Natural Language Processing Assignment 3

Intent

As a part of this assignment , I have implemented the first two required tasks and a couple of other extra tasks . I got to implement the CYK parser without the use of probabilities . I got to read more about grammar and text parsing during the course of the assignment . I have attempted the recognizer and also tried adding imperatives and yes-no questions to the grammar

Required Task 1

Implementation of the CYK parser :

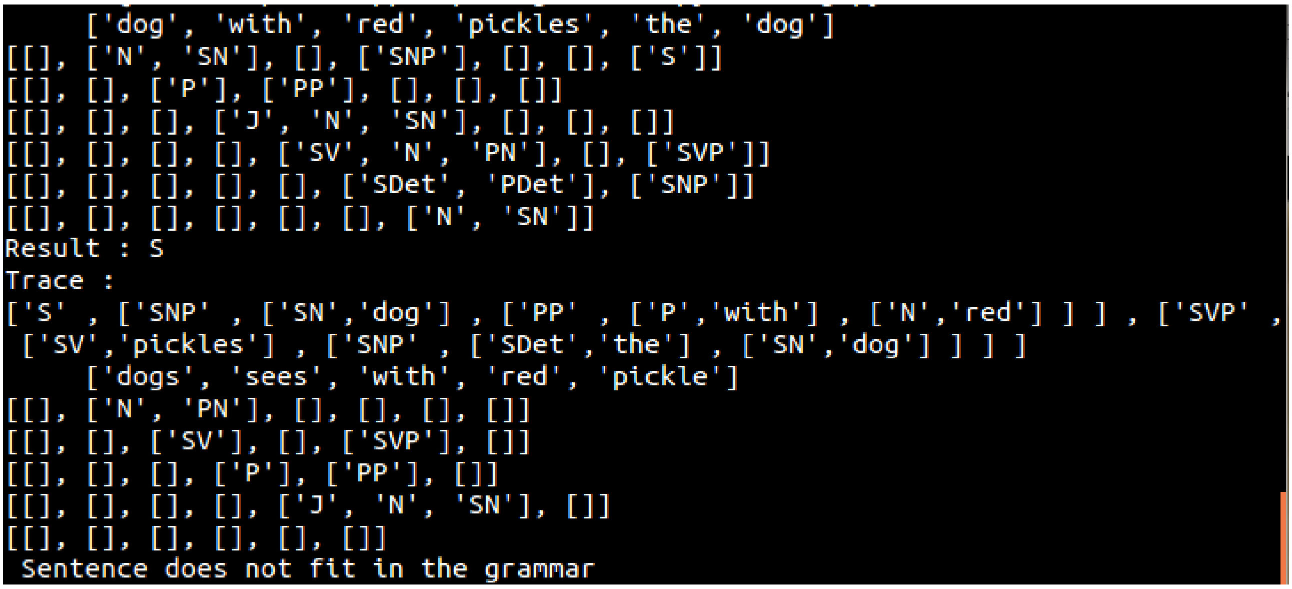
I have implemented the CYK parser without the use of probabilities . A matrix is built and the part of speech of each word pair is obtained from the grammar.

Grammer :

```
S = [['SNP','SVP'],['PNP','PVP']],
SNP = [['SDet','SN'],['SN','PP']],
PNP = [['PDet','PN'],['PN','PP']],
SVP = [['SV','SNP'],['SV','PNP'],['SV','PP']],
PVP = [['PV','SNP'],['PV','PNP'],['PV','PP']],
PP = [['P', 'N']],
SDet = ['the', 'a'],
PDet = ['the'],
P = ['with'],
J = ['red', 'big'],
N = ['red','dog','ball','light','dogs','pickles'],
SN = ['red','dog','ball','light'], # singular nouns
PN = ['dogs','pickles'], # plural nouns
SV = ['sees','pickles'], # present tense singular verbs
PV = ['see','light'], # present tense plural verbs
V = ['liked','slept'] # past tense verbs
```

The grammar is the extended version of the given grammar to support singular and plural verbs and nouns . The grammar has been explained as a part of the next exercise

Output :



```
     ['dog', 'with', 'red', 'pickles', 'the', 'dog']
[[], ['N', 'SN'], [], ['SNP'], [], [], ['S']]
[[], [], ['P'], ['PP'], [], [], []]
[[], [], [], ['J', 'N', 'SN'], [], [], []]
[[], [], [], [], ['SV', 'N', 'PN'], [], ['SVP']]
[[], [], [], [], [], ['SDet', 'PDet'], ['SNP']]
[[], [], [], [], [], [], ['N', 'SN']]
Result : S
Trace :
['S' , ['SNP' , ['SN','dog'] , ['PP' , ['P','with'] , ['N','red'] ] ] , ['SVP' ,
 ['SV','pickles'] , ['SNP' , ['SDet','the'] , ['SN','dog'] ] ] ]
     ['dogs', 'sees', 'with', 'red', 'pickle']
[[], ['N', 'PN'], [], [], [], []]
[[], [], ['SV'], [], ['SVP'], []]
[[], [], [], ['P'], ['PP'], []]
[[], [], [], [], ['J', 'N', 'SN'], []]
[[], [], [], [], [], []]
 Sentence does not fit in the grammar
```

The output shows that the sentence has been parsed and if it obeys the rules of the grammar , a true is returned and a false is returned if the sentence cannot be generated by the grammar

Required Task 2

This exercise required an extension of the grammar to generate valid English sentences .

Some rules I have added include :

1. Singular Subjects being preceded by a singular determiner and the plural Subjects are preceded by plural determiners
2. In the present tense , the Singular subjects are followed by singular verbs and a similar rule for the plural case

I have modified my grammar to make sure these rules hold .

The newly included non terminals are

SNP : singular noun phrase ,SVP : singular verb phrase ,PNP : plural noun phrase ,PVP : plural verb phrase ,SV : singular verb ,PV : plural verb ,SN : singular noun ,PN : plural noun

Output :

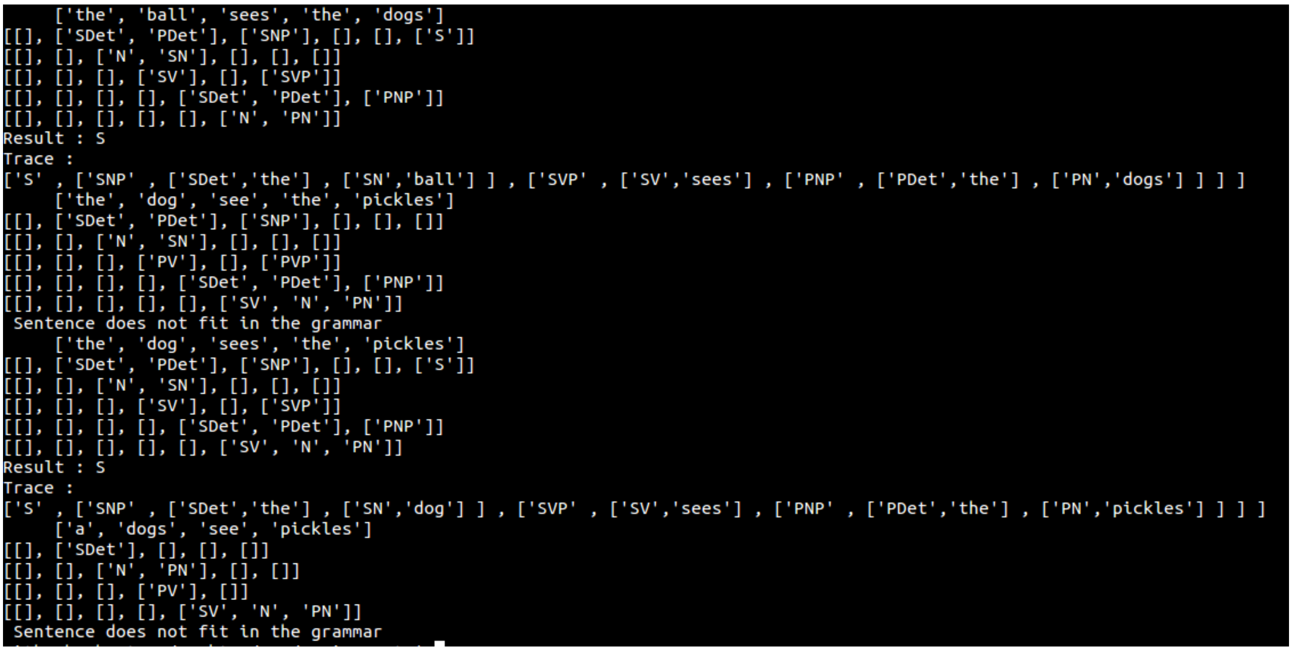
For sentences

parse("the ball sees the dogs".split()) : true

parse("the dog see the pickles".split()) : false (dog followed by see)

parse("the dog sees the pickles".split()) :true

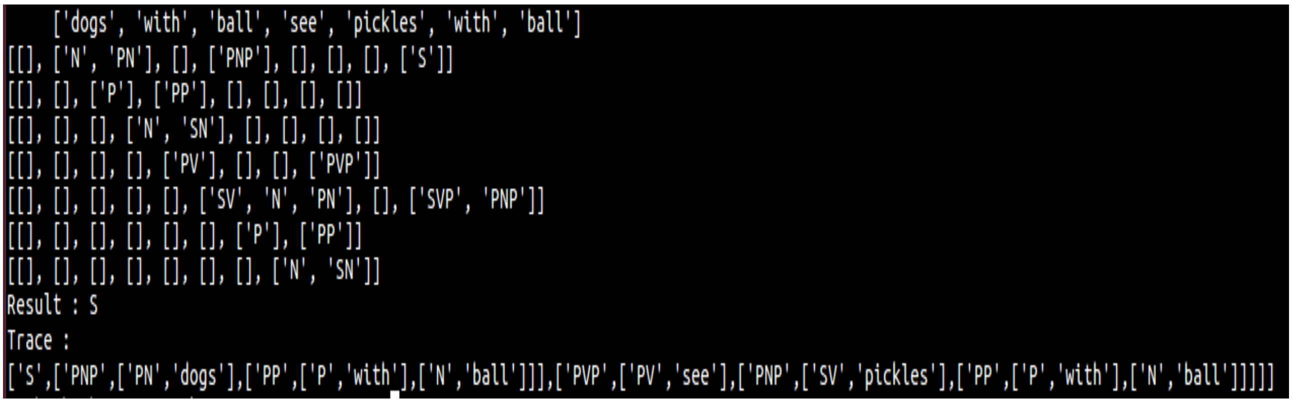
parse("a dogs see pickles".split()) : false (a followed by dog)



```
    ['the', 'ball', 'sees', 'the', 'dogs']
[[], ['SDet', 'PDet'], ['SNP'], [], [], ['S']]
[[], [], ['N', 'SN'], [], [], []]
[[], [], [], ['SV'], [], ['SVP']]
[[], [], [], [], ['SDet', 'PDet'], ['PNP']]
[[], [], [], [], [], ['N', 'PN']]
Result : S
Trace :
['S' , ['SNP' , ['SDet','the'] , ['SN','ball'] ] , ['SVP' , ['SV','sees'] , ['PNP' , ['PDet','the'] , ['PN','dogs'] ] ] ]
    ['the', 'dog', 'see', 'the', 'pickles']
[[], ['SDet', 'PDet'], ['SNP'], [], [], []]
[[], [], ['N', 'SN'], [], [], []]
[[], [], [], ['PV'], [], ['PVP']]
[[], [], [], [], ['SDet', 'PDet'], ['PNP']]
[[], [], [], [], [], ['SV', 'N', 'PN']]
 Sentence does not fit in the grammar
    ['the', 'dog', 'sees', 'the', 'pickles']
[[], ['SDet', 'PDet'], ['SNP'], [], [], ['S']]
[[], [], ['N', 'SN'], [], [], []]
[[], [], [], ['SV'], [], ['SVP']]
[[], [], [], [], ['SDet', 'PDet'], ['PNP']]
[[], [], [], [], [], ['SV', 'N', 'PN']]
Result : S
Trace :
['S' , ['SNP' , ['SDet','the'] , ['SN','dog'] ] , ['SVP' , ['SV','sees'] , ['PNP' , ['PDet','the'] , ['PN','pickles'] ] ] ]
    ['a', 'dogs', 'see', 'pickles']
[[], ['SDet'], [], [], []]
[[], [], ['N', 'PN'], [], []]
[[], [], [], ['PV'], []]
[[], [], [], [], ['SV', 'N', 'PN']]
 Sentence does not fit in the grammar
```

Extra Task 1

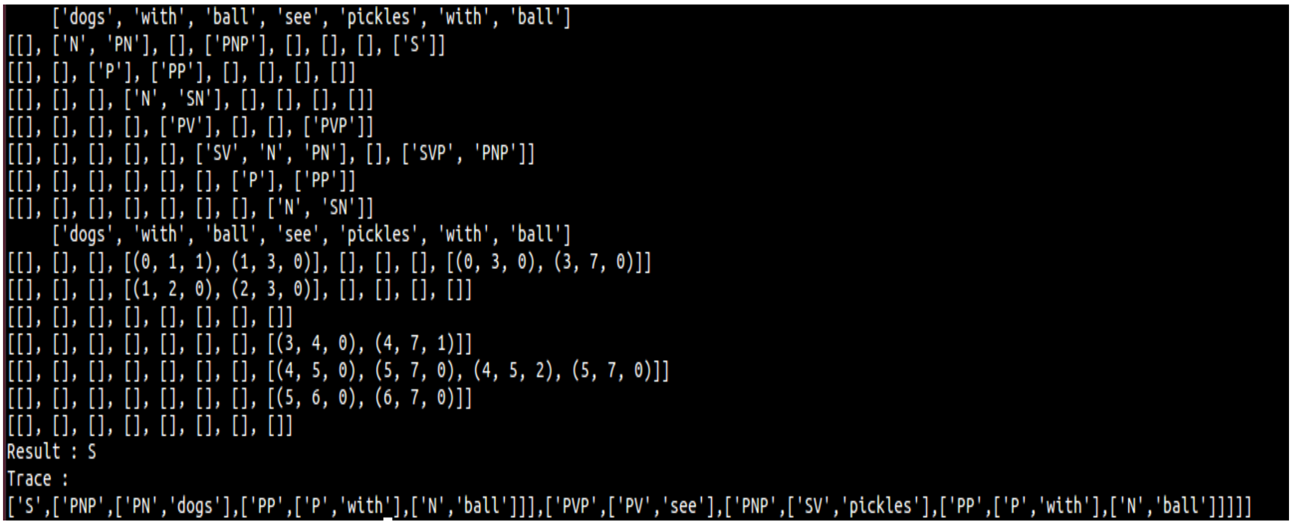
I implemented a recognizer that returns the list of all the parse trees in the same format as the generate tree function



```
    ['dogs', 'with', 'ball', 'see', 'pickles', 'with', 'ball']
[[], ['N', 'PN'], [], ['PNP'], [], [], [], ['S']]
[[], [], ['P'], ['PP'], [], [], [], []]
[[], [], [], ['N', 'SN'], [], [], [], []]
[[], [], [], [], ['PV'], [], [], ['PVP']]
[[], [], [], [], [], ['SV', 'N', 'PN'], [], ['SVP', 'PNP']]
[[], [], [], [], [], [], ['P'], ['PP']]
[[], [], [], [], [], [], [], ['N', 'SN']]
Result : S
Trace :
['S',['PNP',['PN','dogs'],['PP',['P','with'],['N','ball']]],['PVP',['PV','see'],['PNP',['SV','pickles'],['PP',['P','with'],['N','ball']]]]]
```

IMPLEMENTATION DETAIL

I have implemented this using an additional data structure (2d array of tuples) called the trace . This data structure stores the path involved in getting to the given node



```
    ['dogs', 'with', 'ball', 'see', 'pickles', 'with', 'ball']
[[], ['N', 'PN'], [], ['PNP'], [], [], [], ['S']]
[[], [], ['P'], ['PP'], [], [], [], []]
[[], [], [], ['N', 'SN'], [], [], [], []]
[[], [], [], [], ['PV'], [], [], ['PVP']]
[[], [], [], [], [], ['SV', 'N', 'PN'], [], ['SVP', 'PNP']]
[[], [], [], [], [], [], ['P'], ['PP']]
[[], [], [], [], [], [], [], ['N', 'SN']]
    ['dogs', 'with', 'ball', 'see', 'pickles', 'with', 'ball']
[[], [], [], [(0, 1, 1), (1, 3, 0)], [], [], [], [(0, 3, 0), (3, 7, 0)]]
[[], [], [], [(1, 2, 0), (2, 3, 0)], [], [], [], []]
[[], [], [], [], [], [], [], []]
[[], [], [], [], [], [], [], [(3, 4, 0), (4, 7, 1)]]
[[], [], [], [], [], [], [], [(4, 5, 0), (5, 7, 0), (4, 5, 2), (5, 7, 0)]]
[[], [], [], [], [], [], [], [(5, 6, 0), (6, 7, 0)]]
[[], [], [], [], [], [], [], []]
Result : S
Trace :
['S',['PNP',['PN','dogs'],['PP',['P','with'],['N','ball']]],['PVP',['PV','see'],['PNP',['SV','pickles'],['PP',['P','with'],['N','ball']]]]]
```

Here the the entry at 'S” position is (0,3,0) and (3,7,0) which means it was obtained using these two non terminals (ie) PNP and PVP .

0,3,0 means the first element in table[0][3]

The base cases (terminals) are handled by making sure that when x +1 = y ((0,1),(1,2)) by returning the rule non terminal returns terminal

Extra task 2

I tried extending the base grammar to include **adjectives , imperatives and yes-no questions** . I did this on the grammar that does not take care of the singular/plural nouns and other grammatical traits of the English language

Heres my extended grammar

```
Q = [['AUX','NP','VP']],
I = [['V','NP'],['V','PP']],# An imperative begins with a verb phrase and has no subject
S = [['NP','VP']],
NP = [['Det', 'N'], ['N', 'PP'],['Det','JP']],
VP = [['V', 'NP'],['V','CNP'], ['V', 'PP']],
PP = [['P', 'N'],['P','JP']],
Det = ['the', 'a'],
P = ['with'],
JP = [['J','N']],# Adding adjectives
J = ['red', 'big'],
N = ['red', 'dog', 'dogs', 'pickles', 'ball', 'light'],
V = ['pickles', 'see', 'sees', 'liked', 'light', 'slept'],
AUX = [ 'Do','Does','Can'] #Adding yes no questions
```

Some Examples

A sentence with an adjective used :

```
['S', ['NP', ['Det', 'a'], ['JP', ['J', 'red'], ['N', 'dog']]], ['VP', ['V', 'sees'], ['NP', ['Det', 'the'], ['N', 'light']]]]
```

Sentence : **a red dog sees the light**

A question :

```
['Q', ['AUX', 'Do'], ['NP', ['Det', 'the'], ['N', 'dog']], ['VP', ['V', 'slept'], ['PP', ['P', 'with'], ['N', 'ball']]]]
```

Question : **Do the dog slept with ball**

An imperative:

```
['I', ['V', 'see'], ['NP', ['Det', 'the'], ['JP', ['J', 'red'], ['N', 'red']]]]
```

Question : **see the red red**

As an extension , I am also trying to add it to the grammar that supports proper English grammar , but due to a few conflicts I am not able to get it done within the stipulated time .