**Text Pre-processing Using NLTK**

##### Remove Stop Words

1. **Implementing Stemming**

##### Prerequisites:

* + Install nltk library.

##### a) Remove Stop Words

# Import libraries

#import LIbraries import nltk

nltk.download("stopwords")

from nltk.tokenize import sent\_tokenize from nltk.tokenize import word\_tokenize from nltk.corpus import stopwords

# input

text= """Natural Language Processing with Python provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more. The online version of the book has been been updated for Python 3 and NLTK 3."""

# paragraph to sentence division text\_to\_sentence = sent\_tokenize(text) print("Paragraph to Sentences") print(text\_to\_sentence)

print("Number of sentences:",len(text\_to\_sentence))

# word division

tokenized\_word = word\_tokenize(text) print("\nList of Words",tokenized\_word) print("Number of Words",len(tokenized\_word))

# Stop word removal

sw\_nltk = stopwords.words('english') print("\nList of all stop words\n:",sw\_nltk) print("Number of stop words",len(sw\_nltk))

words = [word for word in text.split() if word.lower() not in sw\_nltk] new\_text = " ".join(words)

print(new\_text)

print("\nOld length: ", len(text)) print("\nNew length: ", len(new\_text)) print("\nActual Text\n:",text)

print("\n Text after removing Stop words:\n",new\_text)

##### Output:

Paragraph to Sentences

['Natural Language Processing with Python provides a practical introduction to programming for language processing.', 'Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more.', 'The online version of the book has been been updated for Python 3 and NLTK 3.']

Number of sentences: 3

List of Words ['Natural', 'Language', 'Processing', 'with', 'Python', 'provides', 'a', 'practical', 'introduction', 'to', 'programming', 'for', 'language', 'processing', '.', 'Written', 'by', 'the', 'creators', 'of', 'NLTK', ',', 'it', 'guides',

'the', 'reader', 'through', 'the', 'fundamentals', 'of', 'writing', 'Python', 'programs', ',', 'working', 'with', 'corpora',

',', 'categorizing', 'text', ',', 'analyzing', 'linguistic', 'structure', ',', 'and', 'more', '.', 'The', 'online', 'version', 'of',

'the', 'book', 'has', 'been', 'been', 'updated', 'for', 'Python', '3', 'and', 'NLTK', '3', '.'] Number of Words 65

List of all stop words

: ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours',

'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they',

'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those', 'am', 'is',

'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but',

'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during',

'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then',

'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some',

'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should',

"should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn',

"doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn',

"mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]

Number of stop words 179

Natural Language Processing Python provides practical introduction programming language processing. Written creators NLTK, guides reader fundamentals writing Python programs, working corpora, categorizing text, analyzing linguistic structure, more. online version book updated Python 3 NLTK 3.

Old length: 381

New length: 293 Actual Text

: Natural Language Processing with Python provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more. The online version of the book has been been updated for Python 3 and NLTK 3.

Text after removing Stop words:

Natural Language Processing Python provides practical introduction programming language processing. Written creators NLTK, guides reader fundamentals writing Python programs, working corpora, categorizing text, analyzing linguistic structure, more. online version book updated Python 3 NLTK 3.

##### B) Implementing Stemming

#importing the Stemming function from nltk library from nltk.stem.porter import PorterStemmer #defining a function for stemming

from nltk.stem import PorterStemmer from nltk.stem import SnowballStemmer

snowball = SnowballStemmer(language='english')

#defining the object for stemming

text= """Natural Language Processing with Python provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more. The online version of the book has been been updated for Python 3 and NLTK 3."""

stemmer = PorterStemmer() def stem\_words(text):

return " ".join([stemmer.stem(word) for word in text.split()]) stemmed\_data=stem\_words(text)

print("Actual text\n",text) print("stemmed text\n:" ,stemmed\_data)

##### Output:

Actual text

Natural Language Processing with Python provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more. The online version of the book has been been updated for Python 3 and NLTK 3.

stemmed text

: natur languag process with python provid a practic introduct to program for languag processing. written by the creator of nltk, it guid the reader through the fundament of write python programs, work with corpora, categor text, analyz linguist structure, and more. the onlin version of the book ha been been updat for python 3 and nltk 3.

##### Text processing using NLTK

* **POS (Parts of Speech) tagging**

**Parts of Speech (POS) Tagging**

#import Libraries

#import Libraries import nltk

nltk.download('averaged\_perceptron\_tagger')

text= """Natural Language Processing with Python provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more. The online version of the book has been been updated for Python 3 and NLTK 3."""

tokens = nltk.word\_tokenize(text) print(tokens)

tag = nltk.pos\_tag(tokens) print(tag)

##### Output:

['Natural', 'Language', 'Processing', 'with', 'Python', 'provides', 'a', 'practical', 'introduction', 'to', 'programming', 'for', 'language', 'processing', '.', 'Written', 'by', 'the', 'creators', 'of', 'NLTK', ',', 'it', 'guides', 'the', 'reader', 'through',

'the', 'fundamentals', 'of', 'writing', 'Python', 'programs', ',', 'working', 'with', 'corpora', ',', 'categorizing', 'text', ',',

'analyzing', 'linguistic', 'structure', ',', 'and', 'more', '.', 'The', 'online', 'version', 'of', 'the', 'book', 'has', 'been', 'been', 'updated', 'for', 'Python', '3', 'and', 'NLTK', '3', '.']

[('Natural', 'JJ'), ('Language', 'NNP'), ('Processing', 'NNP'), ('with', 'IN'), ('Python', 'NNP'), ('provides', 'VBZ'),

('a', 'DT'), ('practical', 'JJ'), ('introduction', 'NN'), ('to', 'TO'), ('programming', 'VBG'), ('for', 'IN'), ('language',

'NN'), ('processing', 'NN'), ('.', '.'), ('Written', 'VBN'), ('by', 'IN'), ('the', 'DT'), ('creators', 'NNS'), ('of', 'IN'),

('NLTK', 'NNP'), (',', ','), ('it', 'PRP'), ('guides', 'VBZ'), ('the', 'DT'), ('reader', 'NN'), ('through', 'IN'), ('the', 'DT'),

('fundamentals', 'NNS'), ('of', 'IN'), ('writing', 'VBG'), ('Python', 'NNP'), ('programs', 'NNS'), (',', ','), ('working',

'VBG'), ('with', 'IN'), ('corpora', 'NNS'), (',', ','), ('categorizing', 'VBG'), ('text', 'NN'), (',', ','), ('analyzing', 'VBG'),

('linguistic', 'JJ'), ('structure', 'NN'), (',', ','), ('and', 'CC'), ('more', 'RBR'), ('.', '.'), ('The', 'DT'), ('online', 'JJ'),

('version', 'NN'), ('of', 'IN'), ('the', 'DT'), ('book', 'NN'), ('has', 'VBZ'), ('been', 'VBN'), ('been', 'VBN'), ('updated',

'VBN'), ('for', 'IN'), ('Python', 'NNP'), ('3', 'CD'), ('and', 'CC'), ('NLTK', 'NNP'), ('3', 'CD'), ('.', '.')]