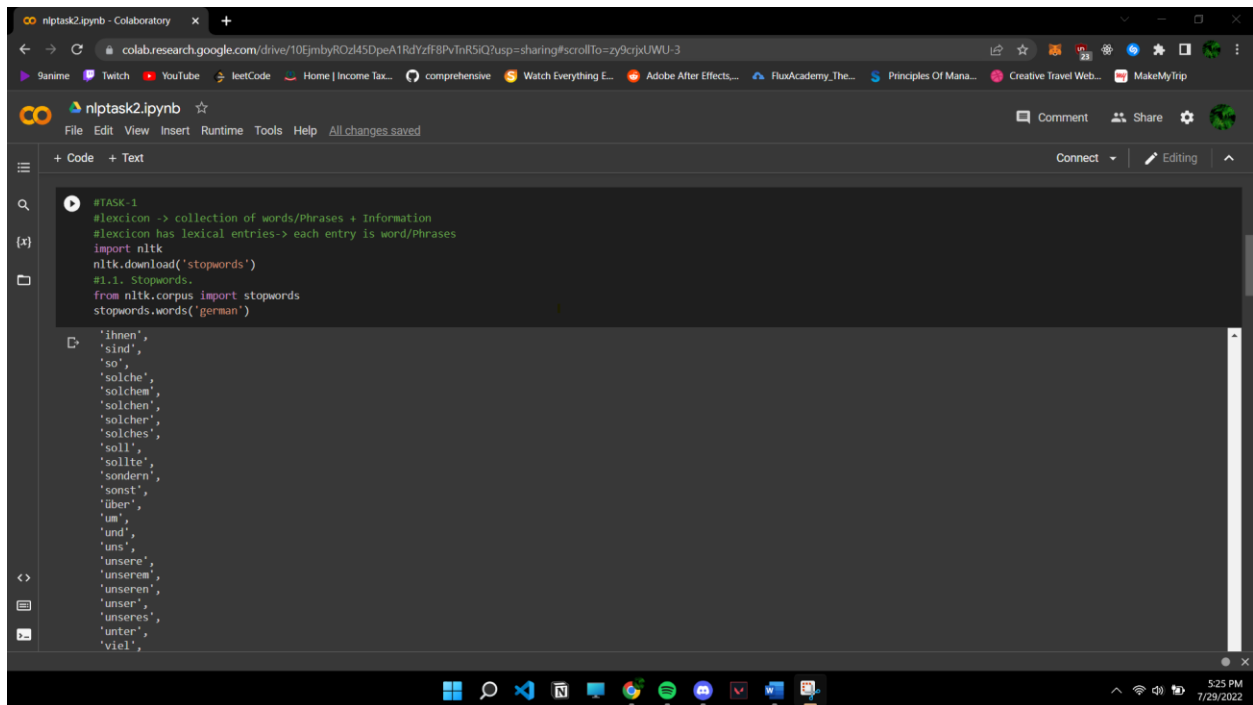


## NLP TASK 11-13

NAME: MOHAMMAD SHAHIL HUSSAIN

REG NO: 19BCE2447

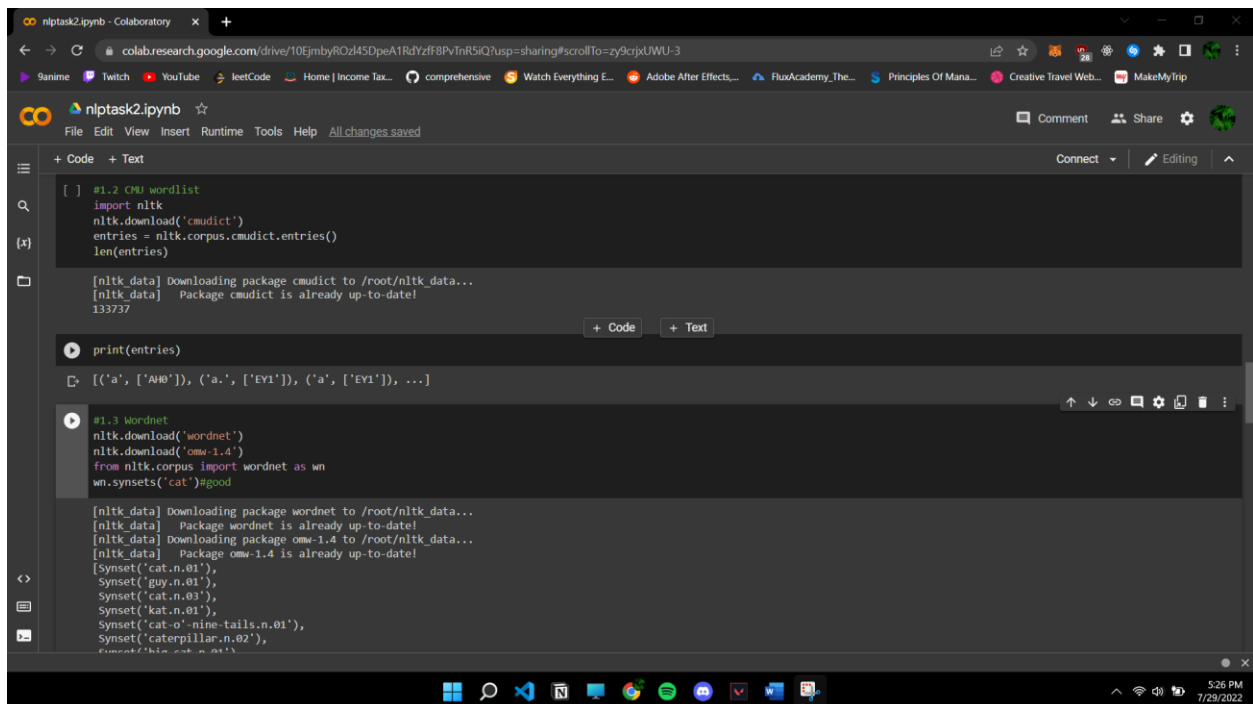


The screenshot shows a Google Colab notebook titled 'nlpTask2.ipynb'. The code cell contains the following Python code:

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

The output of the code is a list of German stopwords:

```
['ihnen',
 'sind',
 'so',
 'solche',
 'solchem',
 'solchen',
 'solcher',
 'solches',
 'soll',
 'sollte',
 'sondern',
 'sonst',
 'über',
 'um',
 'und',
 'uns',
 'unsere',
 'unserem',
 'unseren',
 'unser',
 'unseres',
 'unter',
 'viel',
```



The screenshot shows a Google Colab notebook titled 'nlpTask2.ipynb'. The code cell contains the following Python code:

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

The output of the code is:

```
[nltk_data] Downloading package cmudict to /root/nltk_data...
[nltk_data] Package cmudict is already up-to-date!
133737
```

The next code cell contains the following Python code:

```
print(entries)
```

The output of the code is:

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

The next code cell contains the following Python code:

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

The output of the code is:

```
[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('cat.n.01'),
 Synset('guy.n.01'),
 Synset('cat.n.03'),
 Synset('kat.n.01'),
 Synset('cat.o'-nine-tails.n.01'),
 Synset('caterpillar.n.02'),
 Synset('kitten.n.01'),
```

```
nlptask2.ipynb - Colaboratory
colabresearch.google.com/drive/10EjmyR0z45DpeA1RdYzF8PvTnR5iQ?usp=sharing#scrollTo=zy9crjxUWU-3
nlptask2.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[ ] Synset('kat.n.01'),
    Synset('cat-o'-nine-tails.n.01'),
    Synset('caterpillar.n.02'),
    Synset('big_cat.n.01'),
    Synset('computerized_tomography.n.01'),
    Synset('cat.v.01'),
    Synset('vomit.v.01')]

[ ] wn.synset('kat.n.01').lemma_names()

['kat', 'khat', 'qat', 'quat', 'cat', 'Arabian_tea', 'African_tea']

[ ] #TASK 2- SIMPLE TEXT CLASSIFIER
def gender_features(word):
    return['last_letter'+word[-1]]

[ ] 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!
```

```
nlptask2.ipynb - Colaboratory
colabresearch.google.com/drive/10EjmyR0z45DpeA1RdYzF8PvTnR5iQ?usp=sharing#scrollTo=zy9crjxUWU-3
nlptask2.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[ ] import random
random.shuffle(labeled_names)

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

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

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

[ ] classifier.classify(gender_features('shahil'))

'male'

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

0.806

[ ] #task 3 VECTORISERS & COSINE SIMILA
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer

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

```
CountVectorizer(binary=True)

[ ] vocab = vect.vocabulary_

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
tesseract:7

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

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

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

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

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

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

print(similarity)

[[0.67082039]]
```