Name - Amaan Mashooq Nasser

Reg. No. – 19BCE2501

NLP Tasks pt3

```
#TASK-1
#lexicon -> collection of words/Phrases + Information
#lexicon has lexical entries-> each entry is word/Phrases
import nltk
nltk.download('stopwords')
#1. Stopwords.
from nltk.corpus import stopwords
stopwords.words('spanish')
```

```
 'fuéramos',
 'fuerais',
 'fueran',
 'fuese',
 'fueses',
 'fuésemos',
 'fueseis',
 'fuesen',
 'sintiendo',
 'sentido',
 'sentida',
 'sentidos',
 'sentidas',
 'siente',
 'sentid',
 'tengo',
 'tienes',
 'tiene',
 'tenemos',
 'tenéis',
 'tienen',
 'tenga',
 'tengas',
 'tengamos',
 'tengáis',
 'tengan',
 'tendré',
 'tendrás',
 'tendrá',
```

```
 'tendréis',
 'tendrán',
 'tendría',
 'tendrías',
 'tendríamos',
 'tendríais',
 'tendrían',
 'tenía',
 'tenías',
 'teníamos',
 'teníais',
 'tenían',
 'tuve',
 'tuviste',
 'tuvo',
 'tuvimos',
 'tuvisteis',
 'tuvieron',
 'tuviera',
 'tuvieras',
 'tuviéramos',
 'tuvierais',
 'tuvieran',
 'tuviese',
 'tuvieses',
 'tuviésemos',
 'tuvieseis',
 'tuviesen',
 'teniendo',
 'tenido',
 'tenida',
 'tenidos',
 'tenidas',
 'tened']
```

```
#1.2 CMU wordlist
import nltk
nltk.download('cmudict')
entries = nltk.corpus.cmudict.entries()
len(entries)
```

```
[nltk_data] Downloading package cmudict to /root/nltk_data...
[nltk_data]   Unzipping corpora/cmudict.zip.
133737
```

```
[4] print(entries)
```

```
[('a', ['AH0']), ('a.', ['EY1']), ('a', ['EY1']), ...]
```

```
#1.3 Wordnet
nltk.download('wordnet')
nltk.download('omw-1.4')
from nltk.corpus import wordnet as wn
wn.synsets('sports')
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
[Synset('sport.n.01'),
 Synset('sport.n.02'),
 Synset('sport.n.03'),
 Synset('sport.n.04'),
 Synset('sport.n.05'),
 Synset('mutant.n.01'),
 Synset('fun.n.02'),
 Synset('sport.v.01'),
 Synset('frolic.v.01')]
```

```
[8] wn.synset('fun.n.02').lemma_names()
```

```
['fun', 'play', 'sport']
```

```
[9] #TASK 2- SIMPLE TEXT CLASSIFIER
    def gender_features(word):
      return{'last_letter':word[-1]}
```

```
[10] gender_features('Obama')
```

```
{'last_letter': 'a'}
```

```
import nltk
nltk.download('names')
from nltk.corpus import names
labeled_names = ([(name, 'male') for name in names.words('male.txt')] + [(name,'female') for name in names.words('female.txt')])
```

```
[nltk_data] Downloading package names to /root/nltk_data...
[nltk_data]   Package names is already up-to-date!
```

```
[13] import random
     random.shuffle(labeled_names)
```

```
[14] featuresets = [(gender_features(n), gender) for (n, gender) in labeled_names]
```

```
[15] train_set, test_test = featuresets[500:], featuresets[:500]
```

```
[16] import nltk
     classifier = nltk.NaiveBayesClassifier.train(train_set)
```

```
classifier.classify(gender_features('Mashooq'))
```

```
'female'
```

```
[18] classifier.classify(gender_features('Nasser'))
```

```
'male'
```

```
[19] print(nltk.classify.accuracy(classifier, test_test))
```

```
0.772
```

```
[21] #Task 3 VECTORISERS & COSINE SIMILARITY
     from sklearn.feature_extraction.text import CountVectorizer
     #from sklearn.feature_extraction.text import TfidfVectorizer
```

```
vect = CountVectorizer(binary = True)
corpus = ["Tessaract is good optical character recognition engine","Optical character recognition is significant"]
vect.fit(corpus)
```

```
CountVectorizer(binary=True)
```

```
[24] vocab = vect.vocabulary_
```

```
[25] for key in sorted(vocab.keys()):
       print("{}:{}".format(key, vocab[key]))
```

```
character:0
engine:1
good:2
is:3
optical:4
recognition:5
significant:6
tessaract:7
```

```
[26] print(vect.transform(["This is a good optical illusion"]).toarray())
```

```
[[0 0 1 1 1 0 0 0]]
```

```
[27] print(vect.transform(corpus).toarray())
```

```
[[1 1 1 1 1 1 0 1]
 [1 0 0 1 1 1 1 0]]
```

```
[29] from sklearn.metrics.pairwise import cosine_similarity
     similarity = cosine_similarity(vect.transform(["Google Cloud Vision is a good recognition engine"]).toarray(), vect.transform(["OCR is an optical character recognition engine"]).toarray())
```

```
print(similarity)
```

```
[[0.67082039]]
```