ANISH ADNANI
D17B/01

EXPERIMENT 3

Aim: Apply various other text preprocessing techniques for any given text stop word removal, Lemmatization / Stemming.

Objective: To understand some preprocessing techniques like removal of stopwords and to know about stemming of words

Prerequisites: stopwords, lemmatization, stemming concepts

Theory

Finite Automata (FA) is the simplest machine to recognize patterns. The FA or FSM is an abstract machine which have five elements on tuple. It has a set of states and rodes for moving from one state to another. It depends upon the applied input symbol. Basically, it is an abstract model of digital computer

Formal specification of machine is $\{Q, \Sigma, q, F, \delta\}$ where

- Q finite set of states
- Σ set of input symbols
- q initial state
- F set of final states
- δ Transition function

Stemming and Lemmatization are Text Normalization techniques in the field of NLP. They are used to prepare text, words and documents for further processing

Stemming is the process of reducing inflection in words to their root forms such as a group of words to the same stem even if the stem is not valid word in the language. Lemmatization unlike stemming reduces the inflected words properly ensuring that the root word belongs to the language.

Stemming algorithms

i) Porter's stemmer algorithm: It is one of the most popular method proposed in 1980. It is based on the idea that the suffixes are made up of a combination of smaller and simpler suffixes in english language. Applications include data mining, info retrieval, etc
Advantage: Best output and less error rate
Limitation: morphological variants produced are not always real words

ii) Lovins stemmer: It removes the longest suffix from a word. then the word is recoded to convert this stem into valid words - Advantage is its fast and handles irregulation plurals like teeth
Limitation: Time consuming

iii) Dawson stemmer: It is an extension of Lovins stemmer in which suffixes are stored in the reversed order indexed by their length and last letter
Advantage: Fast in execution
Limitation: The algorithm is very complex to understand

iv) Snowball stemmer: It is an advance version of porter stemmer since it can also map non-English words too

v) Lancester stemmer: They are more aggresive they also are dynamic in nature as compared to Porter stemmer. It is really first, but the algorithm is really composing dealing with small words. It basically saves the rules externally and uses an iterative algorithm

Steps in implementation of porter stemmer algorithm

1. We begin by removing common suffixes and pluralization
2. It turns terminal -y to -i when there is another vowel in the stem
3. It maps double suffixes to single ones: so -ization maps to -ize, etc
4. It deals with the -ic, -ness, etc
5. It takes off -ant, -ence, etc

Conclusion: We tested some techniques to remove stopwords from a given text document. Similarly we also learned a couple of stemming algorithm such as porter stemmer algorithm and Lancaster algorithm. Lancaster algorithm is much more aggresive than porter stemmer algorithm. In general, lemmatization is better than stemming since it offers good precision compared to stemming. But, it is being offered at the expense of recall. Hence Lemmatization is slower than stemming.

Program

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

print("--> Applying PorterStemmer Algorithm \n")

porter = PorterStemmer()

sentence = """He is dancing and singing at same time.
          He has bad habit of eating after dancing for long hours in the Sun."""

word_list = nltk.word_tokenize(sentence)

Filtered_sentence = []
for word in word_list:
   Filtered_sentence.append(porter.stem(word))

print("Sentence before applying porter stemmer: ")
print(word_list)
print()

print("Sentence after applying porter stemmer: ")
print(Filtered_sentence)

print()
print()

print("--> Applying Lemmatization \n")

Lemmatizer = WordNetLemmatizer()

punctuations = "?:!.,;"

Filtered_sentence = []

for word in word_list:
   if word not in punctuations:
      Filtered_sentence.append(Lemmatizer.lemmatize(word))

print("Sentence before applying Lemmatization: ")
print(word_list)
print()
```

```
print("Sentence after applying Lemmatization: ")
print(Filtered_sentence)

print()
print()
print("--> Applying Stopword Removal \n")

stop_words = set(stopwords.words('english'))

Filtered_sentence = [word for word in word_list if word not in stop_words]

print("Sentence before Removing Stopwords: ")
print(word_list)
print()

print("Sentence after Removing Stopwords: ")
print(Filtered_sentence)

print()
print()
```

Output

Applying porter stemmer algorithm

```
print("--> Applying PorterStemmer Algorithm \n")
porter = PorterStemmer()
sentence = """He is dancing and singing at same time.
He has bad habit of eating after dancing for long hours in the Sun."""
word_list = nltk.word_tokenize(sentence)
Filtered_sentence = []
```

```
--> Applying PorterStemmer Algorithm
```

```
for word in word_list:
    Filtered_sentence.append(porter.stem(word))
print("Sentence before applying porter stemmer: ")
print(word_list)
print()
print("Sentence after applying porter stemmer: ")
print(Filtered_sentence)
print()
print()
```

```
Sentence before applying porter stemmer:
['He', 'is', 'dancing', 'and', 'singing', 'at', 'same', 'time', '.', 'He', 'has', 'bad', 'habit', 'of', 'eating', 'after', 'dan
cing', 'for', 'long', 'hours', 'in', 'the', 'Sun', '.']

Sentence after applying porter stemmer:
['He', 'is', 'danc', 'and', 'sing', 'at', 'same', 'time', '.', 'He', 'ha', 'bad', 'habit', 'of', 'eat', 'after', 'danc', 'for',
'long', 'hour', 'in', 'the', 'sun', '.']
```

Applying lemmatization

```
print("--> Applying Lemmatization \n")
Lemmatizer = WordNetLemmatizer()
punctuations = "?:!.,;"
Filtered_sentence = []
for word in word_list:
    if word not in punctuations:
        Filtered_sentence.append(Lemmatizer.lemmatize(word))
print("Sentence before applying Lemmatization: ")
print(word_list)
print()
print("Sentence after applying Lemmatization: ")
print(Filtered_sentence)
print()
print()
```

```
--> Applying Lemmatization

Sentence before applying Lemmatization:
['He', 'is', 'dancing', 'and', 'singing', 'at', 'same', 'time', '.', 'He', 'has', 'bad', 'habit', 'of', 'eating', 'after', 'dan
cing', 'for', 'long', 'hours', 'in', 'the', 'Sun', '.']

Sentence after applying Lemmatization:
['He', 'is', 'dancing', 'and', 'singing', 'at', 'same', 'time', 'He', 'ha', 'bad', 'habit', 'of', 'eating', 'after', 'dancing',
'for', 'long', 'hour', 'in', 'the', 'Sun']
```

Applying stop word removal

```
print("--> Applying Stopword Removal \n")
stop_words = set(stopwords.words('english'))
Filtered_sentence = [word for word in word_list if word not in stop_words]
print("Sentence before Removing Stopwords: ")
print(word_list)
print()
print("Sentence after Removing Stopwords: ")
print(Filtered_sentence)
print()
print()
```

```
--> Applying Stopword Removal

Sentence before Removing Stopwords:
['He', 'is', 'dancing', 'and', 'singing', 'at', 'same', 'time', '.', 'He', 'has', 'bad', 'habit', 'of', 'eating', 'after', 'dan
cing', 'for', 'long', 'hours', 'in', 'the', 'Sun', '.']

Sentence after Removing Stopwords:
['He', 'dancing', 'singing', 'time', '.', 'He', 'bad', 'habit', 'eating', 'dancing', 'long', 'hours', 'Sun', '.']
```