Experiment No.2
Apply various text preprocessing techniques for any given text: Tokenization and Filtration & Script Validation.
Date of Performance:
Date of Submission:

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

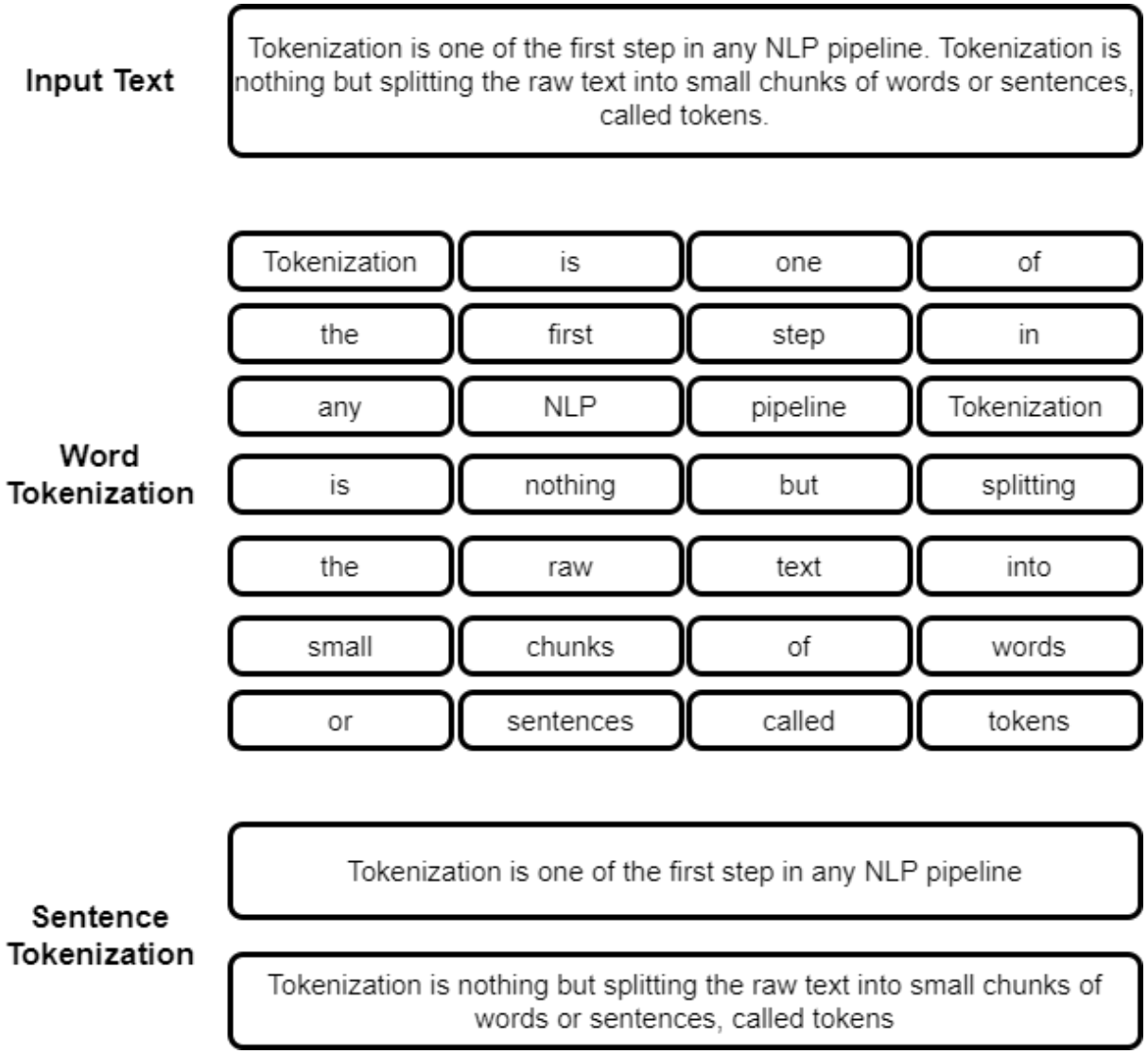
Objective: Able to perform sentence and word tokenization for the given input text for English and Indian Language.

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.



Code:

```
import nltk
# Download necessary NLTK resources (uncomment if you haven't downloaded them yet)
# nltk.download('punkt')
### Sentence Tokenization
from nltk.tokenize import sent_tokenize
text = "'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 solar radii, being larger than almost the entire orbit of Saturn (1,940 - 2,169 solar radii).'"
print("Plaintext: ", text)
# Tokenizing the text into sentences
sentences = sent_tokenize(text)
print("Tokenized sentences: ", sentences)
```

```
### Word Tokenization
from nltk.tokenize import word_tokenize
# Tokenizing the text into words
words = word_tokenize(text)
print("Tokenized words: ", words)
# Printing each tokenized word
for w in words:
    print(w)
### Levels of Sentence Tokenization using List Comprehension
# Tokenizing sentences and then tokenizing each sentence into words
tokenized_sentences = [word_tokenize(t) for t in sent_tokenize(text)]
print("Tokenized sentences into words: ", tokenized_sentences)
### Alternative Word Tokenization
from nltk.tokenize import wordpunct_tokenize
# Tokenizing text using word punct tokenization
wordpunct_tokens = wordpunct_tokenize(text)
print("Word punct tokenization: ", wordpunct_tokens)
### Filtering Text by converting into lower case
lowercase_text = text.lower()
print("Lowercase text: ", lowercase_text)
# If you want to see the uppercase version as well
uppercase_text = text.upper()
print("Uppercase text: ", uppercase_text)
```

Output:

```
(venv) PS D:\Vartak college\sem 7\NLP\EXP\New folder> python.\exp2.py
Plaintext:  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 solar radii, being larger than almost the entire orbit of Saturn (1,940 - 2,169 solar radii).
Tokenized sentences:  ['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 solar radii, being larger than almost the entire orbit of Saturn (1,940 - 2,169 solar radii).']
Tokenized words:  ['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', 'solar', 'radii', ',', 'being', 'larger', 'than', 'almost', 'the', 'entire', 'orbit', 'of', 'Saturn', '(', '1,940', '-', '2,169', 'solar', 'radii', ')', '.']
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
solar
radii
,
being
larger
than
almost
the
entire
orbit
of
Saturn
(
1,940
-
2,169
solar

dius', 'of', '2', ',', '150', 'solar', 'radii', ',', 'being', 'larger', 'than', 'almost', 'the', 'entire', 'orbit', 'of', 'Saturn', '(', '1', ',', '940', '-', '2', ',', '169', 'solar', 'radii', ').']
Lowercase text: 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 solar radii, being larger than almost the entire orbit of saturn (1,940 - 2,169 solar radii).
Uppercase text: 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 SOLAR RADII, BEING LARGER THAN ALMOST THE ENTIRE ORBIT OF SATURN (1,940 - 2,169 SOLAR RADII).

Conclusion:

Comment on the tools used for tokenization of language input.

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