EXPERIMENT NO 1

SHIVAM NAGORI
60009210093 D12

AIM: To study and implement Preprocessing of text (Tokenization, Filtration, Script Validation, Stop Word Removal, Stemming)

THEORY:

1. Tokenization:

Tokenization is a common task in Natural Language Processing (NLP). It's a fundamental step in both traditional NLP methods like Count Vectorizer and Advanced Deep Learning-based architectures like Transformers. Tokenization is a way of separating a piece of text into smaller units called tokens. Here, tokens can be either words, characters, or sub words. Hence, tokenization can be broadly classified into 3 types – word, character, and sub word (n-gram characters) tokenization.

For example, consider the sentence: "Never give up".

The most common way of forming tokens is based on space. Assuming space as a delimiter, the tokenization of the sentence results in 3 tokens – **Never-give-up**. As each token is a word, it becomes an example of Word tokenization.

2. Filtration:

Many of the words used in the phrase are insignificant and hold no meaning. For example – English is a subject. Here, 'English' and 'subject' are the most significant words and 'is', 'a' are almost useless. English subject and subject English hold the same meaning even if we remove the insignificant words – ('is', 'a'). Using the nltk, we can remove the insignificant words by looking at their part-of-speech tags. For that, we must decide which Part-Of-Speech tags are significant.

Word	Tag
a	DT
all	PDT
an	DT
and	CC
or	CC
that	WDT
the	DT

3. Stop Word Removal:

All the words in a query are stop words. If all the query terms are removed during stop word processing, then the result set is empty. To ensure that search results are returned, stop word removal is disabled when all the query terms are stop words. For example, if the word *car* is a stop word and you search for *car*, then the search results contain documents that match the word *car*. If you search for *car buick*, the search results containonly documents that match the word *buick*.

The word in a query is preceded by the plus sign (+).The word is part of an exact match.
The word is inside a phrase, for example, "I love my car".

4. Stemming:

Stemming is a part of linguistic studies in morphology and artificial intelligence (AI) information retrieval and extraction. Stemming and AI knowledge extract meaningful information from vast sources like big data or the Internet since additional forms of a word related to a subject may need to be searched to get the best results. Stemming is also a part of queries and Internet engines. Recognizing, searching and retrieving more forms of words returns more results.

Lab Experiments to be Performed in This Session: -

Perform Following Preprocessing Techniques on the given corpus

1. Tokenization,

```
💡 Click here to ask Blackbox to help you code faster
import nltk
from nltk.tokenize import word_tokenize, sent_tokenize
import inflect
from nltk.corpus import stopwords
nltk.download('punkt')
nltk.download('stopwords')
```

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

```
sentenc = "My name is Shivam and I am from 3rd year Data Science from
DJSCE College of engineering , 400056 - Mumbai. My date of birth is 9
December 2003"
word_tokenize_list = word_tokenize(sentenc)
print("word tokens : ", word_tokenize)
```

2. Converting Text Lower Case

```
print("Lowered tokens : ", word_lower_tokenize)
```

```
my name is shivam and i am from 3rd year data science from djsce college of engineering , 400056 - mumbai .
```

3. Remove Numbers

```
only_char_tokens = [token for token in word_tokenize(sentenc) if
token.isalpha()]
only_numbers = [token for token in word_tokenize(sentenc) if not
token.isalpha()]
```

```
Only alphabets tokens :  ['My', 'name', 'is', 'Shivam', 'and', 'I', 'am', 'from', 'year', 'Data', 'Science', 'from', 'DJSCE', 'College', 'o
```

4. Converting Number to Words

```
p = inflect.engine()
word_tokens = [p.number_to_words(token.lower()) if token.isdigit() else
token.lower() for token in word_tokenize(sentenc)]

for word in word_tokens:
    print(word)
```

```
from djsce college of engineering , four hundred thousand and fifty-six - mumbai . my date of birth is nine december two thousand and three
```

5. Remove Punctuation

```
no_punctuations_tokens = [token for token in word_tokenize(sentenc) if
token.isalpha() or token.isdigit()]
for token in no_punctuations_tokens:
print(token)
```

```
My name is Shivam and I am from year Data Science from DJSCE College of engineering 400056 Mumbai My date of birth is 9 December 2003
```

6. Remove Whitspaces

```
no_white_spaces_token = [token.strip() for token in
word_tokenize(sentenc) if token.isalpha() or token.isdigit()]
```

```
My name is Shivam and I am from year Data Science from DJSCE College of engineering 400056 Mumbai My date of birth is 9 December 2003
```

7. Remove StopWords

```
from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))

no_stop_words_token = [token.lower().strip() for token in
word_tokenize(sentenc) if token.lower().isalpha() or token.isdigit()
not in stop_words]
for token in no_stop_words_token:
    print(token)
```

```
my name is shivam and i am from 3rd year data science from djsce college of engineering , 400056 - mumbai . my date of birth is 9 december
```

8. Count Word Frequency

```
from nltk.probability import FreqDist
frequency_dist = FreqDist(no_stop_words_token)

for word , frequency in frequency_dist.items(): print(f"{word} :
{frequency}")
```

```
My : 2
name : 1
is : 2
Shivam : 1
and : 1
I : 1
am : 1
from : 2
3rd : 1
year : 1
Data : 1
Science : 1
DJSCE : 1
College : 1
of : 2
engineering : 1
, : 1
400056 : 1
- : 1
Mumbai : 1
. : 1
date : 1
birth : 1
9 : 1
December : 1
2003 : 1
```

9. Stemming (Porter Stemmer and Lancaster Stemmer)

PORTER STEMMER :

```
from nltk.stem import PorterStemmer
porter = PorterStemmer()
stemmed_words = [porter.stem(word) for word in no_stop_words_token]

for word in stemmed_words:
      print(word)
```

```
my name is shivam and i am from 3rd year data scienc from djsce colleg of engin , 400056 - mumbai . my date of birth is 9 decemb 2003
```

Lancaster stemmer

```
from nltk.stem import LancasterStemmer
lancaster = LancasterStemmer()
stemmed_words = [lancaster.stem(word) for word in no_stop_words_token]

for token in stemmed_words:
    print(token)
```

```
my nam is shivam and i am from 3rd year dat sci from djsce colleg of engin , 400056 - mumba . my dat of bir is 9 decemb 2003
```

10. Lemmatization

```
import nltk
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer, WordNetLemmatizer
from nltk.corpus import stopwords

nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')

sentence = "My name is shivam. I am from datascience third year."

tokens = word_tokenize(sentence)
stop_words = set(stopwords.words('english'))
tokens_cleaned = [token.lower() for token in tokens if token.lower() not in
stop_words]
wordnet_lemmatizer = WordNetLemmatizer()

lemmatized_tokens = [wordnet_lemmatizer.lemmatize(token) for token in
tokens_cleaned]

print("Lemmatized tokens:", lemmatized_tokens,"\n")
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Shivam\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Shivam\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\Shivam\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
Lemmatized tokens: ['name', 'shivam', '.', 'datascience', 'third', 'year', '.']
```