In [1]:
```
import nltk
from nltk import word_tokenize
from nltk import sent_tokenize
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\user\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping tokenizers\punkt.zip.
```

Out[1]:
```
True
```

In [2]:
```
import pandas as pd
```

In [3]:
```
data=pd.read_csv("expt-7 twitter_parsed.csv")
data
```

Out[3]:

	index	id	Text	Annotation	oh_label
0	5.74948705591165E+017	5.74948705591165E+017	@halalflaws @biebervalue @greenlinerzjm I read...	none	0.0
1	5.71917888690393E+017	5.71917888690393E+017	@ShreyaBafna3 Now you idiots claim that people...	none	0.0
2	3.90255841338601E+017	3.90255841338601E+017	RT @Mooseoftorment Call me sexist, but when I ...	sexism	1.0
3	5.68208850655916E+017	5.68208850655916E+017	@g0ssipsquirrelx Wrong, ISIS follows the examp...	racism	1.0
4	5.75596338802373E+017	5.75596338802373E+017	#mkr No No No No No No	none	0.0
...	...	...	...	...	...
16846	5.75606766236475E+017	5.75606766236475E+017	Feeling so sorry for the girls, they should be...	none	0.0
16847	5.72333822886326E+017	5.72333822886326E+017	#MKR 'pretty good dishes we're happy with' - O...	none	0.0
16848	5.72326950057845E+017	5.72326950057845E+017	RT @colonelkickhead: Deconstructed lemon tart!...	none	0.0
16849	5.74799612642357E+017	5.74799612642357E+017	@versacezaynx @nyazpolitics @greenlinerzjm You...	none	0.0
16850	5.68826121153684E+017	5.68826121153684E+017	And before you protest that you're *not* mad, ...	none	0.0

16851 rows × 5 columns

In [4]:
```
data.columns
```

Out[4]:
```
Index(['index', 'id', 'Text', 'Annotation', 'oh_label'], dtype='object')
```

In [6]:
```
data.Text
```

Out[6]:
```
0        @halalflaws @biebervalue @greenlinerzjm I read...
1        @ShreyaBafna3 Now you idiots claim that people...
2        RT @Mooseoftorment Call me sexist, but when I ...
3        @g0ssipsquirrelx Wrong, ISIS follows the examp...
4                                    #mkr No No No No No No
                               ...
16846    Feeling so sorry for the girls, they should be...
16847    #MKR 'pretty good dishes we're happy with' - O...
16848    RT @colonelkickhead: Deconstructed lemon tart!...
16849    @versacezaynx @nyazpolitics @greenlinerzjm You...
16850    And before you protest that you're *not* mad, ...
Name: Text, Length: 16851, dtype: object
```

In [7]:
```
texts=[]
for t in data['Text'][0:10]:
  texts.append(t)
texts
```

Out[7]:
```
['@halalflaws @biebervalue @greenlinerzjm I read them in context.No change in
meaning. The history of Islamic slavery. https://t.co/xWJzpSodGj', (https://
t.co/xWJzpSodGj',)
 '@ShreyaBafna3 Now you idiots claim that people who tried to stop him from b
ecoming a terrorist made him a terrorist. Islamically brain dead.',
 "RT @Mooseoftorment Call me sexist, but when I go to an auto place, I'd rath
er talk to a guy",
 '@g0ssipsquirrelx Wrong, ISIS follows the example of Mohammed and the Quran
exactly.',
 '#mkr No No No No No No',
 "RT @TRobinsonNewEra: http://t.co/nkkCbpcHEo (http://t.co/nkkCbpcHEo) Saudi
preacher who 'raped and tortured' his five -year-old daughter to death is rel
eased after …",
 'RT @Millhouse66 @Maureen_JS nooo not sexist but most women are bad driver
s',
 "Going to make some pancakes.....Don't hve any strawberries ....🍓🍓🍓🍓bu
t I hve bananas .....🍌🍌🍌🍌. ;))) #MKR",
 'RT @ahtweet: @freebsdgirl How dare you have feelings is a fantastic way to
dehumanize someone.',
 "RT @Newmanzaa: There's something wrong when a girl wins Wayne Rooney street
striker #NotSexist"]
```

In [8]:
```
tokenized_word=word_tokenize(texts[1])
print(tokenized_word)
```

```
['@', 'ShreyaBafna3', 'Now', 'you', 'idiots', 'claim', 'that', 'people', 'wh
o', 'tried', 'to', 'stop', 'him', 'from', 'becoming', 'a', 'terrorist', 'mad
e', 'him', 'a', 'terrorist', '.', 'Islamically', 'brain', 'dead', '.']
```

In [9]:
```
tokenized_sent=sent_tokenize(texts[1])
tokenized_sent
```

Out[9]:
```
['@ShreyaBafna3 Now you idiots claim that people who tried to stop him from b
ecoming a terrorist made him a terrorist.',
 'Islamically brain dead.']
```

In [10]:
```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\user\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.
```

Out[10]:
```
True
```

In [11]:
```
from nltk.corpus import stopwords
stopwords=stopwords.words('english')
for i in tokenized_word:
  if i not in stopwords:
    print(i)
```

```
@
ShreyaBafna3
Now
idiots
claim
people
tried
stop
becoming
terrorist
made
terrorist
.
Islamically
brain
dead
.
```

In [12]:
```
print(stopwords)
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you'r
e", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves',
'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'i
t', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves',
'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those',
'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'ha
d', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but',
'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'wit
h', 'about', 'against', 'between', 'into', 'through', 'during', 'before', 'af
ter', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off',
'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when',
'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most',
'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'th
an', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'shoul
d', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren',
"aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn',
"hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'might
n', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'sh
ouldn', "shouldn't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'w
ouldn', "wouldn't"]
```

In [13]:

```
from nltk.stem import PorterStemmer
ps=PorterStemmer()
for i in tokenized_word:
  print(ps.stem(i))
```

```
@
shreyabafna3
now
you
idiot
claim
that
peopl
who
tri
to
stop
him
from
becom
a
terrorist
made
him
a
terrorist
.
islam
brain
dead
.
```

In [14]:

```
freqdist=nltk.FreqDist(tokenized_word)
for i,j in freqdist.items():
  print(f'{i}-----{j}')
```

```
@-----1
ShreyaBafna3-----1
Now-----1
you-----1
idiots-----1
claim-----1
that-----1
people-----1
who-----1
tried-----1
to-----1
stop-----1
him-----2
from-----1
becoming-----1
a-----2
terrorist-----2
made-----1
.-----2
Islamically-----1
brain-----1
dead-----1
```

In [15]:
```
print(list(nltk.bigrams(tokenized_word)))

print(list(nltk.trigrams(tokenized_word)))

print(list(nltk.ngrams(tokenized_word,4)))
```

```
[('@', 'ShreyaBafna3'), ('ShreyaBafna3', 'Now'), ('Now', 'you'), ('you', 'idi
ots'), ('idiots', 'claim'), ('claim', 'that'), ('that', 'people'), ('people',
'who'), ('who', 'tried'), ('tried', 'to'), ('to', 'stop'), ('stop', 'him'),
('him', 'from'), ('from', 'becoming'), ('becoming', 'a'), ('a', 'terrorist'),
('terrorist', 'made'), ('made', 'him'), ('him', 'a'), ('a', 'terrorist'), ('t
errorist', '.'), ('.', 'Islamically'), ('Islamically', 'brain'), ('brain', 'd
ead'), ('dead', '.')]
[('@', 'ShreyaBafna3', 'Now'), ('ShreyaBafna3', 'Now', 'you'), ('Now', 'you',
'idiots'), ('you', 'idiots', 'claim'), ('idiots', 'claim', 'that'), ('claim',
'that', 'people'), ('that', 'people', 'who'), ('people', 'who', 'tried'), ('w
ho', 'tried', 'to'), ('tried', 'to', 'stop'), ('to', 'stop', 'him'), ('stop',
'him', 'from'), ('him', 'from', 'becoming'), ('from', 'becoming', 'a'), ('bec
oming', 'a', 'terrorist'), ('a', 'terrorist', 'made'), ('terrorist', 'made',
'him'), ('made', 'him', 'a'), ('him', 'a', 'terrorist'), ('a', 'terrorist',
'.'), ('terrorist', '.', 'Islamically'), ('.', 'Islamically', 'brain'), ('Isl
amically', 'brain', 'dead'), ('brain', 'dead', '.')]
[('@', 'ShreyaBafna3', 'Now', 'you'), ('ShreyaBafna3', 'Now', 'you', 'idiot
s'), ('Now', 'you', 'idiots', 'claim'), ('you', 'idiots', 'claim', 'that'),
('idiots', 'claim', 'that', 'people'), ('claim', 'that', 'people', 'who'),
('that', 'people', 'who', 'tried'), ('people', 'who', 'tried', 'to'), ('who',
'tried', 'to', 'stop'), ('tried', 'to', 'stop', 'him'), ('to', 'stop', 'him',
'from'), ('stop', 'him', 'from', 'becoming'), ('him', 'from', 'becoming',
'a'), ('from', 'becoming', 'a', 'terrorist'), ('becoming', 'a', 'terrorist',
'made'), ('a', 'terrorist', 'made', 'him'), ('terrorist', 'made', 'him',
'a'), ('made', 'him', 'a', 'terrorist'), ('him', 'a', 'terrorist', '.'),
('a', 'terrorist', '.', 'Islamically'), ('terrorist', '.', 'Islamically', 'br
ain'), ('.', 'Islamically', 'brain', 'dead'), ('Islamically', 'brain', 'dea
d', '.')]
```

In [21]:
```
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\user\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

Out[21]:
```
True
```

In [23]:

```
from nltk.stem import WordNetLemmatizer
lem=WordNetLemmatizer()
for w in tokenized_word:
  print(lem.lemmatize(w,pos='v'))
```

```
---------------------------------------------------------------------------
LookupError                               Traceback (most recent call last)
File ~\anaconda3\lib\site-packages\nltk\corpus\util.py:84, in LazyCorpusLoa
der.__load(self)
     83 try:
---> 84     root = nltk.data.find(f"{self.subdir}/{zip_name}")
     85 except LookupError:

File ~\anaconda3\lib\site-packages\nltk\data.py:583, in find(resource_name,
paths)
    582 resource_not_found = f"\n{sep}\n{msg}\n{sep}\n"
--> 583 raise LookupError(resource_not_found)

LookupError: 
**********************************************************************
  Resource omw-1.4 not found.
  Please use the NLTK Downloader to obtain the resource:

  >>> import nltk
```

In [25]:

```
nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\user\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping taggers\averaged_perceptron_tagger.zip.
```

Out[25]:

```
True
```

In [26]:

```
from nltk.tag.perceptron import AveragedPerceptron

print(nltk.pos_tag(tokenized_word))
```

```
[('@', 'JJ'), ('ShreyaBafna3', 'NNP'), ('Now', 'RB'), ('you', 'PRP'), ('idiot
s', 'VBP'), ('claim', 'VB'), ('that', 'IN'), ('people', 'NNS'), ('who', 'W
P'), ('tried', 'VBD'), ('to', 'TO'), ('stop', 'VB'), ('him', 'PRP'), ('from',
'IN'), ('becoming', 'VBG'), ('a', 'DT'), ('terrorist', 'NN'), ('made', 'VB
D'), ('him', 'PRP'), ('a', 'DT'), ('terrorist', 'NN'), ('.', '.'), ('Islamica
lly', 'RB'), ('brain', 'NN'), ('dead', 'JJ'), ('.', '.')]
```

In [27]:

```
n_docs=len(corpus)
n_words_set=len(words_set)

df_tf=pd.DataFrame(np.zeros((n_docs,n_words_set)),columns=words_set)
for i in range(n_docs):
  words=corpus[i].split(' ')
  for w in words:
    df_tf[w][i]+(1/len(words))
df_tf
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[27], line 1
----> 1 n_docs=len(corpus)
      2 n_words_set=len(words_set)
      4 df_tf=pd.DataFrame(np.zeros((n_docs,n_words_set)),columns=words_set)

NameError: name 'corpus' is not defined
```