',tennessine,artificial,yes).
phase(118,'Og',oganesson,artificial,yes).


elements_in_a_group(Group,Symbol,Element):-
    period_group(_,Symbol,Element,Group,_,_).

atomic_number(Symbol,Element,Number):-
    period_group(Number,Symbol,Element,_,_,_).

name_of(Symbol,Element):-
    period_group(_,Symbol,Element,_,_,_).

discoverer(Element,Discoverer):-
    period_group(_,_,Element,_,_,Discoverer).

metal_or_nonmetal(Element,Y):-
    pyhsical(_,_,Element,Y,_,_,_,_,_,_).
```

Figure 9: Prolog Facts for Periodic Table


File Edit View Navigate Code Refactor Run Tools VCS Window Help
KBSAssignment - C:\Users\visha\Desktop\Knowledge Base Systems\kbs_assignment2\temp.py
KBSAssignment
Project
main.py
temp.py
db.csv
KBSAssignment C:\Users\visha\Pych
venv library root
db.csv
main.py
External Libraries
Scratches and Consoles

```
    text = re.sub(r"n\'t", " not", text)
    text = re.sub(r"\'re", " are", text)
    text = re.sub(r"\'s", " is", text)
    text = re.sub(r"\'d", " would", text)
    text = re.sub(r"\'ll", " will", text)
    text = re.sub(r"\'t", " not", text)
    text = re.sub(r"\'ve", " have", text)
    text = re.sub(r"\'m", " am", text)
    text = re.sub(r" amp ", " and ", text)
    text = "".join([char for char in text if char not in string.punctuation])
    tokens = word_tokenize(text)
    tokens=[pstem.stem(word) for word in tokens]
    print("After Stemming",tokens)
    tokens=[lemmatizer.lemmatize(word) for word in tokens]
    print("After Lemmatization", tokens)
    print("Before Stopwords Removal",tokens)
    tokens=[word for word in tokens if word not in stopwords.words('english')]
    print("After stopwords removal",tokens)
    text = ' '.join(tokens)
```

Figure 10: NLP Preprocessing using NLTK


File Edit View Navigate Code Refactor Run Tools VCS Window Help
KBSAssignment - C:\Users\visha\Desktop\Knowledge Base Systems\kbs_assignment2\temp.py
KBSAssignment
Project
main.py
temp.py
db.csv
KBSAssignment C:\Users\visha\Pych
venv library root
db.csv
main.py
External Libraries
Scratches and Consoles

```
        if metals_flag==1 or all_values=='all':
            x_values = list(prolog.query("radioactive_list(List)"))
            output = x_values[0]['List']
            for o in output:
                print(o)

    if x_property=='nonmagnet' or x_property=='magnet' and non_value==1:
        if all_values=='all' or metals_flag==1:
            x_values = list(prolog.query("non_magnetic(List)"))
            output = x_values[0]['List']
            for o in output:
                print(o)

    if x_property=='metal' and non_value!=1:
        if all_values=='all' or metals_flag==1:
            x_values = list(prolog.query("all_metals(List)"))
            output = x_values[0]['List']
            for o in output:
                print(o)
```

Figure 11: Connecting Prolog and Python using Pyswip

6 Research Team

The roles and responsibilities of each team member have been displayed below.

Ananya Varsha
Researched and web-scrapped periodic table data. Created entire knowledge base for the periodic table in Prolog. Assisted in the creation of code base for NLP in Python

Saghana Mahesh Sarma
Researched on NLP and it's utilization on Python. Created a knowledge transfer article for the team on NLP. Assisted in the creation of Prolog Knowledge-base.

Sivakumaran Malli Janardhanan
Web-scrapped entire periodic table data. Created all rules, procedures, queries for Prolog Knowledge base. Assisted in the integration of Python with Prolog.

Vishanth Surresh
Integrated Python with Prolog. Developed code base for running Prolog code in python. Applied several NLP rules using pre-defined libraries.

7 Conclusion

In this course, we learnt how to work with Prolog and create numerous real-time knowledge base instances, how to create facts and rules, and accessing databases using Prolog. As Data Engineers/Scientists we wanted to connect Prolog with the latest technologies such as Artificial Intelligence and Machine Learning. So, integration between the logical programming of Prolog and the most widely utilized scripting language Python has been performed. It provided us with a wide array of learning experiences, both with Prolog and NLP.

8 Future Work

As per the current implementation, we have utilised dictionaries to identify the elements in the periodic table. For future work. we can train an NLP model to automatically detect the names of elements as soon as the user types. This reduces the threshold of storing additional hard-coded data and leverages the use of more powerful NLP concepts.

References

[1] First-Order Logic — Baeldung on Computer Science — baeldung.com. https://www.baeldung.com/cs/first-order-logic. [Accessed 11-Aug-2022].

[2] Abhijeet Gupta. Complex aggregates in natural language interface to databases. *International Institute of Information Technology, Hyderabad*, 2013.

[3] What is Natural Language Processing? — ibm.com. https://www.ibm.com/cloud/learn/natural-language-processing. [Accessed 11-Aug-2022].

[4] https://pypi.org/project/pyswip/.

[5] Welcome to Python.org — python.org. https://www.python.org/about/. [Accessed 11-Aug-2022].

[6] Mukesh Kumar Rohil, Rohan Kumar Rohil, Divyesakshi Rohil, and Anurag Runthala. Natural language interfaces to domain specific knowledge bases: an illustration for querying elements of the periodic table. In *2018 IEEE 17th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, pages 517–523. IEEE, 2018.

[7] Vidya Setlur, Sarah E Battersby, Melanie Tory, Rich Gossweiler, and Angel X Chang. Eviza: A natural language interface for visual analysis. In *Proceedings of the 29th annual symposium on user interface software and technology*, pages 365–377, 2016.

[8] manual — swi-prolog.org. https://www.swi-prolog.org/pldoc/doc_for?object=manual. [Accessed 11-Aug-2022].