I081 Aniruddh Kulkarni NLP Exp 2

May 28, 2023

1 Name: Aniruddh Kulkarni

2 Roll no: I081

3 Stream: CS (AI)

4 Division: I

5 Semester: 5th Semester

6 Batch: I-3

7 Subject: NLP

8 Assignment-2

[1]:

```
import nltk
import spacy
```

[2]:

```
import re
```

[3]:

```
# removing punctuation marks
import string
```

[4]:

```
# importing libraries
from nltk.tokenize import word_tokenize
from pprint import pprint # used for beautifying the print text
from nltk.corpus import stopwords
nltk.download('stopwords')
nltk.download('punkt')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/pushpakulkarni/nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     /Users/pushpakulkarni/nltk_data…
[nltk_data]   Package punkt is already up-to-date!
```

```
[4]: True
```

```
[5]: nltk.download('wordnet')
     nltk.download('omw-1.4')
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     /Users/pushpakulkarni/nltk_data…
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package omw-1.4 to
[nltk_data]     /Users/pushpakulkarni/nltk_data…
[nltk_data]   Package omw-1.4 is already up-to-date!
```

```
[5]: True
```

(a) Repeat the same tasks as above using SpaCy.
(b) Create a corpus of 35-45 lines. Perform all the basic NLP tasks on it.

##Solution: ###A)

```
[6]: # SPACY
     import spacy
     from spacy.lang.en.stop_words import STOP_WORDS
     import spacy
     spacy_model = spacy.load('en_core_web_sm')
```

```
[7]: # initialize the corpus
     corpus2 = "Need to finalize the demo corpus which will be used for this␣
      ↪notebook & should be done soon !!. It should be done by the ending of this␣
      ↪month. But will it? This notebook has been run 4 times !!"
     print(corpus2)
```

```
Need to finalize the demo corpus which will be used for this notebook & should
be done soon !!. It should be done by the ending of this month. But will it?
This notebook has been run 4 times !!
```

```
[8]: corpus_lower_2 = corpus2.lower()
     print(corpus_lower_2)
```

```
need to finalize the demo corpus which will be used for this notebook & should
be done soon !!. it should be done by the ending of this month. but will it?
this notebook has been run 4 times !!
```

```
[9]: # substituting digit by space
     corpus_new_2 = re.sub(r'\d+', ' ', corpus_lower_2)
     print(corpus_new_2)
```

```
need to finalize the demo corpus which will be used for this notebook & should
be done soon !!. it should be done by the ending of this month. but will it?
this notebook has been run   times !!
```

[11]:
```
corpus_new_2 = corpus_new_2.translate(str.maketrans("", "", string.punctuation))
print(corpus_new_2)
```

```
need to finalize the demo corpus which will be used for this notebook  should be
done soon  it should be done by the ending of this month but will it this
notebook has been run   times
```

[12]:
```
# removing white spaces
corpus_new_2 = ' '.join([token for token in corpus_new_2.split()])
print(corpus_new_2)
```

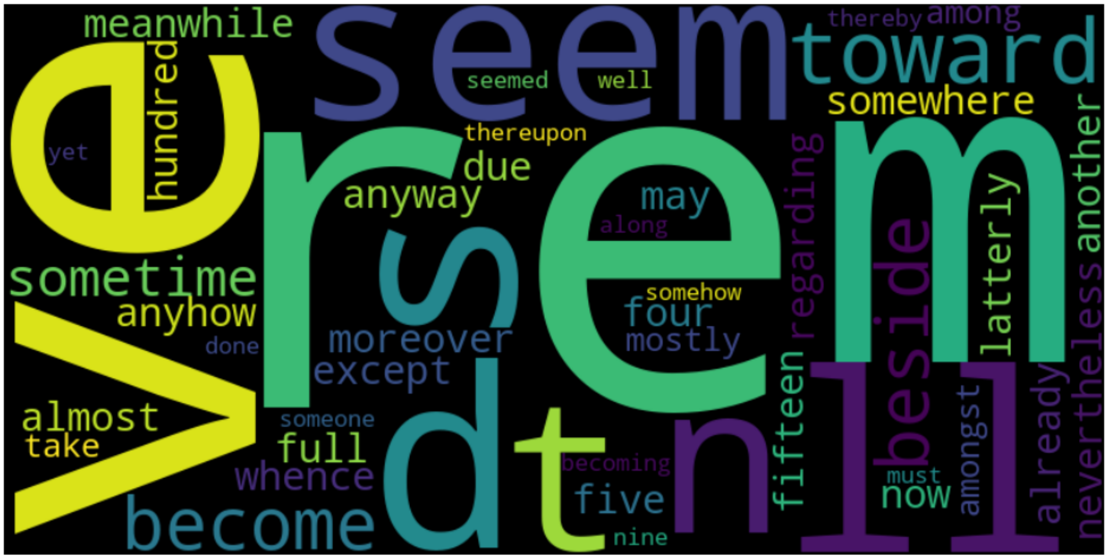
```
need to finalize the demo corpus which will be used for this notebook should be
done soon it should be done by the ending of this month but will it this
notebook has been run times
```

[13]:
```
# importing libraries
all_stopwords = spacy_model.Defaults.stop_words
#all_stopwords.remove('not')
print(all_stopwords)
```

```
{'four', 'itself', 'almost', 'except', 'anyway', 'meanwhile', 'seem', 'whence',
'now', 'somewhere', 'latterly', 'regarding', 'moreover', 'already',
'nevertheless', 'so', 'full', 'anyhow', '’m', 'becomes', 'may', 'due', 'while',
'another', 'such', 'fifteen', 'hundred', 'did', 'five', 'is', 'mostly', 'among',
'amongst', 'take', 'yourselves', "'ve", 'when', 'thereupon', 'if', 'very',
'somehow', 'his', '’ll', 'sometimes', 'here', 'becoming', 'must', 'well', 'are',
"'d", 'me', 'seemed', 'the', 'done', 'she', 'this', 'yet', 'someone', '’s',
'off', 'nine', 'along', 'thereby', 'keep', '’re', 'on', 'be', 'own', 'out',
'alone', 'elsewhere', 'others', 'however', 'hers', 'does', 'might', 'we',
'whither', 'hereafter', 'anything', 'nowhere', 'some', 'enough', 'neither',
'together', 'n’t', 'both', 'am', 'twelve', 'would', 'only', 'into', 'over',
'upon', 'than', 'seeming', 'himself', 'something', 'who', 'without', 'mine',
'their', 'herein', 'at', 'say', 'besides', 'anyone', 'not', 'whereas', '’ve',
'afterwards', 'empty', 'my', 'really', 'nobody', 'more', 'twenty', 'seems',
'everything', 'front', "'s", 'whoever', 'can', 'call', 'still', 'every',
'nothing', 'top', 'whereby', 'its', 'as', 'whereupon', 'been', 'amount', 'they',
'thus', 'or', 'used', 'with', 'per', "'ll", 'made', 'whether', 'could', 'noone',
'whenever', 'toward', 'one', 'above', 'former', 'for', 'being', 'before',
'down', 'though', 'last', 'else', 'doing', 'everywhere', 'what', 'throughout',
'these', 'whole', 'across', 'formerly', 'just', 'will', 'again', "'re", 'an',
're', 'n‘t', "n't", 'back', '‘m', 'until', 'but', 'often', 'to', 'of',
'serious', 'behind', 'using', 'all', 'six', 'yourself', 'beforehand', 'quite',
'that', '’d', 'name', 'a', 'i', 'wherein', 'thru', 'also', 'indeed', 'next',
"'m", 'although', 'thereafter', 'either', 'otherwise', 'become', 'latter', 'he',
'then', 'whom', 'whose', '‘ve', 'eight', '‘s', 'see', 'yours', 'you', 'was',
'hereupon', 'after', 'any', 'hence', 'each', 'unless', 'has', 'same', 'our',
'less', 'other', 'rather', 'around', 'forty', 'ca', 'anywhere', '‘d', 'show',
'no', 'none', 'beside', 'it', 'through', 'have', 'go', 'many', 'during',
'which', 'third', 'too', 'why', 'whatever', 'themselves', 'ever', 'ourselves',
'and', 'via', 'under', 'therein', 'herself', 'thence', 'make', 'put', 'where',
'myself', 'ten', 'various', 'bottom', 'side', 'below', 'several', 'were',
'first', 'about', 'least', 'perhaps', 'onto', 'since', 'towards', 'up', '‘ll',
'those', 'them', 'had', 'everyone', 'much', 'three', 'by', 'move', 'always',
'get', 'hereby', '‘re', 'us', 'became', 'your', 'give', 'nor', 'him', 'in',
'there', 'wherever', 'whereafter', 'sixty', 'further', 'few', 'between',
'within', 'once', 'please', 'ours', 'against', 'because', 'two', 'therefore',
'namely', 'should', 'eleven', 'her', 'beyond', 'most', 'never', 'cannot', 'how',
'do', 'from', 'sometime', 'part', 'even', 'fifty'}
```

[14]:

```
# make word cloud of stopwords
from wordcloud import WordCloud
import matplotlib.pyplot as plt
#WordCloud
s = ' '.join(all_stopwords)
wc = WordCloud(width=800, height=400, max_words=50).generate(s)
plt.figure(figsize=(12,10))
plt.imshow(wc, interpolation="bilinear")
plt.axis("off")
plt.show()
```



[15]:

```
text_tokens = word_tokenize(corpus_new_2)
tokens_without_sw = [word for word in text_tokens if not word in all_stopwords]
print(tokens_without_sw)
```

```
['need', 'finalize', 'demo', 'corpus', 'notebook', 'soon', 'ending', 'month',
'notebook', 'run', 'times']
```

[16]:

```
#Stemming not possible in SpaCy
#Lemmitization
tokens_without_sw=str(tokens_without_sw)
tokens_without_sw=spacy_model(tokens_without_sw)
for word in tokens_without_sw:
    print(word.text,  word.lemma_,end=" ")
```

```
[ [ ' ' need need ' ' , , ' ' finalize finalize ' ' , , ' ' demo demo ' ' , , '
' corpus corpus ' ' , , ' ' notebook notebook ' ' , , ' ' soon soon ' ' , , ' '
ending end ' ' , , ' ' month month ' ' , , ' ' notebook notebook ' ' , , ' ' run
run ' ' , , ' ' times time ' ' ] ]
```

[17]:

```
#SpaCy POS

corpus2=spacy_model(corpus2)

for token in corpus2:
  print(token, token.pos_)
```

```
Need VERB
to PART
finalize VERB
the DET
demo NOUN
corpus NOUN
which PRON
will AUX
be AUX
used VERB
for ADP
this DET
notebook NOUN
& CCONJ
should AUX
be AUX
done VERB
soon ADV
! PUNCT
! PUNCT
. PUNCT
It PRON
should AUX
be AUX
done VERB
by ADP
the DET
ending NOUN
of ADP
```

```
this DET
month NOUN
. PUNCT
But CCONJ
will AUX
it PRON
? PUNCT
This DET
notebook NOUN
has AUX
been AUX
run VERB
4 NUM
times NOUN
! PUNCT
! PUNCT
```

###B)

[18]:
```python
# initialize the corpus
corpus3 = "1.She doesn’t study German on Monday.2.Does she live in Paris?3.He doesn’t teach math.4.Cats hate water.5.Every child likes an ice cream.6.My brother takes out the trash.7.The course starts next Sunday.8.She swims every morning.9.I don’t wash the dishes.10.We see them every week.11.I don’t like tea.12.When does the train usually leave?13.She always forgets her purse.14.You don’t have children.15.I and my sister don’t see each other anymore.16.They don’t go to school tomorrow.17.He loves to play basketball.18.He goes to school.19.The Earth is spherical.20.Julie talks very fast.21.My brother’s dog barks a lot.22.Does he play tennis?23.The train leaves every morning at 18 AM.24.Water freezes at 0°C25.I love my new pets.26.We drink coffee every morning.27.My Dad never works on the weekends.28.She doesn’t teach chemistry.29.I do love my new pets.30.Mary brushes her teeth twice a day.31.He drives to work.32.Mary enjoys cooking.33.She likes bananas.34.My mother never lies.35.You don’t listen to me.36.I run four miles every morning.37.They speak English at work.38.The train does not leave at 12 AM.39.I have no money at the moment.40.Do they talk a lot?"
print(corpus3)
```

```
1.She doesn’t study German on Monday.2.Does she live in Paris?3.He doesn’t teach
math.4.Cats hate water.5.Every child likes an ice cream.6.My brother takes out
the trash.7.The course starts next Sunday.8.She swims every morning.9.I don’t
wash the dishes.10.We see them every week.11.I don’t like tea.12.When does the
train usually leave?13.She always forgets her purse.14.You don’t have
children.15.I and my sister don’t see each other anymore.16.They don’t go to
school tomorrow.17.He loves to play basketball.18.He goes to school.19.The Earth
is spherical.20.Julie talks very fast.21.My brother’s dog barks a lot.22.Does he
play tennis?23.The train leaves every morning at 18 AM.24.Water freezes at
0°C25.I love my new pets.26.We drink coffee every morning.27.My Dad never works
```

```
on the weekends.28.She doesn’t teach chemistry.29.I do love my new pets.30.Mary
brushes her teeth twice a day.31.He drives to work.32.Mary enjoys cooking.33.She
likes bananas.34.My mother never lies.35.You don’t listen to me.36.I run four
miles every morning.37.They speak English at work.38.The train does not leave at
12 AM.39.I have no money at the moment.40.Do they talk a lot?
```

[19]:

```
corpus_lower_3 = corpus3.lower()
print(corpus_lower_3)
```

```
1.she doesn’t study german on monday.2.does she live in paris?3.he doesn’t teach
math.4.cats hate water.5.every child likes an ice cream.6.my brother takes out
the trash.7.the course starts next sunday.8.she swims every morning.9.i don’t
wash the dishes.10.we see them every week.11.i don’t like tea.12.when does the
train usually leave?13.she always forgets her purse.14.you don’t have
children.15.i and my sister don’t see each other anymore.16.they don’t go to
school tomorrow.17.he loves to play basketball.18.he goes to school.19.the earth
is spherical.20.julie talks very fast.21.my brother’s dog barks a lot.22.does he
play tennis?23.the train leaves every morning at 18 am.24.water freezes at
0°c25.i love my new pets.26.we drink coffee every morning.27.my dad never works
on the weekends.28.she doesn’t teach chemistry.29.i do love my new pets.30.mary
brushes her teeth twice a day.31.he drives to work.32.mary enjoys cooking.33.she
likes bananas.34.my mother never lies.35.you don’t listen to me.36.i run four
miles every morning.37.they speak english at work.38.the train does not leave at
12 am.39.i have no money at the moment.40.do they talk a lot?
```

[20]:

```
corpus_new_3 = re.sub(r'\d+', ' ', corpus_lower_3)
print(corpus_new_3)
```

```
 .she doesn’t study german on monday. .does she live in paris? .he doesn’t teach
math. .cats hate water. .every child likes an ice cream. .my brother takes out
the trash. .the course starts next sunday. .she swims every morning. .i don’t
wash the dishes. .we see them every week. .i don’t like tea. .when does the
train usually leave? .she always forgets her purse. .you don’t have children. .i
and my sister don’t see each other anymore. .they don’t go to school tomorrow.
.he loves to play basketball. .he goes to school. .the earth is spherical.
.julie talks very fast. .my brother’s dog barks a lot. .does he play tennis?
.the train leaves every morning at   am. .water freezes at  °c .i love my new
pets. .we drink coffee every morning. .my dad never works on the weekends. .she
doesn’t teach chemistry. .i do love my new pets. .mary brushes her teeth twice a
day. .he drives to work. .mary enjoys cooking. .she likes bananas. .my mother
never lies. .you don’t listen to me. .i run four miles every morning. .they
speak english at work. .the train does not leave at   am. .i have no money at
the moment. .do they talk a lot?
```

[21]:

```
corpus_new_3 = corpus_new_3.translate(str.maketrans("", "", string.punctuation))
print(corpus_new_3)
```

```
 she doesn’t study german on monday does she live in paris he doesn’t teach math
```

```
cats hate water every child likes an ice cream my brother takes out the trash
the course starts next sunday she swims every morning i don’t wash the dishes we
see them every week i don’t like tea when does the train usually leave she
always forgets her purse you don’t have children i and my sister don’t see each
other anymore they don’t go to school tomorrow he loves to play basketball he
goes to school the earth is spherical julie talks very fast my brother’s dog
barks a lot does he play tennis the train leaves every morning at   am water
freezes at  °c i love my new pets we drink coffee every morning my dad never
works on the weekends she doesn’t teach chemistry i do love my new pets mary
brushes her teeth twice a day he drives to work mary enjoys cooking she likes
bananas my mother never lies you don’t listen to me i run four miles every
morning they speak english at work the train does not leave at   am i have no
money at the moment do they talk a lot
```

```
[22]: corpus_new_3 = ' '.join([token for token in corpus_new_3.split()])
      print(corpus_new_3)
```

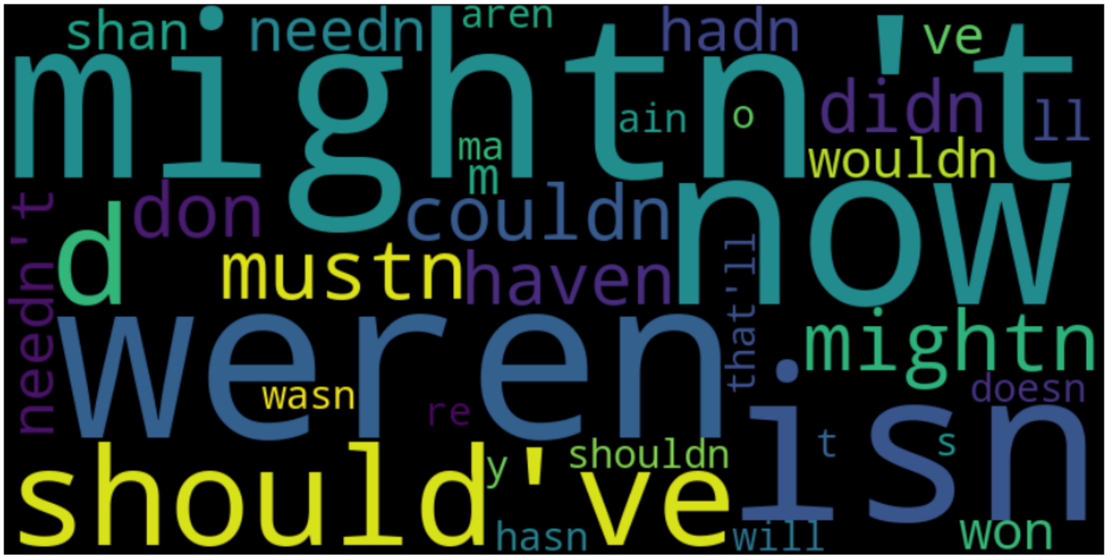
```
she doesn’t study german on monday does she live in paris he doesn’t teach math
cats hate water every child likes an ice cream my brother takes out the trash
the course starts next sunday she swims every morning i don’t wash the dishes we
see them every week i don’t like tea when does the train usually leave she
always forgets her purse you don’t have children i and my sister don’t see each
other anymore they don’t go to school tomorrow he loves to play basketball he
goes to school the earth is spherical julie talks very fast my brother’s dog
barks a lot does he play tennis the train leaves every morning at am water
freezes at °c i love my new pets we drink coffee every morning my dad never
works on the weekends she doesn’t teach chemistry i do love my new pets mary
brushes her teeth twice a day he drives to work mary enjoys cooking she likes
bananas my mother never lies you don’t listen to me i run four miles every
morning they speak english at work the train does not leave at am i have no
money at the moment do they talk a lot
```

```
[23]: stop_words_nltk = set(stopwords.words('english'))
      print(stop_words_nltk)
```

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

[24]:

```python
# make word cloud of stopwords
from wordcloud import WordCloud
import matplotlib.pyplot as plt
#WordCloud
s = ' '.join(stop_words_nltk)
wc = WordCloud(width=800, height=400, max_words=50).generate(s)
plt.figure(figsize=(12,10))
plt.imshow(wc, interpolation="bilinear")
plt.axis("off")
plt.show()
```



[25]:

```python
# tokenize the corpus
tokenized_corpus_nltk_3 = word_tokenize(corpus_new_3)
print(tokenized_corpus_nltk_3)
```

```
['she', 'doesn', '’', 't', 'study', 'german', 'on', 'monday', 'does', 'she',
'live', 'in', 'paris', 'he', 'doesn', '’', 't', 'teach', 'math', 'cats', 'hate',
'water', 'every', 'child', 'likes', 'an', 'ice', 'cream', 'my', 'brother',
'takes', 'out', 'the', 'trash', 'the', 'course', 'starts', 'next', 'sunday',
'she', 'swims', 'every', 'morning', 'i', 'don', '’', 't', 'wash', 'the',
'dishes', 'we', 'see', 'them', 'every', 'week', 'i', 'don', '’', 't', 'like',
'tea', 'when', 'does', 'the', 'train', 'usually', 'leave', 'she', 'always',
'forgets', 'her', 'purse', 'you', 'don', '’', 't', 'have', 'children', 'i',
'and', 'my', 'sister', 'don', '’', 't', 'see', 'each', 'other', 'anymore',
'they', 'don', '’', 't', 'go', 'to', 'school', 'tomorrow', 'he', 'loves', 'to',
'play', 'basketball', 'he', 'goes', 'to', 'school', 'the', 'earth', 'is',
'spherical', 'julie', 'talks', 'very', 'fast', 'my', 'brother', '’', 's', 'dog',
'barks', 'a', 'lot', 'does', 'he', 'play', 'tennis', 'the', 'train', 'leaves',
'every', 'morning', 'at', 'am', 'water', 'freezes', 'at', '°c', 'i', 'love',
'my', 'new', 'pets', 'we', 'drink', 'coffee', 'every', 'morning', 'my', 'dad',
'never', 'works', 'on', 'the', 'weekends', 'she', 'doesn', '’', 't', 'teach',
'chemistry', 'i', 'do', 'love', 'my', 'new', 'pets', 'mary', 'brushes', 'her',
'teeth', 'twice', 'a', 'day', 'he', 'drives', 'to', 'work', 'mary', 'enjoys',
'cooking', 'she', 'likes', 'bananas', 'my', 'mother', 'never', 'lies', 'you',
'don', '’', 't', 'listen', 'to', 'me', 'i', 'run', 'four', 'miles', 'every',
'morning', 'they', 'speak', 'english', 'at', 'work', 'the', 'train', 'does',
'not', 'leave', 'at', 'am', 'i', 'have', 'no', 'money', 'at', 'the', 'moment',
'do', 'they', 'talk', 'a', 'lot']
```

[26]:

```python
# stopword removal
tokenized_corpus_without_stopwords_3 = [i for i in tokenized_corpus_nltk_3 if␣
 ↪not i in stop_words_nltk]
print("Tokenized corpus without stopwords:
 ↪",tokenized_corpus_without_stopwords_3)
```

```
Tokenized corpus without stopwords: ['’', 'study', 'german', 'monday', 'live',
'paris', '’', 'teach', 'math', 'cats', 'hate', 'water', 'every', 'child',
'likes', 'ice', 'cream', 'brother', 'takes', 'trash', 'course', 'starts',
'next', 'sunday', 'swims', 'every', 'morning', '’', 'wash', 'dishes', 'see',
'every', 'week', '’', 'like', 'tea', 'train', 'usually', 'leave', 'always',
'forgets', 'purse', '’', 'children', 'sister', '’', 'see', 'anymore', '’', 'go',
'school', 'tomorrow', 'loves', 'play', 'basketball', 'goes', 'school', 'earth',
'spherical', 'julie', 'talks', 'fast', 'brother', '’', 'dog', 'barks', 'lot',
'play', 'tennis', 'train', 'leaves', 'every', 'morning', 'water', 'freezes',
'°c', 'love', 'new', 'pets', 'drink', 'coffee', 'every', 'morning', 'dad',
'never', 'works', 'weekends', '’', 'teach', 'chemistry', 'love', 'new', 'pets',
'mary', 'brushes', 'teeth', 'twice', 'day', 'drives', 'work', 'mary', 'enjoys',
'cooking', 'likes', 'bananas', 'mother', 'never', 'lies', '’', 'listen', 'run',
'four', 'miles', 'every', 'morning', 'speak', 'english', 'work', 'train',
'leave', 'money', 'moment', 'talk', 'lot']
```

[28]:

```python
#Stemming
from nltk.stem import PorterStemmer
stemmer3= PorterStemmer()
print("Before Stemming:")
print(corpus_new_3)
print("After Stemming:")
for word in tokenized_corpus_without_stopwords_3:
  print(stemmer3.stem(word),end=" ")
```

```
Before Stemming:
she doesn’t study german on monday does she live in paris he doesn’t teach math
cats hate water every child likes an ice cream my brother takes out the trash
the course starts next sunday she swims every morning i don’t wash the dishes we
see them every week i don’t like tea when does the train usually leave she
always forgets her purse you don’t have children i and my sister don’t see each
other anymore they don’t go to school tomorrow he loves to play basketball he
goes to school the earth is spherical julie talks very fast my brother’s dog
barks a lot does he play tennis the train leaves every morning at am water
freezes at °c i love my new pets we drink coffee every morning my dad never
works on the weekends she doesn’t teach chemistry i do love my new pets mary
brushes her teeth twice a day he drives to work mary enjoys cooking she likes
bananas my mother never lies you don’t listen to me i run four miles every
morning they speak english at work the train does not leave at am i have no
money at the moment do they talk a lot
After Stemming:
’ studi german monday live pari ’ teach math cat hate water everi child like ice
cream brother take trash cours start next sunday swim everi morn ’ wash dish see
everi week ’ like tea train usual leav alway forget purs ’ children sister ’ see
anymor ’ go school tomorrow love play basketbal goe school earth spheric juli
talk fast brother ’ dog bark lot play tenni train leav everi morn water freez °c
love new pet drink coffe everi morn dad never work weekend ’ teach chemistri
love new pet mari brush teeth twice day drive work mari enjoy cook like banana
mother never lie ’ listen run four mile everi morn speak english work train leav
money moment talk lot
```

[30]:
```
# lemetizing the text
from nltk.stem import WordNetLemmatizer
print("After Lemetization:")
lemm = WordNetLemmatizer()
for word in tokenized_corpus_without_stopwords_3:
  print(lemm.lemmatize(word),end=" ")
```

```
After Lemetization:
’ study german monday live paris ’ teach math cat hate water every child like
ice cream brother take trash course start next sunday swim every morning ’ wash
dish see every week ’ like tea train usually leave always forgets purse ’ child
sister ’ see anymore ’ go school tomorrow love play basketball go school earth
spherical julie talk fast brother ’ dog bark lot play tennis train leaf every
morning water freeze °c love new pet drink coffee every morning dad never work
weekend ’ teach chemistry love new pet mary brush teeth twice day drive work
mary enjoys cooking like banana mother never lie ’ listen run four mile every
morning speak english work train leave money moment talk lot
```

[31]:

```
# POS tagging
print("POS Tagging using NLTK:")
pprint(nltk.pos_tag(word_tokenize(corpus3)))
```

```
POS Tagging using NLTK:
[('1.She', 'CD'),
 ('doesn', 'JJ'),
 ('’', 'NNP'),
 ('t', 'NN'),
 ('study', 'JJ'),
 ('German', 'NNP'),
 ('on', 'IN'),
 ('Monday.2.Does', 'NNP'),
 ('she', 'PRP'),
 ('live', 'VBP'),
 ('in', 'IN'),
 ('Paris', 'NNP'),
 ('?', '.'),
 ('3.He', 'CD'),
 ('doesn', 'NN'),
 ('’', 'NNP'),
 ('t', 'VBZ'),
 ('teach', 'NN'),
 ('math.4.Cats', 'NNS'),
 ('hate', 'VBP'),
 ('water.5.Every', 'JJ'),
 ('child', 'NN'),
 ('likes', 'VBZ'),
 ('an', 'DT'),
 ('ice', 'NN'),
 ('cream.6.My', 'NN'),
 ('brother', 'NN'),
 ('takes', 'VBZ'),
 ('out', 'RP'),
 ('the', 'DT'),
 ('trash.7.The', 'NN'),
 ('course', 'NN'),
 ('starts', 'VBZ'),
 ('next', 'JJ'),
 ('Sunday.8.She', 'NNP'),
 ('swims', 'VBZ'),
 ('every', 'DT'),
 ('morning.9.I', 'NN'),
 ('don', 'NN'),
 ('’', 'NNP'),
 ('t', 'NN'),
 ('wash', 'VBD'),
 ('the', 'DT'),
 ('dishes.10.We', 'NN'),
 ('see', 'VBP'),
 ('them', 'PRP'),
 ('every', 'DT'),
 ('week.11.I', 'NN'),
 ('don', 'NN'),
 ('’', 'NNP'),
 ('t', 'NN'),
 ('like', 'IN'),
 ('tea.12.When', 'NN'),
 ('does', 'VBZ'),
 ('the', 'DT'),
 ('train', 'NN'),
 ('usually', 'RB'),
 ('leave', 'VB'),
 ('?', '.'),
 ('13.She', 'CD'),
 ('always', 'RB'),
 ('forgets', 'VBZ'),
 ('her', 'PRP$'),
 ('purse.14.You', 'JJ'),
 ('don', 'NN'),
 ('’', 'NN'),
 ('t', 'NN'),
 ('have', 'VBP'),
 ('children.15.I', 'VBN'),
 ('and', 'CC'),
 ('my', 'PRP$'),
 ('sister', 'JJ'),
 ('don', 'NN'),
 ('’', 'NN'),
 ('t', 'NN'),
 ('see', 'VBP'),
 ('each', 'DT'),
 ('other', 'JJ'),
 ('anymore.16.They', 'JJ'),
 ('don', 'NNS'),
 ('’', 'VBP'),
 ('t', 'RB'),
 ('go', 'VBP'),
 ('to', 'TO'),
 ('school', 'NN'),
 ('tomorrow.17.He', 'NN'),
 ('loves', 'NNS'),
 ('to', 'TO'),
 ('play', 'VB'),
 ('basketball.18.He', 'NN'),
 ('goes', 'VBZ'),
 ('to', 'TO'),
 ('school.19.The', 'VB'),
 ('Earth', 'NNP'),
 ('is', 'VBZ'),
 ('spherical.20.Julie', 'JJ'),
 ('talks', 'NNS'),
 ('very', 'RB'),
 ('fast.21.My', 'RB'),
 ('brother', 'NN'),
 ('’', 'JJ'),
 ('s', 'NN'),
 ('dog', 'NN'),
 ('barks', 'VBZ'),
 ('a', 'DT'),
 ('lot.22.Does', 'NN'),
 ('he', 'PRP'),
 ('play', 'VB'),
 ('tennis', 'NN'),
 ('?', '.'),
 ('23.The', 'CD'),
 ('train', 'NN'),
 ('leaves', 'NNS'),
 ('every', 'DT'),
 ('morning', 'NN'),
 ('at', 'IN'),
 ('18', 'CD'),
 ('AM.24.Water', 'NNP'),
 ('freezes', 'NNS'),
 ('at', 'IN'),
 ('0°C25.I', 'CD'),
 ('love', 'NN'),
 ('my', 'PRP$'),
 ('new', 'JJ'),
 ('pets.26.We', 'NN'),
 ('drink', 'VBP'),
 ('coffee', 'NN'),
 ('every', 'DT'),
 ('morning.27.My', 'NN'),
 ('Dad', 'NNP'),
 ('never', 'RB'),
 ('works', 'VBZ'),
 ('on', 'IN'),
 ('the', 'DT'),
 ('weekends.28.She', 'NN'),
 ('doesn', 'NN'),
 ('’', 'NNP'),
 ('t', 'VBZ'),
 ('teach', 'VB'),
 ('chemistry.29.I', 'NN'),
 ('do', 'VBP'),
 ('love', 'VB'),
 ('my', 'PRP$'),
 ('new', 'JJ'),
 ('pets.30.Mary', 'JJ'),
 ('brushes', 'NNS'),
 ('her', 'PRP$'),
 ('teeth', 'NNS'),
 ('twice', 'RB'),
 ('a', 'DT'),
 ('day.31.He', 'JJ'),
 ('drives', 'NNS'),
 ('to', 'TO'),
 ('work.32.Mary', 'JJ'),
 ('enjoys', 'NNS'),
 ('cooking.33.She', 'VBP'),
 ('likes', 'VBZ'),
 ('bananas.34.My', 'NN'),
 ('mother', 'NN'),
 ('never', 'RB'),
 ('lies.35.You', 'VBZ'),
 ('don', 'JJ'),
 ('’', 'NNP'),
 ('t', 'NN'),
 ('listen', 'NN'),
 ('to', 'TO'),
 ('me.36.I', 'VB'),
 ('run', 'VB'),
 ('four', 'CD'),
 ('miles', 'NNS'),
 ('every', 'DT'),
 ('morning.37.They', 'NN'),
 ('speak', 'VBP'),
 ('English', 'NNP'),
 ('at', 'IN'),
 ('work.38.The', 'JJ'),
 ('train', 'NN'),
 ('does', 'VBZ'),
 ('not', 'RB'),
 ('leave', 'VB'),
 ('at', 'IN'),
 ('12', 'CD'),
 ('AM.39.I', 'NNP'),
 ('have', 'VBP'),
 ('no', 'DT'),
 ('money', 'NN'),
 ('at', 'IN'),
 ('the', 'DT'),
 ('moment.40.Do', 'NN'),
 ('they', 'PRP'),
 ('talk', 'VBP'),
 ('a', 'DT'),
 ('lot', 'NN'),
 ('?', '.')]
```

#Conclusion: 1) Explored various methods of language pre processing and tokenizing.
2) Stemming is not possible using nltk.
3) Snowball stemmer is better than porter stemmer in majority of cases.