NAME = PARAG PUJARI

REGISTRATION NUMBER = 19MAI0017

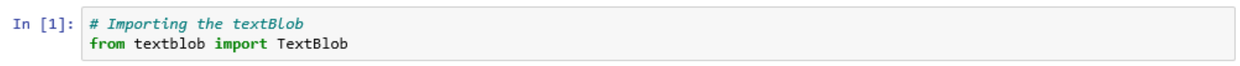
NATURAL LANGUAGE PROCESSING DIGITAL ASSESSMENT

Write up on any NLP tool apart from NLTK

TEXTBLOB:

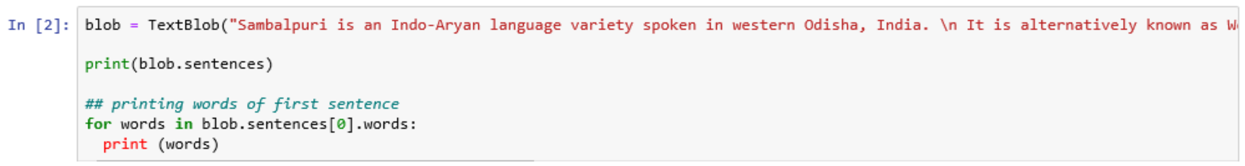
The tool which I am going to use for performing NLP operations is TextBlob. TextBlob is a python library and offers a simple API to access its methods and perform basic NLP tasks. TextBlob behaves like python strings, i.e by which the texts can be transferred and used according to our needs. Textblob is a useful option for implementing NLTK's functionality as it is an open-source Python library for processing textual unstructured data. TextBlob heavily depends on Python NLTK and pattern module by CLIPS. Textblob has an inbuilt API and is relatively lightweight python library hence is very attractive and useful tool for the following tasks like Noun Phrase Extraction, Parts of Speech Tagging, Sentiment Analysis etc.

CODE DEMONSTRATION:



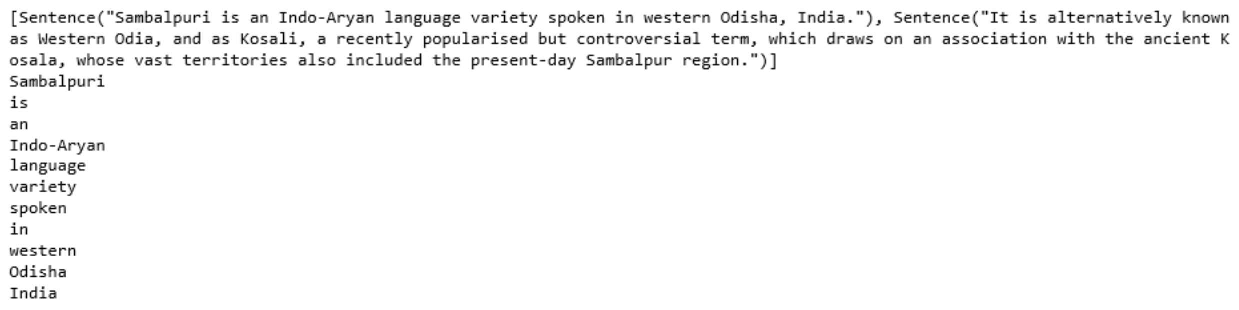
```
In [1]: # Importing the textBlob
        from textblob import TextBlob
```

OBSERVATION: Here I have imported TextBlob. The main purpose of TextBlob is to process textual data. It has a bunch of cool which can help in analyzing and understanding text data in python.



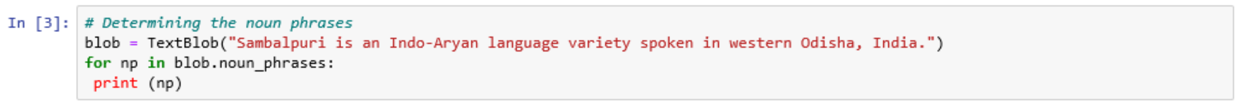
```
In [2]: blob = TextBlob("Sambalpuri is an Indo-Aryan language variety spoken in western Odisha, India. \n It is alternatively known as W

        print(blob.sentences)

        ## printing words of first sentence
        for words in blob.sentences[0].words:
          print (words)
```

OUTPUT:



```
[Sentence("Sambalpuri is an Indo-Aryan language variety spoken in western Odisha, India."), Sentence("It is alternatively known
as Western Odia, and as Kosali, a recently popularised but controversial term, which draws on an association with the ancient K
osala, whose vast territories also included the present-day Sambalpur region.")]
Sambalpuri
is
an
Indo-Aryan
language
variety
spoken
in
western
Odisha
India
```

OBSERVATION: Here I have first created an object for TextBlob called blob. Then this blob object calls the TextBlob function which accepts the sentence in the form of input. Then I have printed the sentence. I have then printed the total number of words in the first sentence.

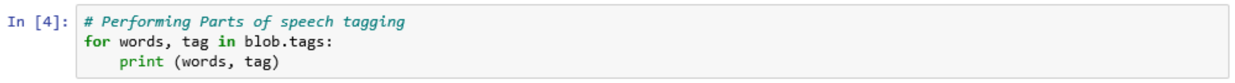


```
In [3]: # Determining the noun phrases
        blob = TextBlob("Sambalpuri is an Indo-Aryan language variety spoken in western Odisha, India.")
        for np in blob.noun_phrases:
         print (np)
```

<u>OUTPUT:</u>

```
sambalpuri
indo-aryan
language variety
odisha
india
```

<u>OBSERVATION</u>: I have printed the total number of nouns used in this TextBlob sentence.

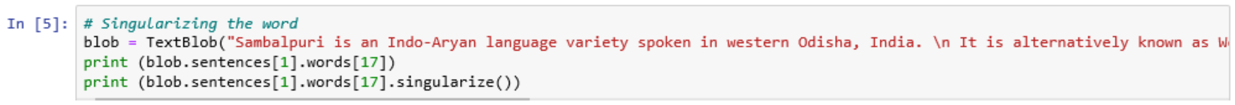


```
In [4]: # Performing Parts of speech tagging
        for words, tag in blob.tags:
            print (words, tag)
```

<u>OUTPUT:</u>

```
Sambalpuri NNP
is VBZ
an DT
Indo-Aryan JJ
language NN
variety NN
spoken VBN
in IN
western JJ
Odisha NNP
India NNP
```

<u>OBSERVATION</u>: Here I have printed each and every word along with its tagged parts of speech.

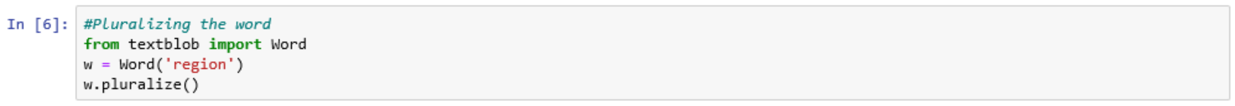


```
In [5]: # Singularizing the word
        blob = TextBlob("Sambalpuri is an Indo-Aryan language variety spoken in western Odisha, India. \n It is alternatively known as W
        print (blob.sentences[1].words[17])
        print (blob.sentences[1].words[17].singularize())
```

<u>OUTPUT:</u>

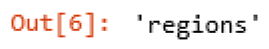
```
draws
draw
```

<u>OBSERVATION</u>: Here I have converted the plural form of the word ‘draws’ present at index position seventeenth of the first sentence into its singular form which is draw through the singularize function.



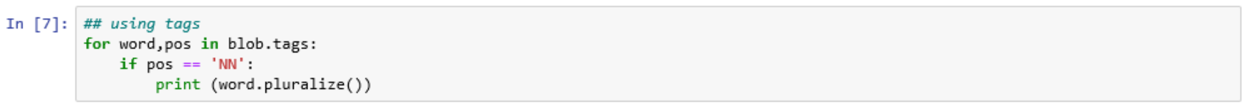
```
In [6]: #Pluralizing the word
        from textblob import Word
        w = Word('region')
        w.pluralize()
```

OUTPUT:



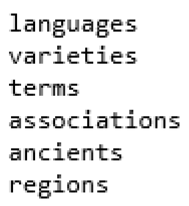
```
Out[6]: 'regions'
```

OBSERVATION: Here I have used the pluralize function which converts the word ‘region’ into its plural form.



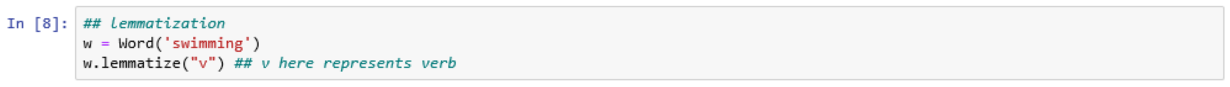
```
In [7]: ## using tags
        for word,pos in blob.tags:
            if pos == 'NN':
                print (word.pluralize())
```

OUTPUT:



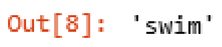
```
languages
varieties
terms
associations
ancients
regions
```

OBSERVATION: Here I have taken the TextBlob sentence and checked every word of the sentence if it is a noun. If noun is encountered, then that noun word is converted to its plural form.



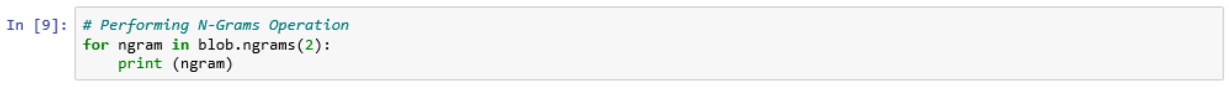
```
In [8]: ## Lemmatization
        w = Word('swimming')
        w.lemmatize("v") ## v here represents verb
```

OUTPUT:



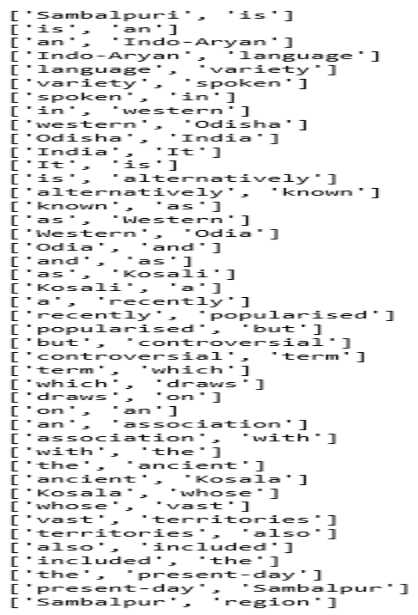
```
Out[8]: 'swim'
```

OBSERVATION: Here I have used the lemmatize function on the word ‘swimming’ so as to convert it back to its root word ‘swim’.



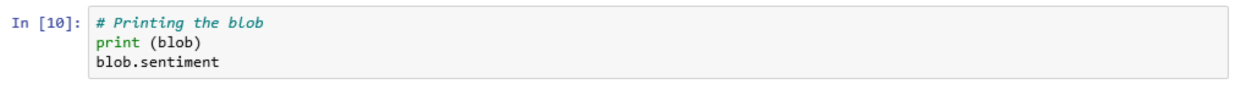
```
In [9]: # Performing N-Grams Operation
        for ngram in blob.ngrams(2):
            print (ngram)
```

OUTPUT:



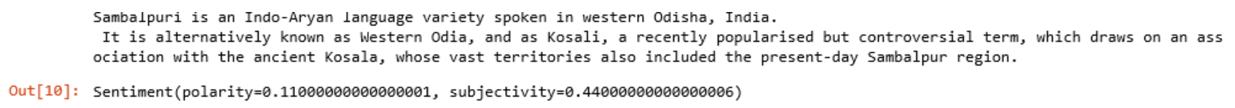
```
['Sambalpuri', 'is']
['is', 'an']
['an', 'Indo-Aryan']
['Indo-Aryan', 'language']
['language', 'variety']
['variety', 'spoken']
['spoken', 'in']
['in', 'western']
['western', 'Odisha']
['Odisha', 'India']
['India', 'It']
['It', 'is']
['is', 'alternatively']
['alternatively', 'known']
['known', 'as']
['as', 'Western']
['Western', 'Odia']
['Odia', 'and']
['and', 'as']
['as', 'Kosali']
['Kosali', 'a']
['a', 'recently']
['recently', 'popularised']
['popularised', 'but']
['but', 'controversial']
['controversial', 'term']
['term', 'which']
['which', 'draws']
['draws', 'on']
['on', 'an']
['an', 'association']
['association', 'with']
['with', 'the']
['the', 'ancient']
['ancient', 'Kosala']
['Kosala', 'whose']
['whose', 'vast']
['vast', 'territories']
['territories', 'also']
['also', 'included']
['included', 'the']
['the', 'present-day']
['present-day', 'Sambalpur']
['Sambalpur', 'region']
```

<u>OBSERVATION</u>: Here I have used n-gram operation on the TextBlob Sentence with two words at each phase. So using the first two words, I have moved and processed one word forward. In this way, I have processed the entire TextBlob sentence.



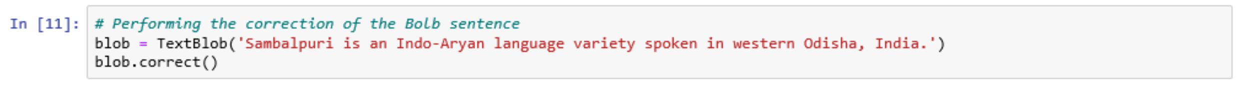
```
In [10]: # Printing the blob
         print (blob)
         blob.sentiment
```

<u>OUTPUT:</u>



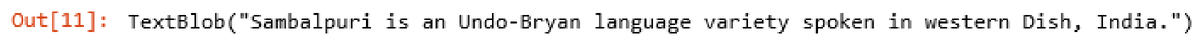
```
         Sambalpuri is an Indo-Aryan language variety spoken in western Odisha, India.
          It is alternatively known as Western Odia, and as Kosali, a recently popularised but controversial term, which draws on an ass
         ociation with the ancient Kosala, whose vast territories also included the present-day Sambalpur region.

Out[10]: Sentiment(polarity=0.11000000000000001, subjectivity=0.44000000000000006)
```

<u>OBSERVATION</u>: Here I have printed the TextBlob sentence and printed the polarity score and subjectivity score of the sentiment function.



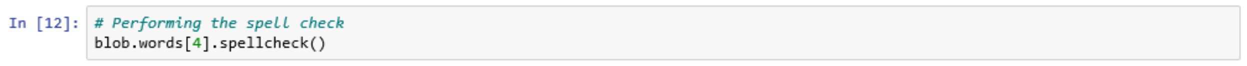
```
In [11]: # Performing the correction of the Bolb sentence
         blob = TextBlob('Sambalpuri is an Indo-Aryan language variety spoken in western Odisha, India.')
         blob.correct()
```

<u>OUTPUT:</u>



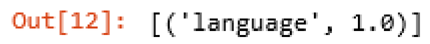
```
Out[11]: TextBlob("Sambalpuri is an Undo-Bryan language variety spoken in western Dish, India.")
```

<u>OBSERVATION</u>: Here I have used blob.correct() function on the TextBlob sentence and returned all the words of the sentence with correct spellings.

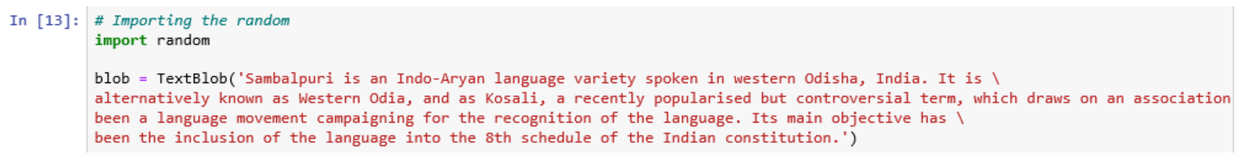


```
In [12]: # Performing the spell check
         blob.words[4].spellcheck()
```

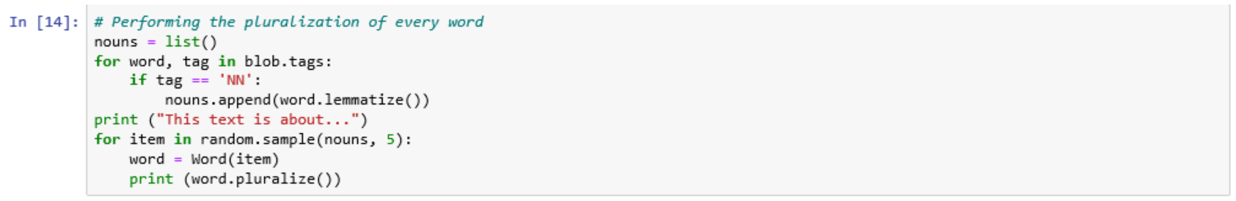
OUTPUT:


```
Out[12]: [('language', 1.0)]
```

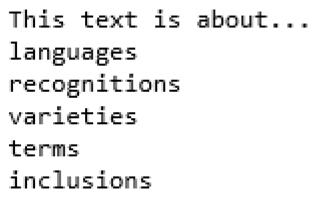
OBSERVATION: Here I have used the spellcheck function and performed this spellcheck operation on the word present at the index position 4. Now this function has returned the word with the correct spelling.


```
In [13]: # Importing the random
         import random

         blob = TextBlob('Sambalpuri is an Indo-Aryan language variety spoken in western Odisha, India. It is \
         alternatively known as Western Odia, and as Kosali, a recently popularised but controversial term, which draws on an association
         been a language movement campaigning for the recognition of the language. Its main objective has \
         been the inclusion of the language into the 8th schedule of the Indian constitution.')
```

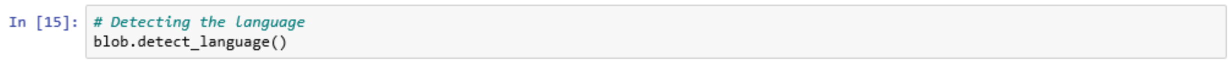
OBSERVATION: Here I have imported the random function with which I can do variety of operations with the random number generated.


```
In [14]: # Performing the pluralization of every word
         nouns = list()
         for word, tag in blob.tags:
             if tag == 'NN':
                 nouns.append(word.lemmatize())
         print ("This text is about...")
         for item in random.sample(nouns, 5):
             word = Word(item)
             print (word.pluralize())
```

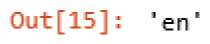
OUTPUT:


```
This text is about...
languages
recognitions
varieties
terms
inclusions
```

OBSERVATION: Here I have first iterated through the entire sentence. Then I have considered every word and checked if it is a noun. If noun, then I have first lemmatized the word to its root word and then appended with the noun words. I have then performed the random sampling of the sentence considering five nouns each and then I have found out the word from the sentences and converted into its plural form.


```
In [15]: # Detecting the language
         blob.detect_language()
```

OUTPUT:


```
Out[15]: 'en'
```

OBSERVATION: I have detected the kind of language used in the TextBlob sentence.

```
In [16]: # Translating the language from english to french
         blob.translate(from_lang='en', to ='fr')
```

OUTPUT:

```
Out[16]: TextBlob("Le sambalpuri est une variété de langue indo-aryenne parlée dans l'ouest d'Odisha, en Inde. Il est également connu so
         us le nom d'Odia occidental et de Kosali, un terme récemment popularisé mais controversé, qui s'appuie sur une association avec
         l'ancienne Kosala, dont les vastes territoires comprenaient également la région actuelle de Sambalpur. Un mouvement linguistiqu
         e a fait campagne pour reconnaissance de la langue. Son objectif principal a été l'inclusion de la langue dans le 8ème calendri
         er de la constitution indienne.")
```

OBSERVATION: I have converted the English form of the TextBlob sentence into the French form.

```
In [17]: # Taking the training and testing data
         training = [
         ('Tom Holland is a terrible spiderman.','pos'),
         ('a terrible Javert (Russell Crowe) ruined Les Miserables for me...','pos'),
         ('The Dark Knight Rises is the greatest superhero movie ever!','neg'),
         ('Fantastic Four should have never been made.','pos'),
         ('Wes Anderson is my favorite director!','neg'),
         ('Captain America 2 is pretty awesome.','neg'),
         ('Let\s pretend "Batman and Robin" never happened..','pos'),
         ]
         testing = [
         ('Superman was never an interesting character.','pos'),
         ('Fantastic Mr Fox is an awesome film!','neg'),
         ('Dragonball Evolution is simply terrible!!','pos')
         ]
```

OBSERVATION: Here I have considered the training data and the testing data.

```
In [18]: # Classifier
         from textblob import classifiers
         classifier = classifiers.NaiveBayesClassifier(training)
```

OBSERVATION: Here I have determined the classifier for the training data.

```
In [19]: ## decision tree classifier
         dt_classifier = classifiers.DecisionTreeClassifier(training)
```

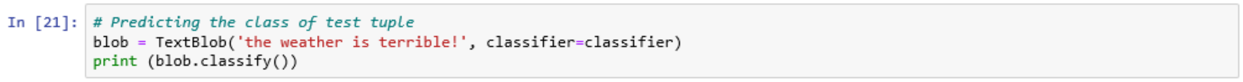
OBSERVATION: Here I have determined the decision tree classifier for the training data.

```
In [20]: # Printing the accuracy level of the classifier and displaying the show_informative_features
         print (classifier.accuracy(testing))
         classifier.show_informative_features(3)
```

OUTPUT:

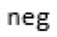
```
1.0
Most Informative Features
            contains(is) = True              neg : pos    =      2.9 : 1.0
             contains(a) = False             neg : pos    =      1.8 : 1.0
      contains(terrible) = False             neg : pos    =      1.8 : 1.0
```

OBSERVATION: Here I have determined the accuracy level of the testing data and even displayed all the informative_features for the classifier.



```
In [21]: # Predicting the class of test tuple
         blob = TextBlob('the weather is terrible!', classifier=classifier)
         print (blob.classify())
```

OUTPUT:



```
neg
```

OBSERVATION: Here I have determined the class of the test tuple which I have inputted in the TextBlob.

REFERENCES:

1. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu , K., & Kuksa,P."Natural language processing (almost) from scratch". The Journal of Machine Learning Research, 12, 2493–2537, 2011.

2. Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., etal. "Part-of-speech tagging for twitter: Annotation, features, and experiments". In Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies: Short papers - volume 2 HLT '11,pp. 42–47,2011.

3. Shrija Madhu "An approach to analyze suicidal tendency in blogs and tweets using Sentiment Analysis". International Journal of Scientific Research in Computer Science and Engineering Vol.6, Issue.4, pp.34-36, August (2018).