In [1]:
```
import os
import nltk
```

In [2]:
```
file = open('sample_text.txt')
raw_text=file.read()
file.close()
```

In [3]:
```
nltk.download('punkt')
```

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

Out[3]:
```
True
```

In [4]:
```
token_list=nltk.word_tokenize(raw_text)
print(token_list[0:20],"\n")
print("Total tokens: ",len(token_list))
```

```
['It', 'was', '7', 'minutes', 'after', 'midnight', '.', 'The', 'dog', 'was', 'lyin
g', 'on', 'the', 'grass', 'in', 'the', 'middle', 'of', 'the', 'lawn']

Total tokens:  306
```

In [5]:
```
from nltk.tokenize import punkt
token_list1 = list(filter(lambda token : punkt.PunktToken(token).is_non_punct,toke
print(token_list1[0:20],"\n")
print("Total tokens : ", len(token_list1))
```

```
['It', 'was', '7', 'minutes', 'after', 'midnight', 'The', 'dog', 'was', 'lying',
'on', 'the', 'grass', 'in', 'the', 'middle', 'of', 'the', 'lawn', 'in']

Total tokens :  277
```

In [6]:
```
nltk.download('stopwords')
```

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

Out[6]:
```
True
```

In [7]:
```
nltk.download('averaged_perceptron_tagger')
```

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

Out[7]:
```
True
```

In [8]:
```
from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))
for i in token_list1:
    wordsList = nltk.word_tokenize(i)
    wordsList = [w for w in wordsList if not w in stop_words]
    tagged = nltk.pos_tag(wordsList)
    print(tagged)
```

```
[('It', 'PRP')]
[]
[('7', 'CD')]
[('minutes', 'NNS')]
[]
[('midnight', 'NN')]
[('The', 'DT')]
[('dog', 'NN')]
[]
[('lying', 'VBG')]
[]
[]
[('grass', 'NN')]
[]
[]
[('middle', 'NN')]
[]
[]
[('lawn', 'NN')]
[]
[('front', 'NN')]
[]
[('Mrs', 'NN')]
[('Shearsâ€™', 'NN')]
[('house', 'NN')]
[('Its', 'PRP$')]
[('eyes', 'NNS')]
[]
[('closed', 'VBD')]
[('It', 'PRP')]
[('looked', 'VBD')]
[]
[]
[]
[]
[('running', 'VBG')]
[]
[]
[('side', 'NN')]
[]
[('way', 'NN')]
[('dogs', 'NNS')]
[('run', 'VB')]
[]
[]
[('think', 'NN')]
[]
[]
[('chasing', 'VBG')]
[]
[('cat', 'NN')]
[]
[]
[('dream', 'NN')]
[('But', 'CC')]
[]
[('dog', 'NN')]
[]
[]
[('running', 'VBG')]
[]
[('asleep', 'NN')]
[('The', 'DT')]
[('dog', 'NN')]
```

```
[]
[('dead', 'JJ')]
[('There', 'EX')]
[]
[]
[('garden', 'NN')]
[('fork', 'NN')]
[('sticking', 'VBG')]
[]
[]
[]
[('dog', 'NN')]
[('The', 'DT')]
[('points', 'NNS')]
[]
[]
[('fork', 'NN')]
[('must', 'MD')]
[]
[('gone', 'VBN')]
[]
[]
[('way', 'NN')]
[]
[]
[('dog', 'NN')]
[]
[]
[]
[('ground', 'NN')]
[]
[]
[('fork', 'NN')]
[]
[]
[('fallen', 'VBN')]
[]
[('I', 'PRP')]
[('decided', 'VBD')]
[]
[]
[('dog', 'NN')]
[]
[('probably', 'RB')]
[('killed', 'VBN')]
[]
[]
[('fork', 'NN')]
[]
[('I', 'PRP')]
[('could', 'MD')]
[]
[('see', 'VB')]
[]
[]
[('wounds', 'NNS')]
[]
[]
[('dog', 'NN')]
[]
[('I', 'PRP')]
[]
[]
[('think', 'NN')]
```

```
[]
[('would', 'MD')]
[('stick', 'NN')]
[]
[('garden', 'NN')]
[('fork', 'NN')]
[]
[]
[('dog', 'NN')]
[]
[]
[]
[('died', 'VBD')]
[]
[]
[]
[('reason', 'NN')]
[('like', 'IN')]
[('cancer', 'NN')]
[]
[('example', 'NN')]
[]
[]
[('road', 'NN')]
[('accident', 'NN')]
[('But', 'CC')]
[('I', 'PRP')]
[('could', 'MD')]
[]
[]
[('certain', 'JJ')]
[]
[]
[('I', 'PRP')]
[('went', 'VBD')]
[]
[('Mrs', 'NN')]
[('Shearsâ€™', 'NN')]
[('gate', 'NN')]
[('closing', 'NN')]
[]
[('behind', 'IN')]
[]
[('I', 'PRP')]
[('walked', 'VBD')]
[('onto', 'IN')]
[]
[('lawn', 'NN')]
[]
[('knelt', 'NN')]
[('beside', 'NN')]
[]
[('dog', 'NN')]
[('I', 'PRP')]
[('put', 'NN')]
[]
[('hand', 'NN')]
[]
[]
[('muzzle', 'NN')]
[]
[]
[('dog', 'NN')]
[('It', 'PRP')]
```

```
[]
[('still', 'RB')]
[('warm', 'NN')]
[('The', 'DT')]
[('dog', 'NN')]
[]
[('called', 'VBN')]
[('Wellington', 'NNP')]
[('It', 'PRP')]
[('belonged', 'VBN')]
[]
[('Mrs', 'NN')]
[('Shears', 'NNS')]
[]
[]
[]
[('friend', 'NN')]
[('She', 'PRP')]
[('lived', 'VBD')]
[]
[]
[('opposite', 'NN')]
[('side', 'NN')]
[]
[]
[('road', 'NN')]
[('two', 'CD')]
[('houses', 'NNS')]
[]
[]
[('left', 'NN')]
[('Wellington', 'NNP')]
[]
[]
[('poodle', 'NN')]
[('Not', 'RB')]
[('one', 'CD')]
[]
[]
[('small', 'JJ')]
[('poodles', 'NNS')]
[]
[]
[('hairstyles', 'NNS')]
[]
[]
[('big', 'JJ')]
[('poodle', 'NN')]
[('It', 'PRP')]
[]
[('curly', 'RB')]
[('black', 'JJ')]
[('fur', 'NN')]
[]
[]
[]
[('got', 'VBD')]
[('close', 'RB')]
[]
[('could', 'MD')]
[('see', 'VB')]
[]
[]
[('skin', 'NN')]
```

```
[('underneath', 'NN')]
[]
[('fur', 'NN')]
[]
[]
[]
[('pale', 'NN')]
[('yellow', 'NN')]
[('like', 'IN')]
[('chicken', 'NN')]
[('I', 'PRP')]
[('stroked', 'VBN')]
[('Wellington', 'NNP')]
[]
[('wondered', 'VBD')]
[]
[]
[('killed', 'VBN')]
[]
[]
[]
```

In [9]:
```
token_list2 = list(filter(lambda token: token not in stopwords.words("english"),tol
print(token_list2[0:20],"\n")
print("Total tokens : ", len(token_list2))
```

```
['It', '7', 'minutes', 'midnight', 'The', 'dog', 'lying', 'grass', 'middle', 'law
n', 'front', 'Mrs', 'Shearsâ€™', 'house', 'Its', 'eyes', 'closed', 'It', 'looked',
'running']

Total tokens :  147
```

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

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

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

In [11]:
```
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
token_list3 = [lemmatizer.lemmatize(word) for word in token_list2]
print(token_list3[0:20],"\n")
print("Total tokens : ", len(token_list3))
```

```
['It', '7', 'minute', 'midnight', 'The', 'dog', 'lying', 'grass', 'middle', 'law
n', 'front', 'Mrs', 'Shearsâ€™', 'house', 'Its', 'eye', 'closed', 'It', 'looked',
'running']

Total tokens :  147
```

In [12]:
```
from nltk.stem import PorterStemmer
ps = PorterStemmer()
```

In [13]:
```
token_list4 = []
for i in token_list2:
    token_list4.append(ps.stem(i))
```

In [15]:
```
print(token_list4[:20])
```

```
['it', '7', 'minut', 'midnight', 'the', 'dog', 'lie', 'grass', 'middl', 'lawn', 'f
ront', 'mr', 'shearsâ€™', 'hous', 'it', 'eye', 'close', 'it', 'look', 'run']
```

In [16]:
```
for i in token_list2:
    print(i,":", ps.stem(i))
```

```
It : it
7 : 7
minutes : minut
midnight : midnight
The : the
dog : dog
lying : lie
grass : grass
middle : middl
lawn : lawn
front : front
Mrs : mr
Shearsâ€™ : shearsâ€™
house : hous
Its : it
eyes : eye
closed : close
It : it
looked : look
running : run
side : side
way : way
dogs : dog
run : run
think : think
chasing : chase
cat : cat
dream : dream
But : but
dog : dog
running : run
asleep : asleep
The : the
dog : dog
dead : dead
There : there
garden : garden
fork : fork
sticking : stick
dog : dog
The : the
points : point
fork : fork
must : must
gone : gone
way : way
dog : dog
ground : ground
fork : fork
fallen : fallen
I : i
decided : decid
dog : dog
probably : probabl
killed : kill
fork : fork
I : i
could : could
see : see
wounds : wound
dog : dog
I : i
think : think
would : would
```

```
stick : stick
garden : garden
fork : fork
dog : dog
died : die
reason : reason
like : like
cancer : cancer
example : exampl
road : road
accident : accid
But : but
I : i
could : could
certain : certain
I : i
went : went
Mrs : mr
Shearsâ€™ : shearsâ€™
gate : gate
closing : close
behind : behind
I : i
walked : walk
onto : onto
lawn : lawn
knelt : knelt
beside : besid
dog : dog
I : i
put : put
hand : hand
muzzle : muzzl
dog : dog
It : it
still : still
warm : warm
The : the
dog : dog
called : call
Wellington : wellington
It : it
belonged : belong
Mrs : mr
Shears : shear
friend : friend
She : she
lived : live
opposite : opposit
side : side
road : road
two : two
houses : hous
left : left
Wellington : wellington
poodle : poodl
Not : not
one : one
small : small
poodles : poodl
hairstyles : hairstyl
big : big
poodle : poodl
It : it
```

```
curly : curli
black : black
fur : fur
got : got
close : close
could : could
see : see
skin : skin
underneath : underneath
fur : fur
pale : pale
yellow : yellow
like : like
chicken : chicken
I : i
stroked : stroke
Wellington : wellington
wondered : wonder
killed : kill
```

In [17]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [18]:
```
tr_idf_model  = TfidfVectorizer()
tf_idf_vector = tr_idf_model.fit_transform(token_list1)
```

In [19]:
```
print(type(tf_idf_vector), tf_idf_vector.shape)
```

```
<class 'scipy.sparse.csr.csr_matrix'> (277, 138)
```

In [20]:
```
tf_idf_array = tf_idf_vector.toarray()
print(tf_idf_array)
```

```
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
```

In [21]:
```
words_set = tr_idf_model.get_feature_names()
print(words_set)
```

```
['about', 'accident', 'after', 'all', 'and', 'any', 'are', 'as', 'asleep', 'be',
'because', 'behind', 'belonged', 'beside', 'big', 'black', 'but', 'called', 'cance
r', 'cat', 'certain', 'chasing', 'chicken', 'close', 'closed', 'closing', 'could',
'curly', 'dead', 'decided', 'died', 'do', 'dog', 'dogs', 'dream', 'example', 'eye
s', 'fallen', 'for', 'fork', 'friend', 'front', 'fur', 'garden', 'gate', 'gone',
'got', 'grass', 'ground', 'had', 'hairstyles', 'hand', 'have', 'her', 'him', 'hous
e', 'houses', 'if', 'in', 'into', 'it', 'its', 'killed', 'knelt', 'lawn', 'left',
'like', 'lived', 'looked', 'lying', 'me', 'middle', 'midnight', 'minutes', 'mrs',
'must', 'muzzle', 'my', 'not', 'of', 'on', 'one', 'onto', 'opposite', 'or', 'othe
r', 'our', 'out', 'over', 'pale', 'points', 'poodle', 'poodles', 'probably', 'pu
t', 'reason', 'road', 'run', 'running', 'see', 'she', 'shears', 'shearsâ', 'side',
'skin', 'small', 'some', 'stick', 'sticking', 'still', 'stroked', 'that', 'the',
'there', 'they', 'think', 'this', 'through', 'to', 'two', 'underneath', 'very', 'w
alked', 'warm', 'was', 'way', 'wellington', 'went', 'were', 'when', 'who', 'why',
'with', 'wondered', 'would', 'wounds', 'yellow', 'you']
```

```
C:\Users\HP\anaconda3\lib\site-packages\sklearn\utils\deprecation.py:87: FutureWar
ning: Function get_feature_names is deprecated; get_feature_names is deprecated in
1.0 and will be removed in 1.2. Please use get_feature_names_out instead.
  warnings.warn(msg, category=FutureWarning)
```

In [23]:
```
import pandas as pd
df_tf_idf = pd.DataFrame(tf_idf_array, columns = words_set)
df_tf_idf
```

Out[23]:

	about	accident	after	all	and	any	are	as	asleep	be	...	were	when	who	why	wi
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	(
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	(
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	(
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	(
4	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	(
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
272	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	(
273	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	(
274	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	(
275	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	(
276	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	1.0	(

277 rows × 138 columns

In [ ]: