AI LAB
EXPERIMENT NO: 11
Implementation of NLP – Cleaning Text

WORKING PRINCIPLE:-

In natural language processing, human language is separated into fragments so that the grammatical structure of sentences and the meaning of words can be analyzed and understood in context.

Let's see the various different steps that are followed while preprocessing the data also used for dimensionality reduction.

1. **Tokenization**
2. **Lower casing**
3. **Stop words removal**
4. **Stemming**
5. **Lemmatization**

Each term is the axis in the vector space model. In muti-dimensional space, the text or document are constituted as vectors. The number of different words represents the number of dimensions.

The python library that is used to do the preprocessing tasks in nlp is nltk. You can install the nltk package using ***"pip install nltk"***.

1. Tokenization:

It is a method in which sentences are converted into words.

```
import nltk
from nltk.tokenize import word_tokenize
token = word_tokenize("My Email address is: taneshbalodi8@gmail.com")
token
```

```
['My', 'Email', 'address', 'is', ':', 'taneshbalodi8', '@', 'gmail.com']
```

Tokenization

(Read also: Sentiment Analysis of YouTube Comments)

2. Lowercasing:

the tokenized words into lower case format. (NLU -> nlu). Words having the same meaning like nlp and NLP if they are not converted into lowercase then these both will constitute as non-identical words in the vector space model.

```
Lowercase = []
for lowercase in token:
    Lowercase.append(lowercase.lower())
Lowercase
```

```
['my', 'email', 'address', 'is', ':', 'taneshbalodi8', '@', 'gmail.com']
```

Lowercasing

---

3. Stop words removal:

These are the most often used that do not have any significance while determining the two different documents like (a, an, the, etc.) so they are to be removed. Check the below image wherefrom the sentence "**Introduction to Natural Language Processing**" the "**to**" word is removed.

```
from nltk.corpus import stopwords
stop_words = stopwords.words('english')
from string import punctuation
punct = list(punctuation)
print(dataset[1]['quote'])
tokens = word_tokenize(dataset[1]['quote'])
len(tokens)
```

```
I'm selfish, impatient and a little insecure. I make mistakes, I am out of control and at times hard to handle. But if you ca
n't handle me at my worst, then you sure as hell don't deserve me at my best.

50
```

Without removing Stopwords

We got to see 50 tokens without removing stopwords, Now we shall remove stopwords.

```
cleaned_tokens = [token for token in tokens if token not in stop_words
                  and token not in punctuation]
len(cleaned_tokens)
```

By cleaning the stopwords we got the length of the dataset as 24.

(Referred blog: What is SqueezeBERT in NLP?)

4. Stemming:

It is the process in which the words are converted to its base from. Check the below code implementation where the words of the sentence are converted to the base form.

```
from nltk.stem import PorterStemmer
ps = PorterStemmer()
print(ps.stem('jumping'))
print(ps.stem('lately'))
print(ps.stem('assess'))
print(ps.stem('ran'))
```

```
jump
late
assess
ran
```

Stemming

---

5. Lemmatization:

Different from stemming, lemmatization lowers the words to word in the present language for example check the below image where word has and is are changed to ha and be respectively.

```
from nltk import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
print(lemmatizer.lemmatize('ran', 'v'))
print(lemmatizer.lemmatize('better', 'a'))
```

```
run
good
```

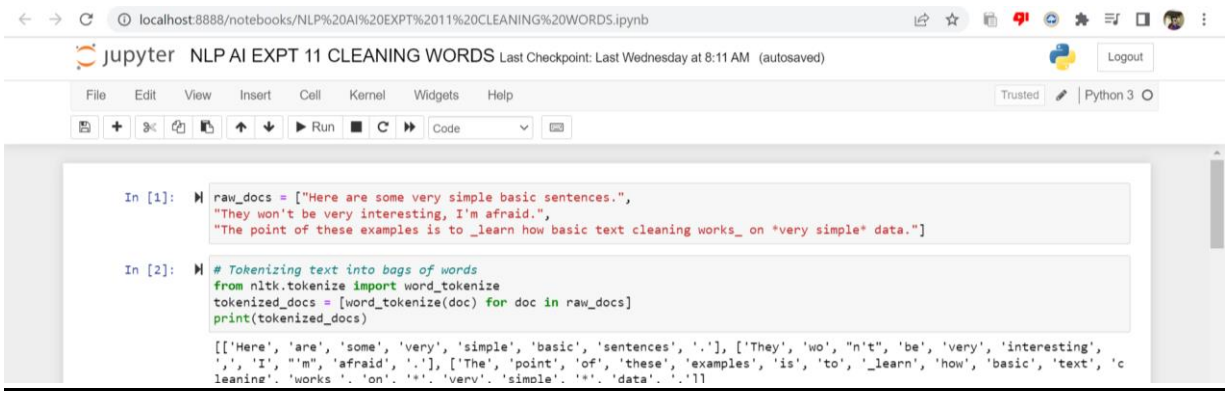
Lemmatization

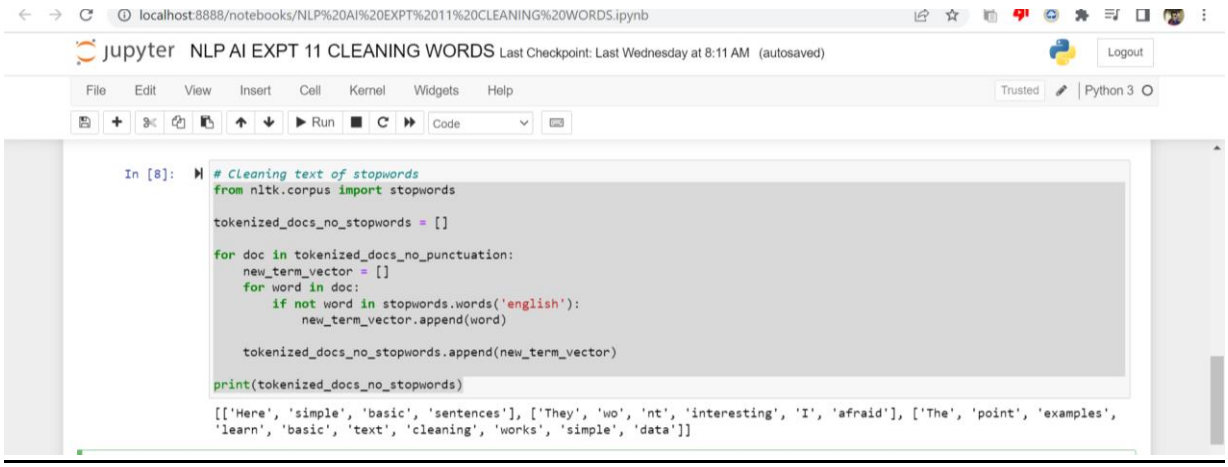
CODE:-

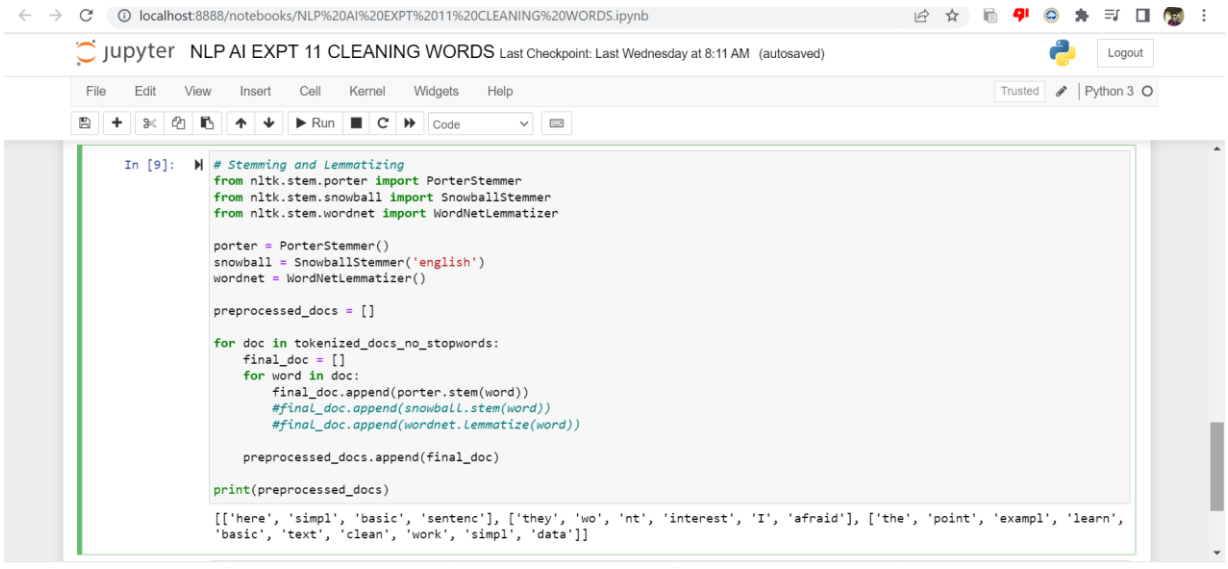
```
raw_docs = ["Here are some very simple basic sentences.",
"They won't be very interesting, I'm afraid.",
"The point of these examples is to _learn how basic text cleaning works_ on *very simple* data."]


# Tokenizing text into bags of words
from nltk.tokenize import word_tokenize
tokenized_docs = [word_tokenize(doc) for doc in raw_docs]
print(tokenized_docs)


# Removing punctuation
import re
import string
regex = re.compile('[%s]' % re.escape(string.punctuation)) #see documentation here:
http://docs.python.org/2/library/string.html

tokenized_docs_no_punctuation = []

for review in tokenized_docs:
   new_review = []
   for token in review:
      new_token = regex.sub(u'', token)
      if not new_token == u'':
         new_review.append(new_token)

   tokenized_docs_no_punctuation.append(new_review)

print(tokenized_docs_no_punctuation)


nltk.download('stopwords')
```

```
from nltk.corpus import stopwords

tokenized_docs_no_stopwords = []

for doc in tokenized_docs_no_punctuation:
   new_term_vector = []
   for word in doc:
      if not word in stopwords.words('english'):
         new_term_vector.append(word)

   tokenized_docs_no_stopwords.append(new_term_vector)

print(tokenized_docs_no_stopwords)



from nltk.stem.porter import PorterStemmer
from nltk.stem.snowball import SnowballStemmer
from nltk.stem.wordnet import WordNetLemmatizer

porter = PorterStemmer()
snowball = SnowballStemmer('english')
wordnet = WordNetLemmatizer()

preprocessed_docs = []

for doc in tokenized_docs_no_stopwords:
   final_doc = []
   for word in doc:
      final_doc.append(porter.stem(word))
      #final_doc.append(snowball.stem(word))
      #final_doc.append(wordnet.lemmatize(word))

   preprocessed_docs.append(final_doc)

print(preprocessed_docs)
```

OUTPUT:-


localhost:8888/notebooks/NLP%20AI%20EXPT%2011%20CLEANING%20WORDS.ipynb
jupyter NLP AI EXPT 11 CLEANING WORDS Last Checkpoint: Last Wednesday at 8:11 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3
Run Code
In [1]:
raw_docs = ["Here are some very simple basic sentences.",
"They won't be very interesting, I'm afraid.",
"The point of these examples is to _learn how basic text cleaning works_ on *very simple* data."]
In [2]:
# Tokenizing text into bags of words
from nltk.tokenize import word_tokenize
tokenized_docs = [word_tokenize(doc) for doc in raw_docs]
print(tokenized_docs)
[['Here', 'are', 'some', 'very', 'simple', 'basic', 'sentences', '.'], ['They', 'wo', "n't", 'be', 'very', 'interesting',
',', 'I', "'m", 'afraid', '.'], ['The', 'point', 'of', 'these', 'examples', 'is', 'to', '_learn', 'how', 'basic', 'text', 'c
leaning', 'works_', 'on', '*', 'very', 'simple', '*', 'data', '.']]


localhost:8888/notebooks/NLP%20AI%20EXPT%2011%20CLEANING%20WORDS.ipynb
jupyter NLP AI EXPT 11 CLEANING WORDS Last Checkpoint: Last Wednesday at 8:11 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3
Run Code
In [8]:
# Cleaning text of stopwords
from nltk.corpus import stopwords
tokenized_docs_no_stopwords = []
for doc in tokenized_docs_no_punctuation:
new_term_vector = []
for word in doc:
if not word in stopwords.words('english'):
new_term_vector.append(word)
tokenized_docs_no_stopwords.append(new_term_vector)
print(tokenized_docs_no_stopwords)
[['Here', 'simple', 'basic', 'sentences'], ['They', 'wo', 'nt', 'interesting', 'I', 'afraid'], ['The', 'point', 'examples',
'learn', 'basic', 'text', 'cleaning', 'works', 'simple', 'data']]


localhost:8888/notebooks/NLP%20AI%20EXPT%2011%20CLEANING%20WORDS.ipynb
jupyter NLP AI EXPT 11 CLEANING WORDS Last Checkpoint: Last Wednesday at 8:11 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3
Run Code
In [9]:
# Stemming and Lemmatizing
from nltk.stem.porter import PorterStemmer
from nltk.stem.snowball import SnowballStemmer
from nltk.stem.wordnet import WordNetLemmatizer
porter = PorterStemmer()
snowball = SnowballStemmer('english')
wordnet = WordNetLemmatizer()
preprocessed_docs = []
for doc in tokenized_docs_no_stopwords:
final_doc = []
for word in doc:
final_doc.append(porter.stem(word))
#final_doc.append(snowball.stem(word))
#final_doc.append(wordnet.lemmatize(word))
preprocessed_docs.append(final_doc)
print(preprocessed_docs)
[['here', 'simpl', 'basic', 'sentenc'], ['they', 'wo', 'nt', 'interest', 'I', 'afraid'], ['the', 'point', 'exampl', 'learn',
'basic', 'text', 'clean', 'work', 'simpl', 'data']]

RESULT:-

Hence, the Implementation of NLP for cleaning text is donesuccessfully.

In [1]:

```
raw_docs = ["Here are some very simple basic sentences.",
"They won't be very interesting, I'm afraid.",
"The point of these examples is to _learn how basic text cleaning works_ on *very simple*
data."]
```

In [2]:

```
# Tokenizing text into bags of words
from nltk.tokenize import word_tokenize
tokenized_docs = [word_tokenize(doc) for doc in raw_docs]
print(tokenized_docs)
```

```
[['Here', 'are', 'some', 'very', 'simple', 'basic', 'sentences', '.'], ['They', 'wo', "n'
t", 'be', 'very', 'interesting', ',', 'I', "'m", 'afraid', '.'], ['The', 'point', 'of', '
these', 'examples', 'is', 'to', '_learn', 'how', 'basic', 'text', 'cleaning', 'works_', '
on', '*', 'very', 'simple', '*', 'data', '.']]
```

In [3]:

```
# Removing punctuation
import re
import string
regex = re.compile('[%s]' % re.escape(string.punctuation)) #see documentation here: http
://docs.python.org/2/library/string.html

tokenized_docs_no_punctuation = []

for review in tokenized_docs:
    new_review = []
    for token in review:
        new_token = regex.sub(u'', token)
        if not new_token == u'':
            new_review.append(new_token)

    tokenized_docs_no_punctuation.append(new_review)

print(tokenized_docs_no_punctuation)
```

```
[['Here', 'are', 'some', 'very', 'simple', 'basic', 'sentences'], ['They', 'wo', 'nt', 'b
e', 'very', 'interesting', 'I', 'm', 'afraid'], ['The', 'point', 'of', 'these', 'examples
', 'is', 'to', 'learn', 'how', 'basic', 'text', 'cleaning', 'works', 'on', 'very', 'simpl
e', 'data']]
```

In [5]:

```
nltk.download('stopwords')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-5-e3fc0c9c9a89> in <module>
----> 1 nltk.download('stopwords')

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

In [6]:

```
import nltk
```

In [7]:

```
nltk.download('stopwords')
```

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

Out[7]:

```
True
```

In [8]:

```
# Cleaning text of stopwords
from nltk.corpus import stopwords

tokenized_docs_no_stopwords = []

for doc in tokenized_docs_no_punctuation:
    new_term_vector = []
    for word in doc:
        if not word in stopwords.words('english'):
            new_term_vector.append(word)

    tokenized_docs_no_stopwords.append(new_term_vector)

print(tokenized_docs_no_stopwords)
```

```
[['Here', 'simple', 'basic', 'sentences'], ['They', 'wo', 'nt', 'interesting', 'I', 'afra
id'], ['The', 'point', 'examples', 'learn', 'basic', 'text', 'cleaning', 'works', 'simple
', 'data']]
```

In [9]:

```
# Stemming and Lemmatizing
from nltk.stem.porter import PorterStemmer
from nltk.stem.snowball import SnowballStemmer
from nltk.stem.wordnet import WordNetLemmatizer

porter = PorterStemmer()
snowball = SnowballStemmer('english')
wordnet = WordNetLemmatizer()

preprocessed_docs = []

for doc in tokenized_docs_no_stopwords:
    final_doc = []
    for word in doc:
        final_doc.append(porter.stem(word))
        #final_doc.append(snowball.stem(word))
        #final_doc.append(wordnet.lemmatize(word))

    preprocessed_docs.append(final_doc)

print(preprocessed_docs)
```

```
[['here', 'simpl', 'basic', 'sentenc'], ['they', 'wo', 'nt', 'interest', 'I', 'afraid'],
['the', 'point', 'exampl', 'learn', 'basic', 'text', 'clean', 'work', 'simpl', 'data']]
```

In [ ]: