

▼ CSE4022 Natural Language Processing

Digital Assignment -1

1.) Utilize Python NLTK (Natural Language Tool Kit) Platform and do the following. Install relevant Packages and Libraries

- ▼ • Explore Brown Corpus and find the size, tokens, categories,

```
import nltk
nltk.download('brown')
from nltk.corpus import brown
```

```
brown.words()
```

```
[nltk_data] Downloading package brown to /root/nltk_data...
[nltk_data]   Unzipping corpora/brown.zip.
['The', 'Fulton', 'County', 'Grand', 'Jury', 'said', ...]
```

```
len(brown.words())
```

```
1161192
```

```
brown.categories()
```

```
['adventure',
 'belles_lettres',
 'editorial',
 'fiction',
 'government',
 'hobbies',
 'humor',
 'learned',
 'lore',
 'mystery',
 'news',
 'religion',
 'reviews',
 'romance',
 'science_fiction']
```

▼ • Find the size of word tokens?

```
len(brown.words())
```

1161192

Size of each token

```
for x in brown.words():  
    print(len(x))
```

Streaming output truncated to the last 5000 lines.

1
2
2
5
2
1
3
8
4
1
4
2
3
6
1
2
3
2
4
8
4
2
6
3
4
2
7
3
6
3
5
1
7
1
4
3
4
4
9
3
8

2
7
6
4
1
6
5
1
3
4
4
3
3
7
1
3

▼ • Find the size of word types?

```
len(brown.categories())
```

15

▼ • Find the size of category “government”

```
len(brown.words(categories='government'))
```

70117

▼ • List the most frequent tokens

```
from nltk import FreqDist  
fdist = FreqDist(brown.words())  
fdist.most_common(2)
```

[('the', 62713), (',', 58334)]

▼ • Count the number of sentences

```
len(brown.sents())
```

57340

▼ 2. Explore the corpora available in NLTK

list of available corpora

```
nltk.download()
```

NLTK Downloader

```
-----  
    d) Download    l) List    u) Update    c) Config    h) Help    q) Quit  
-----
```

Downloader> l

Packages:

```
[ ] abc..... Australian Broadcasting Commission 2006  
[ ] alpino..... Alpino Dutch Treebank  
[ ] averaged_perceptron_tagger Averaged Perceptron Tagger  
[ ] averaged_perceptron_tagger_ru Averaged Perceptron Tagger (Russian)  
[ ] basque_grammars..... Grammars for Basque  
[ ] biocreative_ppi..... BioCreAtIvE (Critical Assessment of Information  
                        Extraction Systems in Biology)  
[ ] bllip_wsj_no_aux.... BLLIP Parser: WSJ Model  
[ ] book_grammars..... Grammars from NLTK Book  
[*] brown..... Brown Corpus  
[ ] brown_tei..... Brown Corpus (TEI XML Version)  
[ ] cess_cat..... CESS-CAT Treebank  
[ ] cess_esp..... CESS-ESP Treebank  
[ ] chat80..... Chat-80 Data Files  
[ ] city_database..... City Database  
[ ] cmudict..... The Carnegie Mellon Pronouncing Dictionary (0.6)  
[ ] comparative_sentences Comparative Sentence Dataset  
[ ] comtrans..... ComTrans Corpus Sample  
[ ] conll2000..... CONLL 2000 Chunking Corpus  
[ ] conll2002..... CONLL 2002 Named Entity Recognition Corpus
```

Hit Enter to continue:

```
[ ] conll2007..... Dependency Treebanks from CoNLL 2007 (Catalan  
                        and Basque Subset)  
[ ] crubadan..... Crubadan Corpus  
[ ] dependency_treebank. Dependency Parsed Treebank  
[ ] dolch..... Dolch Word List  
[ ] europarl_raw..... Sample European Parliament Proceedings Parallel  
                        Corpus  
[ ] extended_omw..... Extended Open Multilingual WordNet  
[ ] floresta..... Portuguese Treebank  
[ ] framenet_v15..... FrameNet 1.5  
[ ] framenet_v17..... FrameNet 1.7  
[ ] gazetteers..... Gazeteer Lists  
[ ] genesis..... Genesis Corpus  
[ ] gutenburg..... Project Gutenberg Selections  
[ ] ieer..... NIST IE-ER DATA SAMPLE  
[ ] inaugural..... C-Span Inaugural Address Corpus  
[ ] indian..... Indian Language POS-Tagged Corpus
```

```
[ ] jeita..... JEITA Public Morphologically Tagged Corpus (in
                  ChaSen format)
[ ] kimmo..... PC-KIMMO Data Files
[ ] knbc..... KNB Corpus (Annotated blog corpus)
Hit Enter to continue: q
```

```
-----
      d) Download   l) List   u) Update   c) Config   h) Help   q) Quit
-----
Downloader> q
True
```

▼ Gutenberg Corpus

```
nltk.download("gutenberg")
```

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

```
from nltk.corpus import gutenberg
gutenberg.fileids()
```

```
['austen-emma.txt',
 'austen-persuasion.txt',
 'austen-sense.txt',
 'bible-kjv.txt',
 'blake-poems.txt',
 'bryant-stories.txt',
 'burgess-busterbrown.txt',
 'carroll-alice.txt',
 'chesterton-ball.txt',
 'chesterton-brown.txt',
 'chesterton-thursday.txt',
 'edgeworth-parents.txt',
 'melville-moby_dick.txt',
 'milton-paradise.txt',
 'shakespeare-caesar.txt',
 'shakespeare-hamlet.txt',
 'shakespeare-macbeth.txt',
 'whitman-leaves.txt']
```

```
emma = gutenberg.words('austen-emma.txt')
emma
```

```
['[', 'Emma', 'by', 'Jane', 'Austen', '1816', ']', ...]
```

```
len(gutenberg.words())
```

```
2621613
```

▼ Reuters corpus

```
nltk.download("reuters")
```

```
[nltk_data] Downloading package reuters to /root/nltk_data...  
True
```

```
from nltk.corpus import reuters  
reuters.fileids()
```

```
['test/14826',  
 'test/14828',  
 'test/14829',  
 'test/14832',  
 'test/14833',  
 'test/14839',  
 'test/14840',  
 'test/14841',  
 'test/14842',  
 'test/14843',  
 'test/14844',  
 'test/14849',  
 'test/14852',  
 'test/14854',  
 'test/14858',  
 'test/14859',  
 'test/14860',  
 'test/14861',  
 'test/14862',  
 'test/14863',  
 'test/14865',  
 'test/14867',  
 'test/14872',  
 'test/14873',  
 'test/14875',  
 'test/14876',  
 'test/14877',  
 'test/14881',  
 'test/14882',  
 'test/14885',  
 'test/14886',  
 'test/14888',  
 'test/14890',  
 'test/14891',  
 'test/14892',  
 'test/14899',  
 'test/14900',  
 'test/14903',  
 'test/14904',  
 'test/14907',  
 'test/14909',  
 'test/14911',
```

```
'test/14912',  
'test/14913',  
'test/14918',  
'test/14919',  
'test/14921',  
'test/14922',  
'test/14923',  
'test/14926',  
'test/14928',  
'test/14930',  
'test/14931',  
'test/14932',  
'test/14933',  
'test/14934',  
'test/14941',  
'test/14943',
```

```
reuters.categories()
```

```
['acq',  
 'alum',  
 'barley',  
 'bop',  
 'carcass',  
 'castor-oil',  
 'cocoa',  
 'coconut',  
 'coconut-oil',  
 'coffee',  
 'copper',  
 'copra-cake',  
 'corn',  
 'cotton',  
 'cotton-oil',  
 'cpi',  
 'cpu',  
 'crude',  
 'dfl',  
 'dlr',  
 'dmk',  
 'earn',  
 'fuel',  
 'gas',  
 'gnp',  
 'gold',  
 'grain',  
 'groundnut',  
 'groundnut-oil',  
 'heat',  
 'hog',  
 'housing',  
 'income',  
 'instal-debt',  
 'interest',  
 'ipi',  
 'iron-steel',
```

```
'jet',  
'jobs',  
'l-cattle',  
'lead',  
'lei',  
'lin-oil',  
'livestock',  
'lumber',  
'meal-feed',  
'money-fx',  
'money-supply',  
'naphtha',  
'nat-gas',  
'nickel',  
'nkr',  
'nzdlr',  
'oat',  
'oilseed',  
'orange',  
'palladium',  
'palm-oil',
```

```
reuters.words(categories='barley')
```

```
['FRENCH', 'FREE', 'MARKET', 'CEREAL', 'EXPORT', ...]
```

▼ Indian Corpus

```
nltk.download("indian")
```

```
[nltk_data] Downloading package indian to /root/nltk_data...  
[nltk_data]   Unzipping corpora/indian.zip.  
True
```

```
from nltk.corpus import indian
```

```
print(nltk.corpus.indian.words('hindi.pos'))
```

```
['पूर्ण', 'प्रतिबंध', 'हटाओ', ':', 'इराक', 'संयुक्त', ...]
```

```
indian.fileids()
```

```
['bangla.pos', 'hindi.pos', 'marathi.pos', 'telugu.pos']
```

```
indian.words("telugu.pos")
```

```
['4', '.', 'అడిట్', 'నిర్వహణ', 'అడిటర్', 'ఒక', 'కొత్త', ...]
```



```
indian.words()
```

```
['মহিষের', 'সন্তান', ':', 'তোড়া', 'উপজাতি', '!', ...]
```

```
len(indian.words())
```

```
48754
```

3. Create a text corpus with minimum 200 words (unique contents).

```
nltk.download('punkt')
nltk.download('wordnet')
nltk.download('omw-1.4')
nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[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!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]   /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
True
```

Creating corpus of the 2 text files file1 and file2

```
import os
from nltk.corpus.reader.plaintext import PlaintextCorpusReader
```

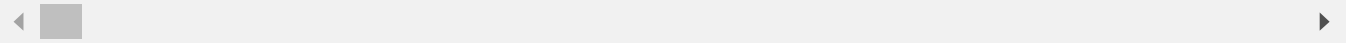
```
corpusdir = '/content/corpus'
```

```
newcorpus = PlaintextCorpusReader(corpusdir, '.*')
```

```
text=newcorpus.raw().strip()
print(newcorpus.raw().strip())
```

```
A path from a point approximately 330 metres east of the most south westerly corner of
Did he look like a doctor?
He ran into debt.
They concluded that he had told a lie.
I ran into Mary at the party last week.
I said nothing about the matter.
His brother is more patient than he is.
```

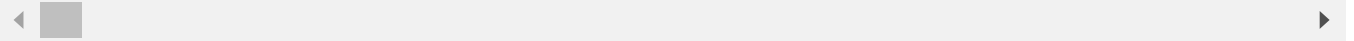
Tom was caught sneaking out of the room.
No one stops to listen to him.
Please wait around for a while.
There is going to be a storm. I clapped my hands. I have just finished my homework. Wha
I asked him if he knew my name.



▼ Paragraphs Seementation

```
print(newcorpus.paras())
```

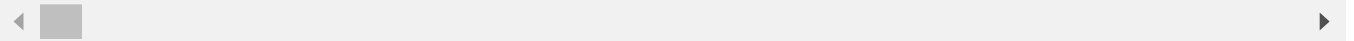
```
[['A', 'path', 'from', 'a', 'point', 'approximately', '330', 'metres', 'east', 'of', 't
```



▼ Sentences Segmentation

```
print(newcorpus.sents())
```

```
[['A', 'path', 'from', 'a', 'point', 'approximately', '330', 'metres', 'east', 'of', 't
```



```
print(nltk.sent_tokenize(text))
```

```
['A path from a point approximately 330 metres east of the most south westerly corner o
```



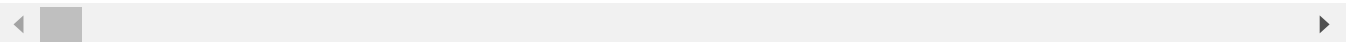
▼ Words Segementation

```
print(newcorpus.words())
```

```
['A', 'path', 'from', 'a', 'point', 'approximately', ...]
```

```
print(nltk.word_tokenize(text))
```

```
['A', 'path', 'from', 'a', 'point', 'approximately', '330', 'metres', 'east', 'of', 'th
```



▼ Convert to Lowercase

```
text=newcorpus.raw().strip()
```

```
text
```

'A path from a point approximately 330 metres east of the most south westerly corner of 17 Batherton Close, Widnes and approximately 208 metres east-south-east of the most southerly corner of Unit 3 Foundry Industrial Estate, Victoria Street, Widnes, proceeding in a generally east-north-easterly direction for approximately 28 metres to a point approximately 202 metres east-south-east of the most south-easterly corner of Unit 4 Foundry Industrial Estate, Victoria Street, and approximately 347 metres east of the most south-easterly corner of 17 Batherton Close, then proceeding in a generally northerly direction for approximately 21 metres to a point approximately 210 metres east of

```
text=text.lower()  
text
```

'a path from a point approximately 330 metres east of the most south westerly corner of 17 batherton close, widnes and approximately 208 metres east-south-east of the most southerly corner of unit 3 foundry industrial estate, victoria street, widnes, proceeding in a generally east-north-easterly direction for approximately 28 metres to a point approximately 202 metres east-south-east of the most south-easterly corner of unit 4 foundry industrial estate, victoria street, and approximately 347 metres east of the most south-easterly corner of 17 batherton close, then proceeding in a generally northerly direction for approximately 21 metres to a point approximately 210 metres east of

▼ Stop Words Removal

```
from nltk.corpus import stopwords  
stopword = stopwords.words('english')  
word_tokens = nltk.word_tokenize(text)  
removing_stopwords = [word for word in word_tokens if word not in stopword]  
print (removing_stopwords)
```

['path', 'point', 'approximately', '330', 'metres', 'east', 'south', 'westerly', 'corne

▼ Stemming (Porter Stemmer Algorithm)

```
from nltk.stem import SnowballStemmer  
stopword = stopwords.words('english')  
snowball_stemmer = SnowballStemmer('english')  
word_tokens = nltk.word_tokenize(text)  
stemmed_word = [snowball_stemmer.stem(word) for word in word_tokens]  
print (stemmed_word)
```

['a', 'path', 'from', 'a', 'point', 'approxim', '330', 'metr', 'east', 'of', 'the', 'mo

▼ Lemmatization

```
from nltk.stem import WordNetLemmatizer
stopword = stopwords.words('english')
wordnet_lemmatizer = WordNetLemmatizer()
word_tokens = nltk.word_tokenize(text)
lemmatized_word = [wordnet_lemmatizer.lemmatize(word) for word in word_tokens]
print (lemmatized_word)
```

['A', 'path', 'from', 'a', 'point', 'approximately', '330', 'metre', 'east', 'of', 'the']

▼ POS Tagging

```
word = nltk.word_tokenize(text)
pos_tag = nltk.pos_tag(word)
print (pos_tag)
```

[('A', 'DT'), ('path', 'NN'), ('from', 'IN'), ('a', 'DT'), ('point', 'NN'), ('approxima