Project Report

November 23, 2021

0.0.1 Introduction

In this project, I implemented an NLP-based interface for the Elective-advisory system I build using prolog. It uses python modules to interact with the prolog file and get the desired output. I used the nltk library and its various modules for the working of the code. The program takes input from user in English language and process it to pass input for querying the prolog file. The prolog file provides the necessary ouput in form of a generator and conversion is required to get it in required form. In this way, I implemented the NLP based interface for interacting with prolog database.

0.0.2 Code

The code of the program is in a file 'nl.py' attached along with this file. To run the program, open a command prompt from the same directory as the program file. Run the command "python3 nl.py" and press enter.
The code will become more clear with the below explanations.

In the below code, I have imported the 'nltk' library and some other required modules like word_tokenize for tokenizing and stopwords to remove unncessary words from tokenized array of words. I have imported prolog module from pyswip library to interact with the prolog file and running queries from here.

[1]:

```
import nltk
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
from nltk.corpus import stopwords
from pyswip import Prolog
```

Here, I have created an object 'swipl' of Prolog. It will help in consulting the 'new.pl' file and running queries later. In the second line, I have consulted the required file. There will be some warnings associated with the prolog file. It won't have any impact on the working of the program and we can ignore them.

[2]:

```
swipl = Prolog()
swipl.consult("new.pl")
```

```
Warning: /home/shivansh/new.pl:204:
Warning:    Singleton variables: [Batch,Y]
Warning: /home/shivansh/new.pl:234:
Warning:    Clauses of start/2 are not together in the source-file
```

```
Warning:    Earlier definition at /home/shivansh/new.pl:224
Warning:    Current predicate: print/2
Warning:    Use :- discontiguous start/2. to suppress this message
Warning: /home/shivansh/new.pl:242:
Warning:    Singleton variables: [Y,List]
Warning: /home/shivansh/new.pl:259:
Warning:    Singleton variables: [H]
Warning: /home/shivansh/new.pl:263:
Warning:    Singleton variables: [Sem,X]
Warning: /home/shivansh/new.pl:263:
Warning:    Clauses of helper/3 are not together in the source-file
Warning:    Earlier definition at /home/shivansh/new.pl:246
Warning:    Current predicate: adder/5
Warning:    Use :- discontiguous helper/3. to suppress this message
Warning: /home/shivansh/new.pl:280:
Warning:    Singleton variables: [H]
Warning: /home/shivansh/new.pl:283:
Warning:    Clauses of preChecker/2 are not together in the source-file
Warning:    Earlier definition at /home/shivansh/new.pl:267
Warning:    Current predicate: adder2/4
Warning:    Use :- discontiguous preChecker/2. to suppress this message
```

In the next part, it starts with a print statement to make the code more interactive. Here, I will take the input from the user regarding the semester for which the user wants suggestions regarding the electives. I remove the stopwords to make the code more efficient. The filtered array contains the final words left after removing the stopwords.

[3]:

```python
print('Hello! Welcome to the NLP interface of course advisory system.')
sentence = input('Please enter the semester for course suggestion\n')
words= set(stopwords.words("english"))
tokens = word_tokenize(sentence)
filtered=[]
for i in tokens:
    if i not in words:
        filtered.append(i)
filtered
```

```
Hello! Welcome to the NLP interface of course advisory system.

Please enter the semester for course suggestion
 I want suggestions for the fifth semester.
```

[3]:

```
['I', 'want', 'suggestions', 'fifth', 'semester', '.']
```

In this part, I passed the array to query to find the valid semester number to fetch further information. I check for all the elements present in the array and it will return true only if it finds a hit with the rules present in the rulebase. Then, I take the semester number to fetch further information regarding domains and courses.

[4]:

```
res=[]
sem=[]
for I in filtered:
    I = I.lower()
    if bool(swipl.query(f"num({I},What)")):
        if len(list(swipl.query(f"num({I},What)")))>0:
            res.append(list(swipl.query(f"num({I},What)")))

for i in res:
    sem.append(i[0]['What'])

sem
number = sem[0]
```

In this part, I will use the semester number fetched in previous part to fetch the domains available for course selections from the prolog database. It will append them in the res array.

[5]:

```
res=[]
courses=[]
for i in sem:
    if bool(swipl.query(f"course(_,_,_,_,{i},What)")):
        res.append(list(swipl.query(f"course(_,_,_,_,{i},What)")))
```

The informtaion fetched in the previous are in the form of dicts stored in an array. In this part, the name of the domain will be extracted from the dict and stored in an array. The next statement will then print the elements stored in the array for selection by the user.

[6]:

```
temp = res[0]
temp
interest=set()
for i in temp:
    interest.add(i['What'])
print('Courses with these domains are available in the selected semester/s:-')
for i in interest:
    print(i)
```

```
Courses with these domains are available in the selected semester/s:-
electronics
computer_security
image_processing
cryptography
theorotical_computer_science
mathematics
psychology
biology
reinforcement_learning
social_science
```

```
number_theory
blockchain
natural_language_processing
data_mining
economics
design
artificial_intelligence
machine_learning
sociology
```

As you can see, a list of domains available is printed. The user now have to provide an input which must contain the name of the domain to be selected. This sentence will go through a similar process as the first input by the user. It will also provide a final list after removal of stopwords.

[7]:

```
selection=input("What are your interests for selection of the courses?\n")
tokens1 = word_tokenize(selection)
filtered1=[]
for i in tokens1:
    if i not in words:
        filtered1.append(i)
filtered1
```

```
What are your interests for selection of the courses?
 I am interested in artificial_intelligence, mathematics and economics.
```

[7]:

```
['I',
 'interested',
 'artificial_intelligence',
 ',',
 'mathematics',
 'economics',
 '.']
```

Here, I will get the list of courses by querying the prolog database with semester number and the chosen domains. It will get the list of all the courses which will satisfy the criteria and store it in an array mid.

[8]:

```
final = []
mid = []
for i in filtered1:
    i = i.lower()
    if bool(swipl.query(f"course(_,Name,_,_,{number},{i})")):
        if len(list(swipl.query(f"course(_,Name,_,_,{number},{i})")))>0:
            mid.append(list(swipl.query(f"course(_,Name,_,_,{number},{i})")))
```

This part will do the same work I did earlier after getting the list of dicts in case of domains. Here, also the previous part has provided the list of courses in the same format. It will remove unncessary parts in the final response and provide us with an array only containing the list of courses.

[9]:

```
for i in mid:
    for j in range(0,len(i)):
        final.append(i[j]['Name'])
```

This will mark the end of the program as it will print the array containing the final list of courses obtained from the last part of the code. A list of available courses will be printed and the program will be terminated.

[10]:

```
print('You can opt for following courses:-')
serial =1
for i in final:
    print(serial,end=". ")
    print(i)
    serial+=1
```

```
You can opt for following courses:-
1. Artificial Intelligence
2. Complex Analysis
3. Microeconomics
4. Game Theory
```