**NLP Assignment Solutions**

**Exercises on Tokenization and Stopwords:**

1. Why do we need to tokenize a text in NLP?

**NLP involves text analysis and a text is tokenized in NLP to enable the textual content to be broken down into small pieces or chunks (which could be broken down into a token of each word if a sentence is tokenized or sentence if a paragraph is tokenized).**

1. Can you tell me how many tokens contains in the bold text: **The wolf said: "Little pig, little pig, let me come in."**

**Number of tokens is: 17 or 11 if I space tokenize**

**PROGRAM:**

#Word Tokenization

# Importing word tokenizer from NLTK

from nltk.tokenize import word\_tokenize

# Create an input title

msg = '''The wolf said: "Little pig, little pig, let me come in."'''

# breaking the title into word/space tokens and assigning it to a new variable.

tokenized\_text = word\_tokenize(msg)

tokenized\_text1 = SpaceTokenizer().tokenize(msg)

# Printing the tokenized paragraph. It will return a list of tokens.

print("This is Word Tokenization")

print(tokenized\_text)

print(len(tokenized\_text))

print("This is Space Tokenization")

print(tokenized\_text1)

print(len(tokenized\_text1))

**OUTPUT:**

This is Word Tokenization

['The', 'wolf', 'said', ':', '``', 'Little', 'pig', ',', 'little', 'pig', ',', 'let', 'me', 'come', 'in', '.', "''"]

17

This is Space Tokenization

['The', 'wolf', 'said:', '"Little', 'pig,', 'little', 'pig,', 'let', 'me', 'come', 'in."']

11

1. Consider the following book title: ***This Is the Beat Generation: New York-San Francisco-Paris.*** What would it take to be able to tokenize such strings so that each city name was stored as a single token?

**For each city name to be stored as a single token was achieved through *Tokenize using regular expression.***

**PROGRAM:**

# Importing word tokenizer from NLTK

from nltk.tokenize import word\_tokenize, regexp\_tokenize

import regex, re

# The text title

book\_title = "This Is the Beat Generation: New York-San Francisco-Paris"

#normal python built-in function

itokens = book\_title.split()

#print(itokens)

# Tokenize using regular expression

atokens = re.split('[:-]', book\_title)

print ("This is first tokenized Sentence using Regex - regular expression: \n" )

print(atokens)

print ("========================\n")

# Tokenize using nltk package

tokens = tokenized\_text = word\_tokenize(book\_title)

print ("This is word tokenized using nltk package: \n" )

print(tokens)

print ("========================\n")

reg\_pattern1 = '(\\w+|\\-?:?|)'

reg\_pattern2 = '(\\w+|\\-:|)'

tokens1 = regexp\_tokenize(book\_title, reg\_pattern1)

tokens2 = regexp\_tokenize(book\_title, reg\_pattern2)

print ("This is 1st tokenized Sentence using Regex - nltk package: \n" )

print(tokens1)

print ("========================\n")

print ("This is 2nd tokenized Sentence using Regex - nltk package: \n" )

print(tokens2)

**OUTPUT:**

This is first tokenized Sentence using Regex - regular expression:

['This Is the Beat Generation', ' New York', 'San Francisco', 'Paris']

========================

This is word tokenized using nltk package:

['This', 'Is', 'the', 'Beat', 'Generation', ':', 'New', 'York-San', 'Francisco-Paris']

========================

This is 1st tokenized Sentence using Regex - nltk package:

['This', '', 'Is', '', 'the', '', 'Beat', '', 'Generation', ':', '', 'New', '', 'York', '-', 'San', '', 'Francisco', '-', 'Paris', '']

========================

This is 2nd tokenized Sentence using Regex - nltk package:

['This', '', 'Is', '', 'the', '', 'Beat', '', 'Generation', '', '', 'New', '', 'York', '', 'San', '', 'Francisco', '', 'Paris', '']

>>>

1. Is this text tokenized?
   * The wolf said: “Little pig, little pig, let me come in. "

**It looks like the Sentence is either Sentence or Line Tokenized, see program output below:**

**PROGRAM:**

# Sample Sentence: The wolf said: “Little pig, little pig, let me come in. "

# Sentence is either Sentence or Line Tokenized see code

# Importing word tokenizer from NLTK

from nltk.tokenize import word\_tokenize, sent\_tokenize, LineTokenizer

txt = '''The wolf said : " Little pig , little pig , let me come in . "'''

tokenized\_text1 = sent\_tokenize(txt)

print ("This is Sentence Tokenize: \n" )

print(tokenized\_text1 )

print ("========================\n")

tokenized\_text2 = LineTokenizer().tokenize(txt)

print ("This is Line Tokenize call: \n" )

print(tokenized\_text2)

print ("========================\n")

**OUTPUT:**

This is Sentence Tokenize:

['The wolf said : " Little pig , little pig , let me come in . "']

========================

This is Line Tokenize call:

['The wolf said : " Little pig , little pig , let me come in . "']

========================

* + Welcome

!  
Can

I  
help  
you  
?  
I

’m  
fine  
.

**It looks like the about Sentence is either Line or Tab Tokenized, see program output below:**

**PROGRAM:**

from nltk.tokenize import word\_tokenize, LineTokenizer, TabTokenizer

txt1 = "Welcome! Can I help you? I’m fine."

tokenized\_text = LineTokenizer().tokenize(txt1)

print(tokenized\_text)

print ("========================\n")

tokenized\_text2 = TabTokenizer().tokenize(txt1)

print(tokenized\_text2)

print ("========================\n")

**OUTPUT:**

This is Line Tokenize call:

['Welcome! Can I help you? I’m fine.']

========================

This is Tab Tokenize call:

['Welcome! Can I help you? I’m fine.']

========================

1. Given the text:

***’The receptor-ligand complexes were then released from the cell membrane preparation ’ +\ ’by incubation with RIB + detergent, which contained RIB appended with 50 mM ’ +\ ’N-dodecyl-N, N (dimethylammonio) butyrate, 1.5% glycerol, and 2% NP-40 for 1 h at 37C’ +\ ’and centrifuged at 95,000 rpm for 3 h at 15C.’***

- Tokenize using regular expression

- Tokenize using nltk package

- For each tokenization, compute the number of tokens, using the len() function, and print the result.

**PROGRAM:**

# Importing word tokenizer from NLTK

from nltk.tokenize import word\_tokenize, regexp\_tokenize

import regex, re

# The text

txt = "The receptor-ligand complexes were then released from the cell membrane preparation ’ +\ ’by incubation with RIB + detergent, which contained RIB appended with 50 mM ’ +\ ’N-dodecyl-N, N (dimethylammonio) butyrate, 1.5% glycerol, and 2% NP-40 for 1 h at 37C’ +\ ’and centrifuged at 95,000 rpm for 3 h at 15C."

print("This is the main Text." )

print("========================\n")

print(txt + "\n")

#normal python built-in Split function

itokens = txt.split()

print ("Using normal python built-in split function : \n" )

print(itokens)

print ("No. of Tokens is = " )

print (len(itokens))

print ("========================\n")

# Tokenize using regular expression

atokens = re.split('\W', txt)

print ("This is first tokenized Sentence using Regex - regular expression: \n" )

print(atokens)

print ("No. of Tokens is = " )

print (len(atokens))

print ("========================\n")

print ("OR\n")

atokens1 = re.split('\W+', txt)

print(atokens1)

print ("No. of Tokens is = " )

print (len(atokens1))

print ("========================\n")

# Tokenize using nltk package

tokens = tokenized\_text = word\_tokenize(txt)

print ("This is word tokenized using nltk package: \n" )

print(tokens)

print ("No. of Tokens is = " )

print (len(tokens))

print ("========================\n")

reg\_pattern1 = '(\\w+|\\+?%?|)'

reg\_pattern2 = '(\\w+)'

tokens1 = regexp\_tokenize(txt, reg\_pattern1)

tokens2 = regexp\_tokenize(txt, reg\_pattern2)

print ("This is 1st tokenized Sentence using Regex - nltk package: \n" )

print(tokens1)

print ("No. of Tokens is = " )

print (len(tokens1))

print ("========================\n")

print ("This is 2nd tokenized Sentence using Regex - nltk package: \n" )

print(tokens2)

print ("No. of Tokens is = " )

print (len(tokens2))

print ("========================\n")

**OUTPUT:**

This is the main Text.

========================

The receptor-ligand complexes were then released from the cell membrane preparation ’ +\ ’by incubation with RIB + detergent, which contained RIB appended with 50 mM ’ +\ ’N-dodecyl-N, N (dimethylammonio) butyrate, 1.5% glycerol, and 2% NP-40 for 1 h at 37C’ +\ ’and centrifuged at 95,000 rpm for 3 h at 15C.

Using normal python built-in split function :

['The', 'receptor-ligand', 'complexes', 'were', 'then', 'released', 'from', 'the', 'cell', 'membrane', 'preparation', '’', '+\\', '’by', 'incubation', 'with', 'RIB', '+', 'detergent,', 'which', 'contained', 'RIB', 'appended', 'with', '50', 'mM', '’', '+\\', '’N-dodecyl-N,', 'N', '(dimethylammonio)', 'butyrate,', '1.5%', 'glycerol,', 'and', '2%', 'NP-40', 'for', '1', 'h', 'at', '37C’', '+\\', '’and', 'centrifuged', 'at', '95,000', 'rpm', 'for', '3', 'h', 'at', '15C.']

No. of Tokens is =

53

========================

This is first tokenized Sentence using Regex - regular expression:

['The', 'receptor', 'ligand', 'complexes', 'were', 'then', 'released', 'from', 'the', 'cell', 'membrane', 'preparation', '', '', '', '', '', '', 'by', 'incubation', 'with', 'RIB', '', '', 'detergent', '', 'which', 'contained', 'RIB', 'appended', 'with', '50', 'mM', '', '', '', '', '', '', 'N', 'dodecyl', 'N', '', 'N', '', 'dimethylammonio', '', 'butyrate', '', '1', '5', '', 'glycerol', '', 'and', '2', '', 'NP', '40', 'for', '1', 'h', 'at', '37C', '', '', '', '', '', 'and', 'centrifuged', 'at', '95', '000', 'rpm', 'for', '3', 'h', 'at', '15C', '']

No. of Tokens is =

81

========================

OR

['The', 'receptor', 'ligand', 'complexes', 'were', 'then', 'released', 'from', 'the', 'cell', 'membrane', 'preparation', 'by', 'incubation', 'with', 'RIB', 'detergent', 'which', 'contained', 'RIB', 'appended', 'with', '50', 'mM', 'N', 'dodecyl', 'N', 'N', 'dimethylammonio', 'butyrate', '1', '5', 'glycerol', 'and', '2', 'NP', '40', 'for', '1', 'h', 'at', '37C', 'and', 'centrifuged', 'at', '95', '000', 'rpm', 'for', '3', 'h', 'at', '15C', '']

No. of Tokens is =

54

========================

This is word tokenized using nltk package:

['The', 'receptor-ligand', 'complexes', 'were', 'then', 'released', 'from', 'the', 'cell', 'membrane', 'preparation', '’', '+\\', '’', 'by', 'incubation', 'with', 'RIB', '+', 'detergent', ',', 'which', 'contained', 'RIB', 'appended', 'with', '50', 'mM', '’', '+\\', '’', 'N-dodecyl-N', ',', 'N', '(', 'dimethylammonio', ')', 'butyrate', ',', '1.5', '%', 'glycerol', ',', 'and', '2', '%', 'NP-40', 'for', '1', 'h', 'at', '37C', '’', '+\\', '’', 'and', 'centrifuged', 'at', '95,000', 'rpm', 'for', '3', 'h', 'at', '15C', '.']

No. of Tokens is =

66

========================

This is 1st tokenized Sentence using Regex - nltk package:

['The', '', 'receptor', '', 'ligand', '', 'complexes', '', 'were', '', 'then', '', 'released', '', 'from', '', 'the', '', 'cell', '', 'membrane', '', 'preparation', '', '', '', '+', '', '', '', 'by', '', 'incubation', '', 'with', '', 'RIB', '', '+', '', 'detergent', '', '', 'which', '', 'contained', '', 'RIB', '', 'appended', '', 'with', '', '50', '', 'mM', '', '', '', '+', '', '', '', 'N', '', 'dodecyl', '', 'N', '', '', 'N', '', '', 'dimethylammonio', '', '', 'butyrate', '', '', '1', '', '5', '%', '', 'glycerol', '', '', 'and', '', '2', '%', '', 'NP', '', '40', '', 'for', '', '1', '', 'h', '', 'at', '', '37C', '', '', '+', '', '', '', 'and', '', 'centrifuged', '', 'at', '', '95', '', '000', '', 'rpm', '', 'for', '', '3', '', 'h', '', 'at', '', '15C', '', '']

No. of Tokens is =

134

========================

This is 2nd tokenized Sentence using Regex - nltk package:

['The', 'receptor', 'ligand', 'complexes', 'were', 'then', 'released', 'from', 'the', 'cell', 'membrane', 'preparation', 'by', 'incubation', 'with', 'RIB', 'detergent', 'which', 'contained', 'RIB', 'appended', 'with', '50', 'mM', 'N', 'dodecyl', 'N', 'N', 'dimethylammonio', 'butyrate', '1', '5', 'glycerol', 'and', '2', 'NP', '40', 'for', '1', 'h', 'at', '37C', 'and', 'centrifuged', 'at', '95', '000', 'rpm', 'for', '3', 'h', 'at', '15C']

No. of Tokens is =

53

========================

>>>

**Stop Words: Exercise 2**

1. Lists the approach used to remove stop words?
2. Remove stop words from the text below:

"Stop words are most common words found in any natural language which carries very little or no significant semantic context in a sentence. It just carry syntactic importance which aid in formation of sentence. As a preprocessing operation it must be removed to ease further task and speedup core task in text processing. Ibrahim A [3] conducted a comparative study on the effect of stop words elimination on Arabic Information Retrieval where three stop lists viz, General Stop list, corpus based stop-list and combined stop list were used for comparative study. General stop-list performed better than the rest of the two. Ashish T, et al [4] eliminated stop-word in Gujarati language by preparing frequency list from Gujarati corpus by analyzing popular Gujarati newspapers. Riyad A, et al [5], used Finite State Machine (FSM) algorithm to eliminate stop-words for Arabic Language. Basim A, et al [6] have designed and implemented a new stop-word removal technique for Arabic language based on dedicated list and algorithm which compares stopwords if it fulfills desired string length criteria. Vijayarani S, et al[7] used Zipf’s Law (Z method) for creation of stop-words. Rakholia and Saini [8] have presented a rule-based approach to dynamically identify stop words for Gujarati language. They have also deployed this approach with additional cosine similarity based Vector Space Model for information retrieval in Gujarati language [9]. Kaur J and Saini JR have presented the list of Punjabi stop words [10], its Partof-Speech class based classification [11] and its Gurumukhi and Shahmukhi script versions [12]. Saini and Rakholia [13] have presented an analytic in-depth report on continent and script-wise divisions-based statistical measures for stopwords lists of various international Languages."

1. Lists the approach used to remove stop words?
2. Create an input string
3. Create a set of English Stopwords
4. Removing Stopwords from word\_tokens and inserting rest of the words in filtered sentence list using looping or list comprehension.
5. Program to Remove stop words from the sample text:

**PROGRAM:**

# Create an input string like a story for instance

# Create a set of English Stopwords

# Removing Stopwords from word\_tokens and inserting rest of the words in filtered sentence list using looping or list comprehension.

# Importing stopwords and word tokenize from NLTK.corpus and tokenize

# respectively

from nltk.corpus import stopwords

from nltk.tokenize import word\_tokenize

# The Story

story = "Stop words are most common words found in any natural language which carries very little or no significant semantic context in a sentence. It just carry syntactic importance which aid in formation of sentence. As a preprocessing operation it must be removed to ease further task and speedup core task in text processing. Ibrahim A [3] conducted a comparative study on the effect of stop words elimination on Arabic Information Retrieval where three stop lists viz, General Stop list, corpus based stop-list and combined stop list were used for comparative study. General stop-list performed better than the rest of the two. Ashish T, et al [4] eliminated stop-word in Gujarati language by preparing frequency list from Gujarati corpus by analyzing popular Gujarati newspapers. Riyad A, et al [5], used Finite State Machine (FSM) algorithm to eliminate stop-words for Arabic Language. Basim A, et al [6] have designed and implemented a new stop-word removal technique for Arabic language based on dedicated list and algorithm which compares stopwords if it fulfills desired string length criteria. Vijayarani S, et al[7] used Zipf’s Law (Z method) for creation of stop-words. Rakholia and Saini [8] have presented a rule-based approach to dynamically identify stop words for Gujarati language. They have also deployed this approach with additional cosine similarity based Vector Space Model for information retrieval in Gujarati language [9]. Kaur J and Saini JR have presented the list of Punjabi stop words [10], its Partof-Speech class based classification [11] and its Gurumukhi and Shahmukhi script versions [12]. Saini and Rakholia [13] have presented an analytic in-depth report on continent and script-wise divisions-based statistical measures for stopwords lists of various international Languages."

#English stopwords

stop\_words = set(stopwords.words('english'))

#Tokenize story

word\_tokens = word\_tokenize(story)

#Printing out Tokenize story

print("See Tokenized words list below:")

print ("=======================\n")

print (stop\_words)

print ("=======================\n")

print ("See the No.# of words below:")

print ("=======================\n")

print (len(stop\_words))

print ("=======================\n")

print("See Stop words list below:")

print ("=======================\n")

print (word\_tokens)

print ("=======================\n")

print ("See the No.# of Stop words below:")

print ("=======================\n")

print (len(word\_tokens))

print ("=======================\n")

# Removing Stopwords from word\_tokens and inserting rest of the words in

# filtered\_sentence list using list comprehension.

filtered\_sentence = [w for w in word\_tokens if not w in stop\_words]

print("See Tokenize words list below:")

print ("=======================\n")

print("Word Tokens : ", word\_tokens)

print ("=======================\n")

print("See the NO.# of Stop words below:")

print ("=======================\n")

print (len(stop\_words))

print ("=======================\n")

print("See Filtered Sentence list below:")

print ("=======================\n")

print("Filtered Sentence : ", filtered\_sentence)

print ("=======================\n")

print("See the NO.# of Filtered Sentences below:")

print ("=======================\n")

print (len(filtered\_sentence))

print ("=======================\n")

**OUTPUT:**

See Tokenized words list below:

=======================

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

=======================

See the No.# of words below:

=======================

179

=======================

See Stop words list below:

=======================

['Stop', 'words', 'are', 'most', 'common', 'words', 'found', 'in', 'any', 'natural', 'language', 'which', 'carries', 'very', 'little', 'or', 'no', 'significant', 'semantic', 'context', 'in', 'a', 'sentence', '.', 'It', 'just', 'carry', 'syntactic', 'importance', 'which', 'aid', 'in', 'formation', 'of', 'sentence', '.', 'As', 'a', 'preprocessing', 'operation', 'it', 'must', 'be', 'removed', 'to', 'ease', 'further', 'task', 'and', 'speedup', 'core', 'task', 'in', 'text', 'processing', '.', 'Ibrahim', 'A', '[', '3', ']', 'conducted', 'a', 'comparative', 'study', 'on', 'the', 'effect', 'of', 'stop', 'words', 'elimination', 'on', 'Arabic', 'Information', 'Retrieval', 'where', 'three', 'stop', 'lists', 'viz', ',', 'General', 'Stop', 'list', ',', 'corpus', 'based', 'stop-list', 'and', 'combined', 'stop', 'list', 'were', 'used', 'for', 'comparative', 'study', '.', 'General', 'stop-list', 'performed', 'better', 'than', 'the', 'rest', 'of', 'the', 'two', '.', 'Ashish', 'T', ',', 'et', 'al', '[', '4', ']', 'eliminated', 'stop-word', 'in', 'Gujarati', 'language', 'by', 'preparing', 'frequency', 'list', 'from', 'Gujarati', 'corpus', 'by', 'analyzing', 'popular', 'Gujarati', 'newspapers', '.', 'Riyad', 'A', ',', 'et', 'al', '[', '5', ']', ',', 'used', 'Finite', 'State', 'Machine', '(', 'FSM', ')', 'algorithm', 'to', 'eliminate', 'stop-words', 'for', 'Arabic', 'Language', '.', 'Basim', 'A', ',', 'et', 'al', '[', '6', ']', 'have', 'designed', 'and', 'implemented', 'a', 'new', 'stop-word', 'removal', 'technique', 'for', 'Arabic', 'language', 'based', 'on', 'dedicated', 'list', 'and', 'algorithm', 'which', 'compares', 'stopwords', 'if', 'it', 'fulfills', 'desired', 'string', 'length', 'criteria', '.', 'Vijayarani', 'S', ',', 'et', 'al', '[', '7', ']', 'used', 'Zipf', '’', 's', 'Law', '(', 'Z', 'method', ')', 'for', 'creation', 'of', 'stop-words', '.', 'Rakholia', 'and', 'Saini', '[', '8', ']', 'have', 'presented', 'a', 'rule-based', 'approach', 'to', 'dynamically', 'identify', 'stop', 'words', 'for', 'Gujarati', 'language', '.', 'They', 'have', 'also', 'deployed', 'this', 'approach', 'with', 'additional', 'cosine', 'similarity', 'based', 'Vector', 'Space', 'Model', 'for', 'information', 'retrieval', 'in', 'Gujarati', 'language', '[', '9', ']', '.', 'Kaur', 'J', 'and', 'Saini', 'JR', 'have', 'presented', 'the', 'list', 'of', 'Punjabi', 'stop', 'words', '[', '10', ']', ',', 'its', 'Partof-Speech', 'class', 'based', 'classification', '[', '11', ']', 'and', 'its', 'Gurumukhi', 'and', 'Shahmukhi', 'script', 'versions', '[', '12', ']', '.', 'Saini', 'and', 'Rakholia', '[', '13', ']', 'have', 'presented', 'an', 'analytic', 'in-depth', 'report', 'on', 'continent', 'and', 'script-wise', 'divisions-based', 'statistical', 'measures', 'for', 'stopwords', 'lists', 'of', 'various', 'international', 'Languages', '.']

=======================

See the No.# of Stop words below:

=======================

326

=======================

See Tokenize words list below:

=======================

Word Tokens : ['Stop', 'words', 'are', 'most', 'common', 'words', 'found', 'in', 'any', 'natural', 'language', 'which', 'carries', 'very', 'little', 'or', 'no', 'significant', 'semantic', 'context', 'in', 'a', 'sentence', '.', 'It', 'just', 'carry', 'syntactic', 'importance', 'which', 'aid', 'in', 'formation', 'of', 'sentence', '.', 'As', 'a', 'preprocessing', 'operation', 'it', 'must', 'be', 'removed', 'to', 'ease', 'further', 'task', 'and', 'speedup', 'core', 'task', 'in', 'text', 'processing', '.', 'Ibrahim', 'A', '[', '3', ']', 'conducted', 'a', 'comparative', 'study', 'on', 'the', 'effect', 'of', 'stop', 'words', 'elimination', 'on', 'Arabic', 'Information', 'Retrieval', 'where', 'three', 'stop', 'lists', 'viz', ',', 'General', 'Stop', 'list', ',', 'corpus', 'based', 'stop-list', 'and', 'combined', 'stop', 'list', 'were', 'used', 'for', 'comparative', 'study', '.', 'General', 'stop-list', 'performed', 'better', 'than', 'the', 'rest', 'of', 'the', 'two', '.', 'Ashish', 'T', ',', 'et', 'al', '[', '4', ']', 'eliminated', 'stop-word', 'in', 'Gujarati', 'language', 'by', 'preparing', 'frequency', 'list', 'from', 'Gujarati', 'corpus', 'by', 'analyzing', 'popular', 'Gujarati', 'newspapers', '.', 'Riyad', 'A', ',', 'et', 'al', '[', '5', ']', ',', 'used', 'Finite', 'State', 'Machine', '(', 'FSM', ')', 'algorithm', 'to', 'eliminate', 'stop-words', 'for', 'Arabic', 'Language', '.', 'Basim', 'A', ',', 'et', 'al', '[', '6', ']', 'have', 'designed', 'and', 'implemented', 'a', 'new', 'stop-word', 'removal', 'technique', 'for', 'Arabic', 'language', 'based', 'on', 'dedicated', 'list', 'and', 'algorithm', 'which', 'compares', 'stopwords', 'if', 'it', 'fulfills', 'desired', 'string', 'length', 'criteria', '.', 'Vijayarani', 'S', ',', 'et', 'al', '[', '7', ']', 'used', 'Zipf', '’', 's', 'Law', '(', 'Z', 'method', ')', 'for', 'creation', 'of', 'stop-words', '.', 'Rakholia', 'and', 'Saini', '[', '8', ']', 'have', 'presented', 'a', 'rule-based', 'approach', 'to', 'dynamically', 'identify', 'stop', 'words', 'for', 'Gujarati', 'language', '.', 'They', 'have', 'also', 'deployed', 'this', 'approach', 'with', 'additional', 'cosine', 'similarity', 'based', 'Vector', 'Space', 'Model', 'for', 'information', 'retrieval', 'in', 'Gujarati', 'language', '[', '9', ']', '.', 'Kaur', 'J', 'and', 'Saini', 'JR', 'have', 'presented', 'the', 'list', 'of', 'Punjabi', 'stop', 'words', '[', '10', ']', ',', 'its', 'Partof-Speech', 'class', 'based', 'classification', '[', '11', ']', 'and', 'its', 'Gurumukhi', 'and', 'Shahmukhi', 'script', 'versions', '[', '12', ']', '.', 'Saini', 'and', 'Rakholia', '[', '13', ']', 'have', 'presented', 'an', 'analytic', 'in-depth', 'report', 'on', 'continent', 'and', 'script-wise', 'divisions-based', 'statistical', 'measures', 'for', 'stopwords', 'lists', 'of', 'various', 'international', 'Languages', '.']

=======================

See the NO.# of Stop words below:

=======================

179

=======================

See Filtered Sentence list below:

=======================

Filtered Sentence : ['Stop', 'words', 'common', 'words', 'found', 'natural', 'language', 'carries', 'little', 'significant', 'semantic', 'context', 'sentence', '.', 'It', 'carry', 'syntactic', 'importance', 'aid', 'formation', 'sentence', '.', 'As', 'preprocessing', 'operation', 'must', 'removed', 'ease', 'task', 'speedup', 'core', 'task', 'text', 'processing', '.', 'Ibrahim', 'A', '[', '3', ']', 'conducted', 'comparative', 'study', 'effect', 'stop', 'words', 'elimination', 'Arabic', 'Information', 'Retrieval', 'three', 'stop', 'lists', 'viz', ',', 'General', 'Stop', 'list', ',', 'corpus', 'based', 'stop-list', 'combined', 'stop', 'list', 'used', 'comparative', 'study', '.', 'General', 'stop-list', 'performed', 'better', 'rest', 'two', '.', 'Ashish', 'T', ',', 'et', 'al', '[', '4', ']', 'eliminated', 'stop-word', 'Gujarati', 'language', 'preparing', 'frequency', 'list', 'Gujarati', 'corpus', 'analyzing', 'popular', 'Gujarati', 'newspapers', '.', 'Riyad', 'A', ',', 'et', 'al', '[', '5', ']', ',', 'used', 'Finite', 'State', 'Machine', '(', 'FSM', ')', 'algorithm', 'eliminate', 'stop-words', 'Arabic', 'Language', '.', 'Basim', 'A', ',', 'et', 'al', '[', '6', ']', 'designed', 'implemented', 'new', 'stop-word', 'removal', 'technique', 'Arabic', 'language', 'based', 'dedicated', 'list', 'algorithm', 'compares', 'stopwords', 'fulfills', 'desired', 'string', 'length', 'criteria', '.', 'Vijayarani', 'S', ',', 'et', 'al', '[', '7', ']', 'used', 'Zipf', '’', 'Law', '(', 'Z', 'method', ')', 'creation', 'stop-words', '.', 'Rakholia', 'Saini', '[', '8', ']', 'presented', 'rule-based', 'approach', 'dynamically', 'identify', 'stop', 'words', 'Gujarati', 'language', '.', 'They', 'also', 'deployed', 'approach', 'additional', 'cosine', 'similarity', 'based', 'Vector', 'Space', 'Model', 'information', 'retrieval', 'Gujarati', 'language', '[', '9', ']', '.', 'Kaur', 'J', 'Saini', 'JR', 'presented', 'list', 'Punjabi', 'stop', 'words', '[', '10', ']', ',', 'Partof-Speech', 'class', 'based', 'classification', '[', '11', ']', 'Gurumukhi', 'Shahmukhi', 'script', 'versions', '[', '12', ']', '.', 'Saini', 'Rakholia', '[', '13', ']', 'presented', 'analytic', 'in-depth', 'report', 'continent', 'script-wise', 'divisions-based', 'statistical', 'measures', 'stopwords', 'lists', 'various', 'international', 'Languages', '.']

=======================

See the NO.# of Filtered Sentences below:

=======================

249

=======================

>>>

RESTART: C:\Users\DELL\Desktop\IT Bootcamp\_Training\NLP\Assignment\word\_tokenizer\_q2.py

This is Word Tokenization

['The', 'wolf', 'said', ':', '``', 'Little', 'pig', ',', 'little', 'pig', ',', 'let', 'me', 'come', 'in', '.', "''"]

17

This is Space Tokenization

['The', 'wolf', 'said:', '"Little', 'pig,', 'little', 'pig,', 'let', 'me', 'come', 'in."']

11

>>>

RESTART: C:\Users\DELL\Desktop\IT Bootcamp\_Training\NLP\Assignment\tokenize\_stopwords\_q6.py

See Tokenized words list below:

=======================

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

=======================

See the No.# of words below:

=======================

179

=======================

See Stop words list below:

=======================

['Stop', 'words', 'are', 'most', 'common', 'words', 'found', 'in', 'any', 'natural', 'language', 'which', 'carries', 'very', 'little', 'or', 'no', 'significant', 'semantic', 'context', 'in', 'a', 'sentence', '.', 'It', 'just', 'carry', 'syntactic', 'importance', 'which', 'aid', 'in', 'formation', 'of', 'sentence', '.', 'As', 'a', 'preprocessing', 'operation', 'it', 'must', 'be', 'removed', 'to', 'ease', 'further', 'task', 'and', 'speedup', 'core', 'task', 'in', 'text', 'processing', '.', 'Ibrahim', 'A', '[', '3', ']', 'conducted', 'a', 'comparative', 'study', 'on', 'the', 'effect', 'of', 'stop', 'words', 'elimination', 'on', 'Arabic', 'Information', 'Retrieval', 'where', 'three', 'stop', 'lists', 'viz', ',', 'General', 'Stop', 'list', ',', 'corpus', 'based', 'stop-list', 'and', 'combined', 'stop', 'list', 'were', 'used', 'for', 'comparative', 'study', '.', 'General', 'stop-list', 'performed', 'better', 'than', 'the', 'rest', 'of', 'the', 'two', '.', 'Ashish', 'T', ',', 'et', 'al', '[', '4', ']', 'eliminated', 'stop-word', 'in', 'Gujarati', 'language', 'by', 'preparing', 'frequency', 'list', 'from', 'Gujarati', 'corpus', 'by', 'analyzing', 'popular', 'Gujarati', 'newspapers', '.', 'Riyad', 'A', ',', 'et', 'al', '[', '5', ']', ',', 'used', 'Finite', 'State', 'Machine', '(', 'FSM', ')', 'algorithm', 'to', 'eliminate', 'stop-words', 'for', 'Arabic', 'Language', '.', 'Basim', 'A', ',', 'et', 'al', '[', '6', ']', 'have', 'designed', 'and', 'implemented', 'a', 'new', 'stop-word', 'removal', 'technique', 'for', 'Arabic', 'language', 'based', 'on', 'dedicated', 'list', 'and', 'algorithm', 'which', 'compares', 'stopwords', 'if', 'it', 'fulfills', 'desired', 'string', 'length', 'criteria', '.', 'Vijayarani', 'S', ',', 'et', 'al', '[', '7', ']', 'used', 'Zipf', '’', 's', 'Law', '(', 'Z', 'method', ')', 'for', 'creation', 'of', 'stop-words', '.', 'Rakholia', 'and', 'Saini', '[', '8', ']', 'have', 'presented', 'a', 'rule-based', 'approach', 'to', 'dynamically', 'identify', 'stop', 'words', 'for', 'Gujarati', 'language', '.', 'They', 'have', 'also', 'deployed', 'this', 'approach', 'with', 'additional', 'cosine', 'similarity', 'based', 'Vector', 'Space', 'Model', 'for', 'information', 'retrieval', 'in', 'Gujarati', 'language', '[', '9', ']', '.', 'Kaur', 'J', 'and', 'Saini', 'JR', 'have', 'presented', 'the', 'list', 'of', 'Punjabi', 'stop', 'words', '[', '10', ']', ',', 'its', 'Partof-Speech', 'class', 'based', 'classification', '[', '11', ']', 'and', 'its', 'Gurumukhi', 'and', 'Shahmukhi', 'script', 'versions', '[', '12', ']', '.', 'Saini', 'and', 'Rakholia', '[', '13', ']', 'have', 'presented', 'an', 'analytic', 'in-depth', 'report', 'on', 'continent', 'and', 'script-wise', 'divisions-based', 'statistical', 'measures', 'for', 'stopwords', 'lists', 'of', 'various', 'international', 'Languages', '.']

=======================

See the No.# of Stop words below:

=======================

326

=======================

See Tokenize words list below:

=======================

Word Tokens : ['Stop', 'words', 'are', 'most', 'common', 'words', 'found', 'in', 'any', 'natural', 'language', 'which', 'carries', 'very', 'little', 'or', 'no', 'significant', 'semantic', 'context', 'in', 'a', 'sentence', '.', 'It', 'just', 'carry', 'syntactic', 'importance', 'which', 'aid', 'in', 'formation', 'of', 'sentence', '.', 'As', 'a', 'preprocessing', 'operation', 'it', 'must', 'be', 'removed', 'to', 'ease', 'further', 'task', 'and', 'speedup', 'core', 'task', 'in', 'text', 'processing', '.', 'Ibrahim', 'A', '[', '3', ']', 'conducted', 'a', 'comparative', 'study', 'on', 'the', 'effect', 'of', 'stop', 'words', 'elimination', 'on', 'Arabic', 'Information', 'Retrieval', 'where', 'three', 'stop', 'lists', 'viz', ',', 'General', 'Stop', 'list', ',', 'corpus', 'based', 'stop-list', 'and', 'combined', 'stop', 'list', 'were', 'used', 'for', 'comparative', 'study', '.', 'General', 'stop-list', 'performed', 'better', 'than', 'the', 'rest', 'of', 'the', 'two', '.', 'Ashish', 'T', ',', 'et', 'al', '[', '4', ']', 'eliminated', 'stop-word', 'in', 'Gujarati', 'language', 'by', 'preparing', 'frequency', 'list', 'from', 'Gujarati', 'corpus', 'by', 'analyzing', 'popular', 'Gujarati', 'newspapers', '.', 'Riyad', 'A', ',', 'et', 'al', '[', '5', ']', ',', 'used', 'Finite', 'State', 'Machine', '(', 'FSM', ')', 'algorithm', 'to', 'eliminate', 'stop-words', 'for', 'Arabic', 'Language', '.', 'Basim', 'A', ',', 'et', 'al', '[', '6', ']', 'have', 'designed', 'and', 'implemented', 'a', 'new', 'stop-word', 'removal', 'technique', 'for', 'Arabic', 'language', 'based', 'on', 'dedicated', 'list', 'and', 'algorithm', 'which', 'compares', 'stopwords', 'if', 'it', 'fulfills', 'desired', 'string', 'length', 'criteria', '.', 'Vijayarani', 'S', ',', 'et', 'al', '[', '7', ']', 'used', 'Zipf', '’', 's', 'Law', '(', 'Z', 'method', ')', 'for', 'creation', 'of', 'stop-words', '.', 'Rakholia', 'and', 'Saini', '[', '8', ']', 'have', 'presented', 'a', 'rule-based', 'approach', 'to', 'dynamically', 'identify', 'stop', 'words', 'for', 'Gujarati', 'language', '.', 'They', 'have', 'also', 'deployed', 'this', 'approach', 'with', 'additional', 'cosine', 'similarity', 'based', 'Vector', 'Space', 'Model', 'for', 'information', 'retrieval', 'in', 'Gujarati', 'language', '[', '9', ']', '.', 'Kaur', 'J', 'and', 'Saini', 'JR', 'have', 'presented', 'the', 'list', 'of', 'Punjabi', 'stop', 'words', '[', '10', ']', ',', 'its', 'Partof-Speech', 'class', 'based', 'classification', '[', '11', ']', 'and', 'its', 'Gurumukhi', 'and', 'Shahmukhi', 'script', 'versions', '[', '12', ']', '.', 'Saini', 'and', 'Rakholia', '[', '13', ']', 'have', 'presented', 'an', 'analytic', 'in-depth', 'report', 'on', 'continent', 'and', 'script-wise', 'divisions-based', 'statistical', 'measures', 'for', 'stopwords', 'lists', 'of', 'various', 'international', 'Languages', '.']

=======================

See the NO.# of Stop words below:

=======================

179

=======================

See Filtered Sentence list below:

=======================

Filtered Sentence : ['Stop', 'words', 'common', 'words', 'found', 'natural', 'language', 'carries', 'little', 'significant', 'semantic', 'context', 'sentence', '.', 'It', 'carry', 'syntactic', 'importance', 'aid', 'formation', 'sentence', '.', 'As', 'preprocessing', 'operation', 'must', 'removed', 'ease', 'task', 'speedup', 'core', 'task', 'text', 'processing', '.', 'Ibrahim', 'A', '[', '3', ']', 'conducted', 'comparative', 'study', 'effect', 'stop', 'words', 'elimination', 'Arabic', 'Information', 'Retrieval', 'three', 'stop', 'lists', 'viz', ',', 'General', 'Stop', 'list', ',', 'corpus', 'based', 'stop-list', 'combined', 'stop', 'list', 'used', 'comparative', 'study', '.', 'General', 'stop-list', 'performed', 'better', 'rest', 'two', '.', 'Ashish', 'T', ',', 'et', 'al', '[', '4', ']', 'eliminated', 'stop-word', 'Gujarati', 'language', 'preparing', 'frequency', 'list', 'Gujarati', 'corpus', 'analyzing', 'popular', 'Gujarati', 'newspapers', '.', 'Riyad', 'A', ',', 'et', 'al', '[', '5', ']', ',', 'used', 'Finite', 'State', 'Machine', '(', 'FSM', ')', 'algorithm', 'eliminate', 'stop-words', 'Arabic', 'Language', '.', 'Basim', 'A', ',', 'et', 'al', '[', '6', ']', 'designed', 'implemented', 'new', 'stop-word', 'removal', 'technique', 'Arabic', 'language', 'based', 'dedicated', 'list', 'algorithm', 'compares', 'stopwords', 'fulfills', 'desired', 'string', 'length', 'criteria', '.', 'Vijayarani', 'S', ',', 'et', 'al', '[', '7', ']', 'used', 'Zipf', '’', 'Law', '(', 'Z', 'method', ')', 'creation', 'stop-words', '.', 'Rakholia', 'Saini', '[', '8', ']', 'presented', 'rule-based', 'approach', 'dynamically', 'identify', 'stop', 'words', 'Gujarati', 'language', '.', 'They', 'also', 'deployed', 'approach', 'additional', 'cosine', 'similarity', 'based', 'Vector', 'Space', 'Model', 'information', 'retrieval', 'Gujarati', 'language', '[', '9', ']', '.', 'Kaur', 'J', 'Saini', 'JR', 'presented', 'list', 'Punjabi', 'stop', 'words', '[', '10', ']', ',', 'Partof-Speech', 'class', 'based', 'classification', '[', '11', ']', 'Gurumukhi', 'Shahmukhi', 'script', 'versions', '[', '12', ']', '.', 'Saini', 'Rakholia', '[', '13', ']', 'presented', 'analytic', 'in-depth', 'report', 'continent', 'script-wise', 'divisions-based', 'statistical', 'measures', 'stopwords', 'lists', 'various', 'international', 'Languages', '.']

=======================

See the NO.# of Filtered Sentences below:

=======================

249

=======================

>>>