Experiment No. 1

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

Theory:

To preprocess your text simply means to bring your text into a form that is predictable and analyzable for your task. A task here is a combination of approach and domain.

Machine Learning needs data in the numeric form. We basically used encoding techniques (BagOfWord, Bi-gram,n-gram, TF-IDF, Word2Vec) to encode text into numeric vectors. But before encoding we first need to clean the text data and this process to prepare (or clean) text data before encoding is called text preprocessing, this is the very first step to solve the NLP problems.

Tokenization:

Tokenization is about splitting strings of text into smaller pieces, or “tokens”. Paragraphs can be tokenized into sentences and sentences can be tokenized into words.

Filtration:

Similarly, if we are doing simple word counts, or trying to visualize our text with a word cloud, stopwords are some of the most frequently occurring words but don’t really tell us anything. We’re often better off tossing the stopwords out of the text. By checking the Filter Stopwords option in the Text Preprocessing tool, you can automatically filter these words out.

Script Validation:

The script must be validated properly.

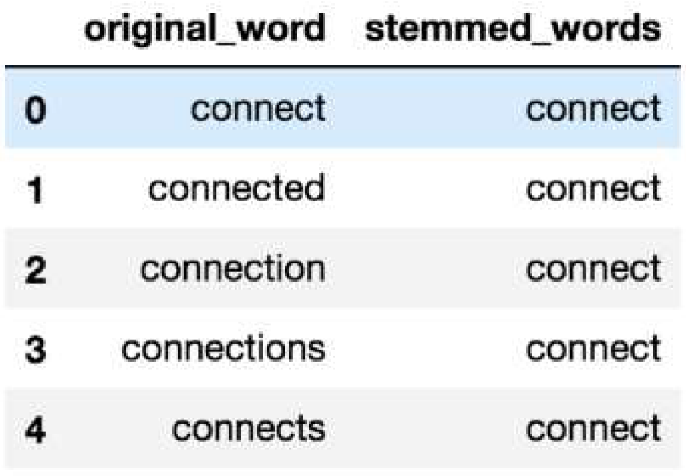
Stemming:

Stemming is the process of reducing inflection in words (e.g. troubled, troubles) to their root form (e.g. trouble). The “root” in this case may not be a real root word, but just a canonical form of the original word. Stemming uses a crude heuristic process that chops off the ends of words in the hope of correctly transforming words into its root form. So, the words “trouble”, “troubled” and “troubles” might actually be converted to trouble instead of trouble because the ends were just chopped off (ughh, how crude!).

There are different algorithms for stemming. The most common algorithm, which is also known to be empirically effective for English, is Porter's Algorithm. Here is an example of stemming in action with

Porter Stemmer:

Jaydev Singh Charitable Trust's (Regd.) THAKUR COLLEGE OF ENGINEERING & TECHNOLOGY tcet Estd. 2001 ISO 9001 : 2015 Certified NBA and NAAC Accredited



	original_word	stemmed_words
0	connect	connect
1	connected	connect
2	connection	connect
3	connections	connect
4	connects	connect

Stopword Removal

Stop words are a set of commonly used words in a language. Examples of stop words in English are “a”, “the”, “is”, “are” and etc. The intuition behind using stop words is that, by removing low information words from text, we can focus on the important words instead. For example, in the context of a search system, if your search query is *“what is text preprocessing?”*, you want the search system to focus on surfacing documents that talk about text preprocessing over documents that talk about what is. This can be done by preventing all words from your stop word list from being analyzed. Stop words are commonly applied in search systems, text classification applications, topic modeling, topic extraction and others.
In my experience, stop word removal, while effective in search and topic extraction systems, showed to be non-critical in classification systems. However, it does help reduce the number of features in consideration which helps keep your models decently sized. Here is an example of stop word removal in action. All stop words are replaced with a dummy
character, W:

ISO 9001 : 2015 Certified
NBA and NAAC Accredited

Code:

String handling:

```
print(len("what it is what it isnt"))

s=["what","it","is" ,"what","it","isnt"]

print(len(s))

x=sorted(s)

print(s)
print(x)

d=x+s

print(d)

print(d)
```

Output:

```
23

6

['what', 'it', 'is', 'what', 'it', 'isnt']

['is', 'isnt', 'it', 'it', 'what', 'what']

['is', 'isnt', 'it', 'it', 'what', 'what', 'what', 'it', 'is', 'what', 'it', 'isnt']s
```

File handling (tokenization and filtering):

```
for line in open("file.txt"):

for word in line.split():

if word.endswith('ing'):

print(word)

print(len(word))
```

Output:

```
eating
6
dancing
7
jumping
7
File.txt
```

Conclusion:

In the above experiment we have studied preprocessing of text in detail like filtration, stop word removal, tokenization, stemming, script validation and have tried to implement the code for it and have successfully executed it.

Zagdu Singh Charitable Trust's (Regd.)
THAKUR COLLEGE OF ENGINEERING & TECHNOLOGY
Estd. 2001
ISO 9001 : 2015 Certified
NBA and NAAC Accredited