Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

AY:2024-25

Class:	**BE**	**Semester:**	**VII**
Course Code:	**CSDOL7011**	**Course Name:**	**Natural Language Processing**

Name of Student:	**Parth Raut**
Roll No.:	**40**
Experiment No.:	**2**
Title of the Experiment:	**Text Preprocessing Techniques: Tokenization & Filtration**
Date of Performance:	
Date of Submission:	

Evaluation

Performance Indicator	Max. Marks	Marks Obtained
Performance	5	
Understanding	5	
Journal work and timely submission	10	
Total	20	

Performance Indicator	Exceed Expectations (EE)	Meet Expectations (ME)	Below Expectations (BE)
Performance	4-5	2-3	1
Understanding	4-5	2-3	1
Journal work and timely submission	8-10	5-8	1-4

Checked by

Name of Faculty :

Signature :

Date :

Experiment 2

Aim: Apply various text preprocessing techniques for any given text: Tokenization and Filtration & Script Validation.

Objective: Able to perform sentence and word tokenization for the given input text for English and Indian Language.

Theory:

Tokenization is one of the first step in any NLP pipeline. Tokenization is nothing but splitting the raw text into small chunks of words or sentences, called tokens. If the text is split into words, then its called as 'Word Tokenization' and if it's split into sentences then its called as 'Sentence Tokenization'. Generally 'space' is used to perform the word tokenization and characters like 'periods, exclamation point and newline char are used for Sentence Tokenization. We have to choose the appropriate method as per the task in hand. While performing the tokenization few characters like spaces, punctuations are ignored and will not be the part of final list of tokens.

Why Tokenization is Required?

Every sentence gets its meaning by the words present in it. So by analyzing the words present in the text we can easily interpret the meaning of the text. Once we have a list of words we can also use statistical tools and methods to get more insights into the text. For example, we can use word count and word frequency to find out important of word in that sentence or document.

Input Text

Tokenization is one of the first step in any NLP pipeline. Tokenization is nothing but splitting the raw text into small chunks of words or sentences, called tokens.

Word Tokenization

Tokenization	is	one	of
the	first	step	in
any	NLP	pipeline	Tokenization
is	nothing	but	splitting
the	raw	text	into
small	chunks	of	words
or	sentences	called	tokens

Sentence Tokenization

Tokenization is one of the first step in any NLP pipeline

Tokenization is nothing but splitting the raw text into small chunks of words or sentences, called tokens

Code:

In [1]:
```
!pip install nltk
```

```
Requirement already satisfied: nltk in c:\users\admin\appdata\local\programs\python\python37\lib\site-packages (3.6.2)
Requirement already satisfied: tqdm in c:\users\admin\appdata\local\programs\python\python37\lib\site-packages (from nltk)
(4.60.0)
Requirement already satisfied: joblib in c:\users\admin\appdata\local\programs\python\python37\lib\site-packages (from nltk)
(1.0.0)
Requirement already satisfied: click in c:\users\admin\appdata\local\programs\python\python37\lib\site-packages (from nltk)
(7.1.2)
Requirement already satisfied: regex in c:\users\admin\appdata\local\programs\python\python37\lib\site-packages (from nltk)
(2021.4.4)
WARNING: You are using pip version 22.0; however, version 23.2.1 is available.
You should consider upgrading via the 'c:\users\admin\appdata\local\programs\python\python37\python.exe -m pip install --upgr
ade pip' command.
```

In [2]:
```
import nltk
```

In [3]:
```
nltk.download()
```

```
showing info https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/index.xml
```

Out[3]:
```
True
```

Sentence Tokenization

In [4]:
```
from nltk.tokenize import sent_tokenize
```

In [15]:
```
text = '''Stephenson 2-18 is now known as being one of the largest, if not the current largest star ever discovered, surpass
        Stephenson 2-18 has a radius of 2,150 solar radii, being larger than almost the entire orbit of Saturn (1,940 - 2,16
```

In [16]:
```
text
```

Out[16]:
```
'Stephenson 2-18 is now known as being one of the largest, if not the current largest star ever discovered, surpassing other
stars like VY Canis Majoris and UY Scuti.\n        Stephenson 2-18 has a radius of 2,150 solar radii, being larger than almo
st the entire orbit of Saturn (1,940 - 2,169 solar radii).'
```

In [17]:
```
sentences = sent_tokenize (text)
```

In [18]:
```
sentences
```

Out[18]:
```
['Stephenson 2-18 is now known as being one of the largest, if not the current largest star ever discovered, surpassing othe
r stars like VY Canis Majoris and UY Scuti.',
 'Stephenson 2-18 has a radius of 2,150 solar radii, being larger than almost the entire orbit of Saturn (1,940 - 2,169 sola
r radii).']
```

Word Tokenization

In [19]:
```
from nltk.tokenize import word_tokenize
```

In [20]:
```
words = word_tokenize (text)
```

In [21]:
```
words
```

Out[21]:
```
['Stephenson',
 '2-18',
 'is',
 'now',
 'known',
 'as',
 'being',
 'one',
 'of',
 'the',
 'largest',
 ',',
 'if',
 'not',
 'the',
 'current',
 'largest',
 'star',
 'ever',
 'discovered',
 ',',
 'surpassing',
 'other',
 'stars',
 'like',
 'VY',
 'Canis',
 'Majoris',
 'and',
 'UY',
 'Scuti',
 '.',
 'Stephenson',
 '2-18',
 'has',
 'a',
 'radius',
 'of',
 '2,150',
```

Levels of Sentences Tokenization using Comprehension

In [23]:
```
sent_tokenize (text)
```

Out[23]:
```
['Stephenson 2-18 is now known as being one of the largest, if not the current largest star ever discovered, surpassing othe
r stars like VY Canis Majoris and UY Scuti.',
 'Stephenson 2-18 has a radius of 2,150 solar radii, being larger than almost the entire orbit of Saturn (1,940 - 2,169 sola
r radii).']
```

In [24]:
```
[word_tokenize (text) for t in sent_tokenize(text)]
```

Out[24]:
```
[['Stephenson',
  '2-18',
  'is',
  'now',
  'known',
  'as',
  'being',
  'one',
  'of',
  'the',
  'largest',
  ',',
  'if',
  'not',
  'the',
  'current',
  'largest',
  'star',
  'ever',
  'discovered',
  ',',
  'surpassing',
  'other',
  'stars',
  'like',
```

```
In [25]: from nltk.tokenize import wordpunct_tokenize

In [26]: wordpunct_tokenize (text)

Out[26]: ['Stephenson',
          '2',
          '-',
          '18',
          'is',
          'now',
          'known',
          'as',
          'being',
          'one',
          'of',
          'the',
          'largest',
          ',',
          'if',
          'not',
          'the',
          'current',
          'largest',
          'star',
          'ever',
          'discovered',
          ',',
          'surpassing',
```

Filteration of Text by converting into lower case

```
In [27]: text.lower()

Out[27]: 'stephenson 2-18 is now known as being one of the largest, if not the current largest star ever discovered, surpassing other
         stars like vy canis majoris and uy scuti.\n        stephenson 2-18 has a radius of 2,150 solar radii, being larger than almo
         st the entire orbit of saturn (1,940 - 2,169 solar radii).'

In [28]: text.upper()

Out[28]: 'STEPHENSON 2-18 IS NOW KNOWN AS BEING ONE OF THE LARGEST, IF NOT THE CURRENT LARGEST STAR EVER DISCOVERED, SURPASSING OTHER
         STARS LIKE VY CANIS MAJORIS AND UY SCUTI.\n        STEPHENSON 2-18 HAS A RADIUS OF 2,150 SOLAR RADII, BEING LARGER THAN ALMO
         ST THE ENTIRE ORBIT OF SATURN (1,940 - 2,169 SOLAR RADII).'
```

Conclusion:

Tokenization is a crucial step in natural language processing (NLP) that involves breaking down text into smaller units, or tokens, such as words, phrases, or sentences. Here are some commonly used tools and techniques for tokenization:

1. NLTK (Natural Language Toolkit): A comprehensive library in Python that provides functions for tokenizing text into words and sentences using various algorithms.
2. spaCy: A fast and efficient NLP library that offers robust tokenization capabilities, handling punctuation and special characters effectively.
3. Transformers by Hugging Face: This library includes tokenizers specifically designed for transformer models, like BERT and GPT, ensuring that input is correctly processed for model compatibility.