Terna Engineering College
Computer Engineering Department
Program: Sem VIII

Course: Natural Language Processing

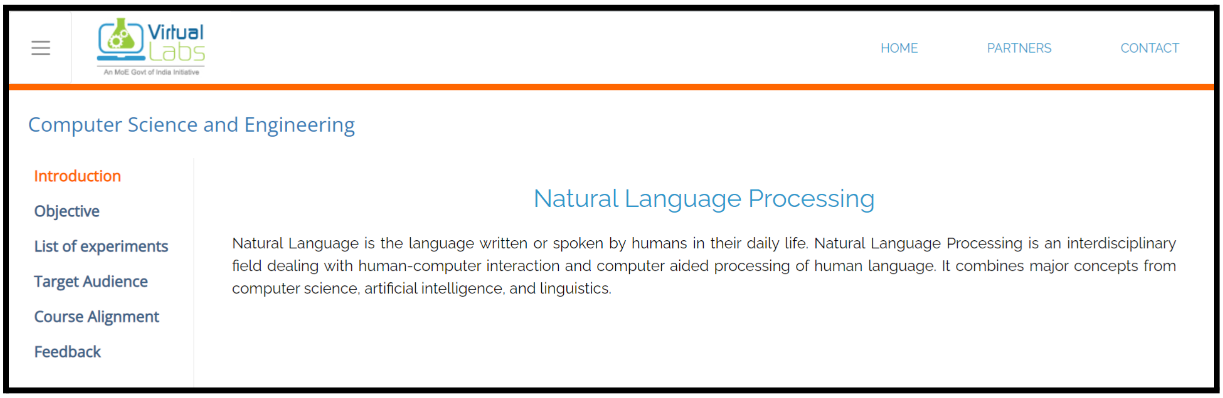
Experiment No. 1

A.1 Aim: Word Analysis and different types of Morphologies associated with languages.

PART B
(PART B: TO BE COMPLETED BY STUDENTS)

Roll No. 50	**Name:** AMEY MAHENDRA THAKUR
Class: BE COMPS B 50	**Batch:** B3
Date of Experiment: 19-01-2021	**Date of Submission:** 19-01-2021
Grade:	

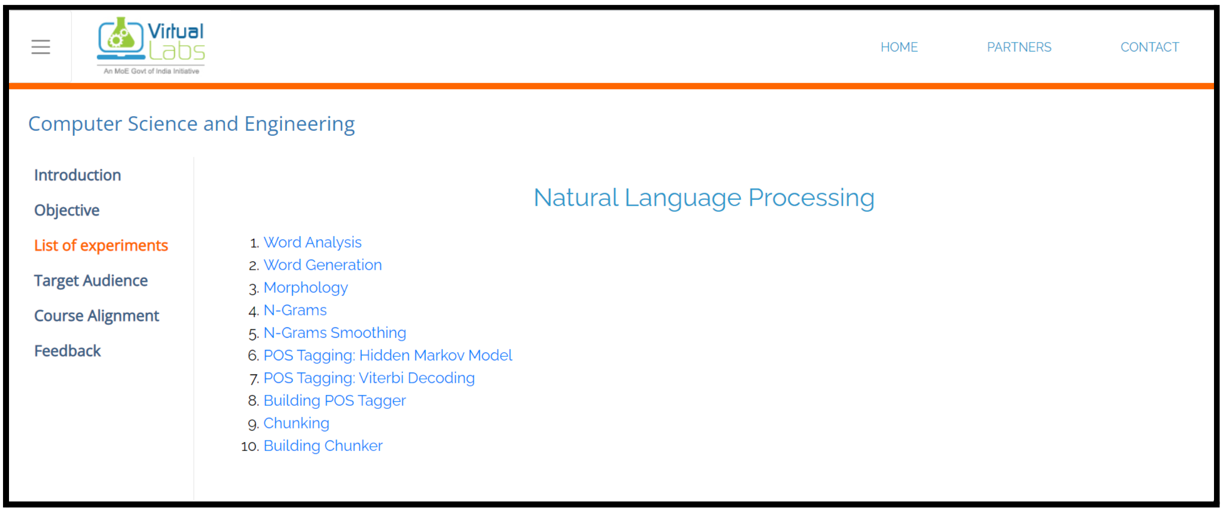
B.1 Virtual Lab (Input & Output):



Computer Science and Engineering

Natural Language Processing

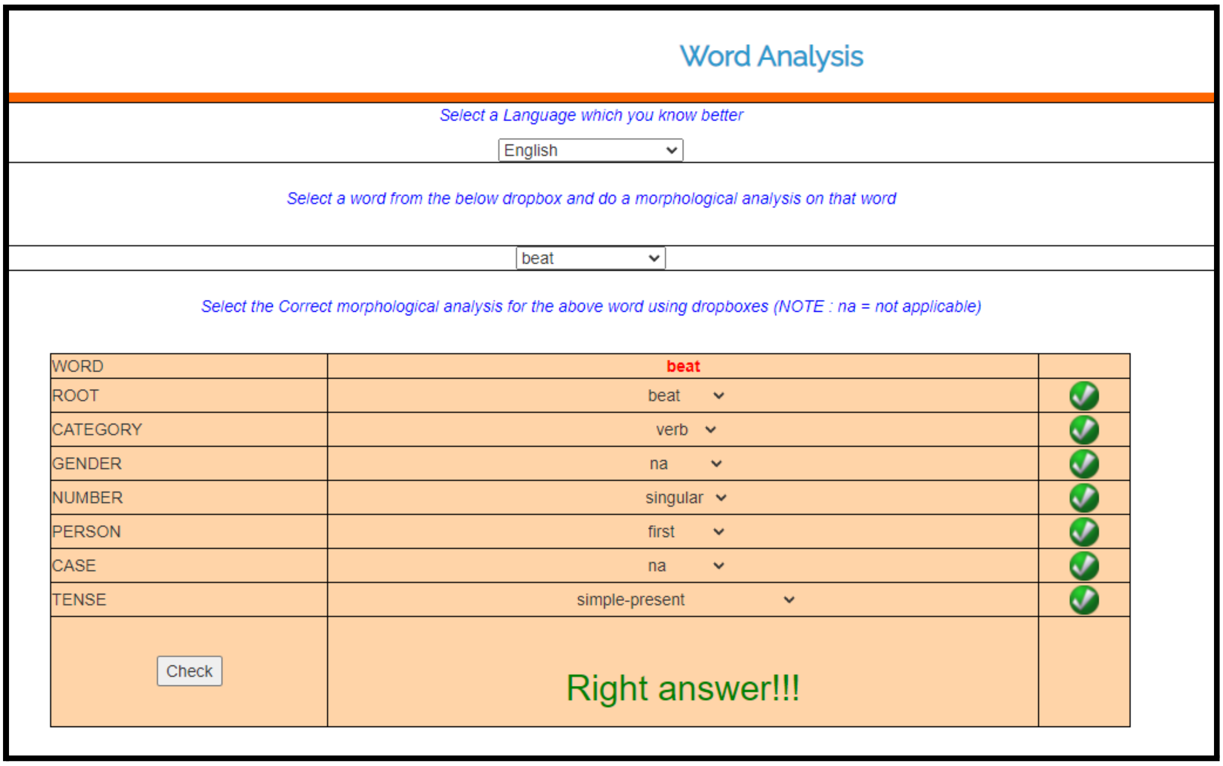
Natural Language is the language written or spoken by humans in their daily life. Natural Language Processing is an interdisciplinary field dealing with human-computer interaction and computer aided processing of human language. It combines major concepts from computer science, artificial intelligence, and linguistics.



Computer Science and Engineering

Natural Language Processing

1. Word Analysis
2. Word Generation
3. Morphology
4. N-Grams
5. N-Grams Smoothing
6. POS Tagging: Hidden Markov Model
7. POS Tagging: Viterbi Decoding
8. Building POS Tagger
9. Chunking
10. Building Chunker



Word Analysis

Select a Language which you know better

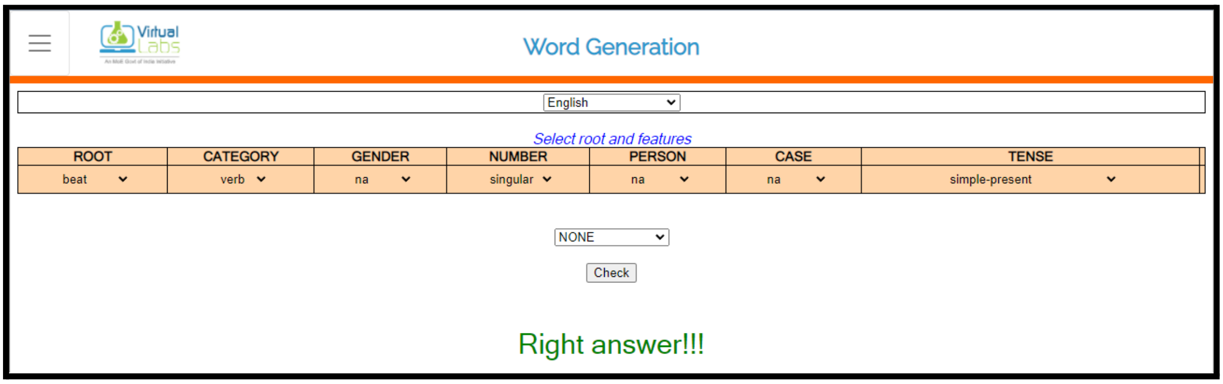
English

Select a word from the below dropbox and do a morphological analysis on that word

beat

Select the Correct morphological analysis for the above word using dropboxes (NOTE : na = not applicable)

WORD	beat	
ROOT	beat	✓
CATEGORY	verb	✓
GENDER	na	✓
NUMBER	singular	✓
PERSON	first	✓
CASE	na	✓
TENSE	simple-present	✓
Check	Right answer!!!	



Word Generation

English

Select root and features

ROOT	CATEGORY	GENDER	NUMBER	PERSON	CASE	TENSE
beat	verb	na	singular	na	na	simple-present

NONE

Check

Right answer!!!

B.3 Observations & Learning:

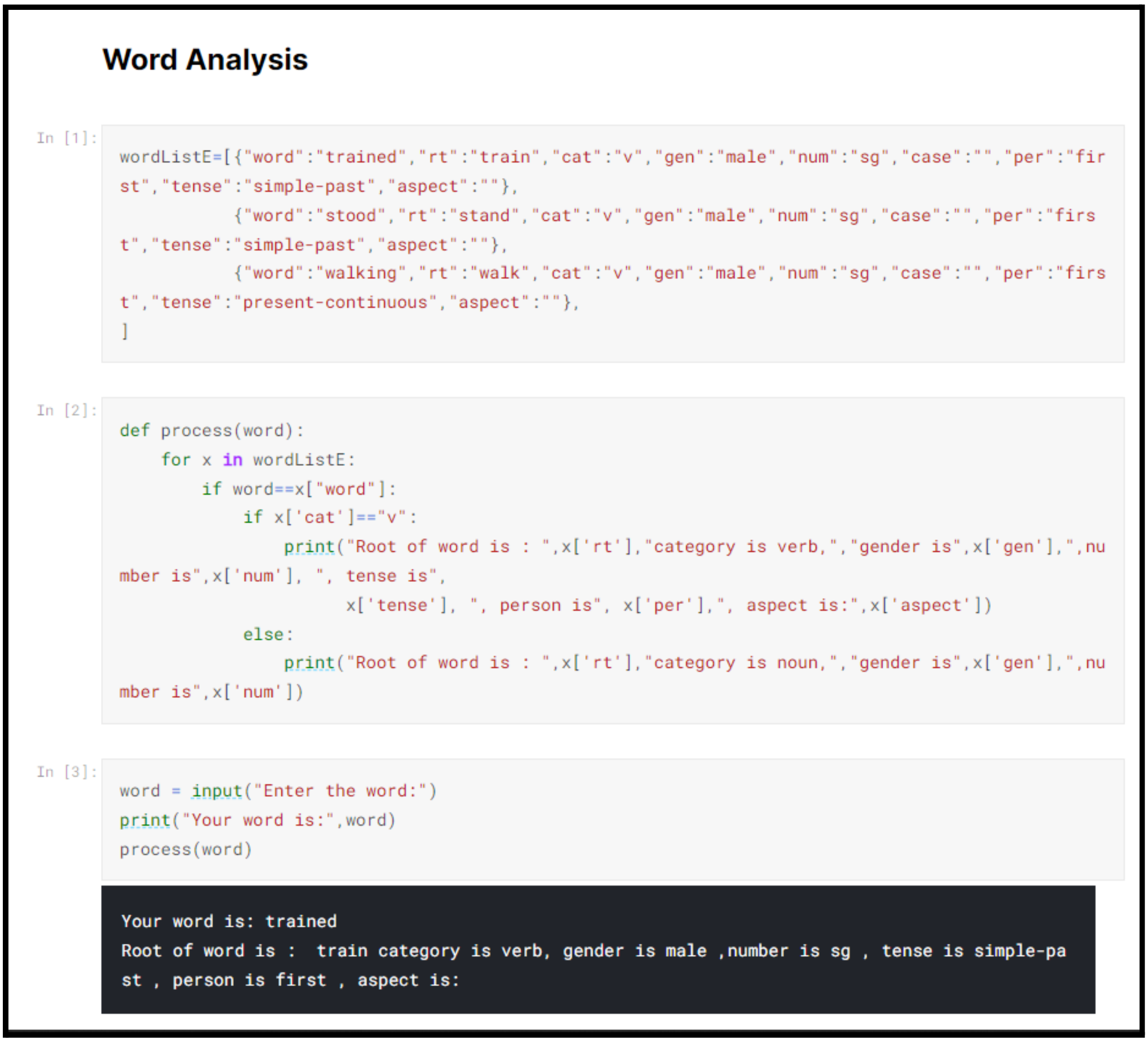
- NLP (Natural Language Processing) is a branch of artificial intelligence that examines how computers interact with human languages, specifically how to design computers to process and evaluate huge volumes of natural language data. NLP is frequently used to categorise text data.
- Text classification is the challenge of categorising text data based on its content. The process of producing features for a machine learning model from raw text input is the most significant aspect of text categorization.

B.4 Conclusion:

- We comprehend the ideas of Word Analysis and different sorts of Morphologies related with languages after successfully completing the task.

B.5 Questions of Curiosity:

Q1: Display grammatical features for words. Accepts words as input Refer below code for word analysis and try for another minimum of 10 English Words.



Word Analysis

In [1]:

```
wordListE=[{"word":"trained","rt":"train","cat":"v","gen":"male","num":"sg","case":"","per":"fir
st","tense":"simple-past","aspect":""},
           {"word":"stood","rt":"stand","cat":"v","gen":"male","num":"sg","case":"","per":"firs
t","tense":"simple-past","aspect":""},
           {"word":"walking","rt":"walk","cat":"v","gen":"male","num":"sg","case":"","per":"firs
t","tense":"present-continuous","aspect":""},
]
```

In [2]:

```
def process(word):
    for x in wordListE:
        if word==x["word"]:
            if x['cat']=="v":
                print("Root of word is : ",x['rt'],"category is verb,","gender is",x['gen'],",nu
mber is",x['num'], ", tense is",
                      x['tense'], ", person is", x['per'],", aspect is:",x['aspect'])
            else:
                print("Root of word is : ",x['rt'],"category is noun,","gender is",x['gen'],",nu
mber is",x['num'])
```

In [3]:

```
word = input("Enter the word:")
print("Your word is:",word)
process(word)
```

```
Your word is: trained
Root of word is :  train category is verb, gender is male ,number is sg , tense is simple-pa
st , person is first , aspect is:
```

Ans:

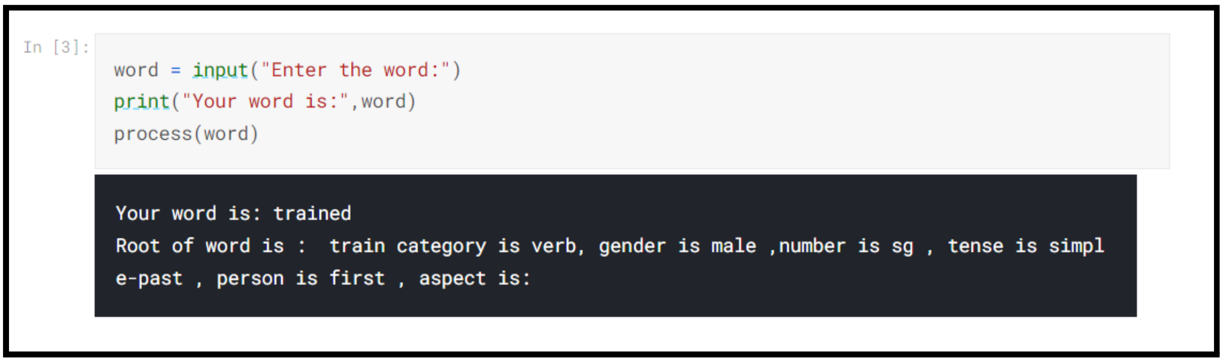
Program Code:

```
wordListE=[{"word":"boy","rt":"boy","cat":"n","gen":"m","num":"sg","case":"","per":"","tense":"","aspect":""},
{"word":"boys","rt":"boy","cat":"n","gen":"m","num":"pl","case":"","per":"","tense":"","aspect":""},
{"word":"toys","rt":"boy","cat":"n","gen":"m","num":"pl","case":"","per":"","tense":"","aspect":""},
{"word":"play","rt":"play","cat":"v","gen":"","num":"sg","case":"","per":"first","tense":"simple-present","aspect":""},
{"word":"trained","rt":"train","cat":"v","gen":"male","num":"sg","case":"","per":"first","tense":"simple-past","aspect":""},
{"word":"stood","rt":"stand","cat":"v","gen":"male","num":"sg","case":"","per":"first","tense":"simple-past","aspect":""},
{"word":"children","rt":"child","cat":"n","gen":"","num":"pl","case":"","per":"","tense":"","aspect":""},
{"word":"walking","rt":"walk","cat":"v","gen":"male","num":"sg","case":"","per":"first","tense":"present-continuous","aspect":""},
{"word":"chair","rt":"chiar","cat":"n","gen":"","num":"sg","case":"","per":"","tense":"","aspect":""},
{"word":"crawl","rt":"crawl","cat":"v","gen":"","num":"sg","case":"","per":"first","tense":"simple-present","aspect":""}
]

def process(word):
   for x in wordListE:
      if word==x["word"]:
         if x['cat']=="v":
            print("Root of word is : ",x['rt'],"category is verb,","gender is",x['gen'],",number is",x['num'], ", tense is", x['tense'], ", person is", x['per'],", aspect is:",x['aspect'])
         else:
            print("Root of word is : ",x['rt'],"category is noun,","gender is",x['gen'],",number is",x['num'])

word = input("Enter the word:")
print("Your word is:",word)
process(word)
```

Output:



```
In [3]:
word = input("Enter the word:")
print("Your word is:",word)
process(word)
```

```
Your word is: trained
Root of word is :  train category is verb, gender is male ,number is sg , tense is simpl
e-past , person is first , aspect is:
```

Q2: Refer to the following data and answer the question below.

1. List 1: taller, shorter, higher, lower, smarter
2. List 2: mower, teacher, sailor, caller, operator
3. List 3: never, cover, finger, river

Do the words ending with 'er'/'or' have some common features?

ANS:

1. List 1: The words ending with -er are comparative words which shows a relationship between two entities
2. List 2: All these words are related to a particular profession or kind of work a person is doing. For example 1. Teacher- a person whose job is to teach; 2. Operator- a person whose job is to work a particular machine or piece of equipment
3. List 3: The words don't have a common feature except they are rhyming.

Q3: Identify root and suffix in the following words.

1. Stronger
2. Goodness
3. Teacher
4. Achievement

Ans:

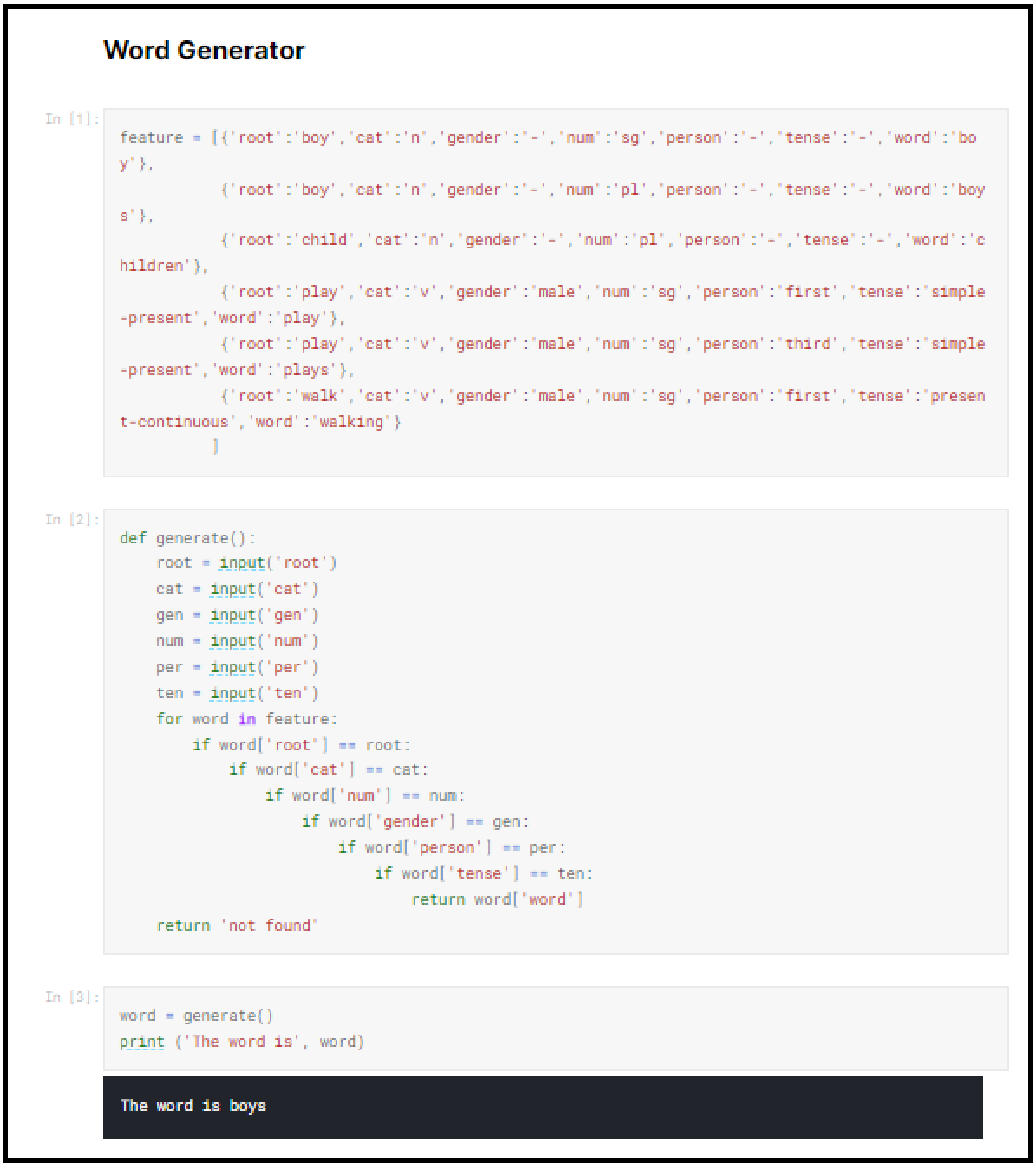
WORD	ROOT	SUFFIX
Stronger	strong	e
Goodness	good	ness
Teacher	teach	er
Achievement	achieve	ment

Q4: Generate words for the following features.

English:

root: boy	category: noun	number: singular			
root: child	category: noun	number: plural			
root: play	category: verb	gender: male	number: singular	person: first	tense: simple-present

root: play	category: verb	gender: male	number: singular	person: third	tense: simple-present



Word Generator

In [1]:
```
feature = [{'root':'boy','cat':'n','gender':'-','num':'sg','person':'-','tense':'-','word':'bo
y'},
           {'root':'boy','cat':'n','gender':'-','num':'pl','person':'-','tense':'-','word':'boy
s'},
           {'root':'child','cat':'n','gender':'-','num':'pl','person':'-','tense':'-','word':'c
hildren'},
           {'root':'play','cat':'v','gender':'male','num':'sg','person':'first','tense':'simple
-present','word':'play'},
           {'root':'play','cat':'v','gender':'male','num':'sg','person':'third','tense':'simple
-present','word':'plays'},
           {'root':'walk','cat':'v','gender':'male','num':'sg','person':'first','tense':'presen
t-continuous','word':'walking'}
          ]
```

In [2]:
```
def generate():
    root = input('root')
    cat = input('cat')
    gen = input('gen')
    num = input('num')
    per = input('per')
    ten = input('ten')
    for word in feature:
        if word['root'] == root:
            if word['cat'] == cat:
                if word['num'] == num:
                    if word['gender'] == gen:
                        if word['person'] == per:
                            if word['tense'] == ten:
                                return word['word']
    return 'not found'
```

In [3]:
```
word = generate()
print ('The word is', word)
```

```
The word is boys
```

Ans:

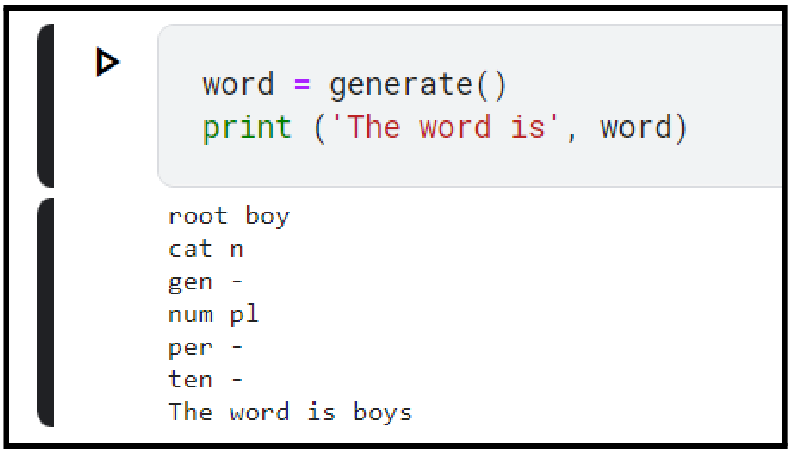
Program Code

feature =
[{'root':'boy','cat':'n','gender':'-','num':'sg','person':'-','tense':'-','word':'boy'},
        {'root':'boy','cat':'n','gender':'-','num':'pl','person':'-','tense':'-','word':'boys'},

{'root':'child','cat':'n','gender':'-','num':'pl','person':'-','tense':'-','word':'children'},

```
{'root':'play','cat':'v','gender':'male','num':'sg','person':'first','tense':'simple-present'
,'word':'play'},
{'root':'play','cat':'v','gender':'male','num':'sg','person':'third','tense':'simple-present
','word':'plays'},
{'root':'walk','cat':'v','gender':'male','num':'sg','person':'first','tense':'present-contin
uous','word':'walking'}
     ]
def generate():
  root = input('root')
  cat = input('cat')
  gen = input('gen')
  num = input('num')
  per = input('per')
  ten = input('ten')
  for word in feature:
    if word['root'] == root:
      if word['cat'] == cat:
        if word['num'] == num:
          if word['gender'] == gen:
            if word['person'] == per:
              if word['tense'] == ten:
                return word['word']
  return 'not found'
word = generate()
print ('The word is', word)
```

Output:



```
word = generate()
print ('The word is', word)
```

```
root boy
cat n
gen -
num pl
per -
ten -
The word is boys
```

Answers:

1. Boy
2. Children
3. Play
4. Plays