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



# Vidyavardhini's College of Engineering and Technology Department of Artificial Intelligence & Data Science

**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 it's called as 'Word Tokenization' and if it's split into sentences then it's 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.



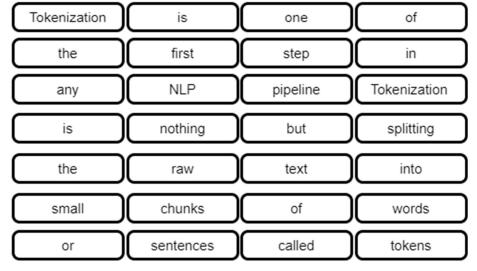
## Vidyavardhini's College of Engineering and Technology

### Department of Artificial Intelligence & Data Science

### 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.





### 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

### **Implementation:**

!pip install nltk import nltk nltk.download()

from nltk.tokenize import sent\_tokenize text = "'I have 2 cats. They eat 3 times a day."' text

'I have 2 cats. They eat 3 times a day.'



# Vidyavardhini's College of Engineering and Technology Department of Artificial Intelligence & Data Science

```
sentences = sent tokenize (text)
sentences
['I have 2 cats.', 'They eat 3 times a day.']
from nltk.tokenize import word tokenize
words = word tokenize (text)
words
 ['I', 'have', '2', 'cats', '.', 'They', 'eat', '3', 'times', 'a', 'day', '.']
for w in words:
  print (w)
T
have
2
cats
They
eat
3
times
day
sent tokenize (text)
 ['I have 2 cats.', 'They eat 3 times a day.']
[word tokenize (text) for t in sent tokenize(text)]
[['I', 'have', '2', 'cats', '.', 'They', 'eat', '3', 'times', 'a', 'day', '.'], ['I', 'have', '2', 'cats', '.', 'They', 'eat', '3', 'times', 'a', 'day', '.']]
from nltk.tokenize import wordpunct tokenize
wordpunct tokenize (text)
['I', 'have', '2', 'cats', '.', 'They', 'eat', '3', 'times', 'a', 'day', '.']
text.lower()
 'i have 2 cats. they eat 3 times a day.'
text.upper()
 'I HAVE 2 CATS. THEY EAT 3 TIMES A DAY.'
```



# Vidyavardhini's College of Engineering and Technology

### Department of Artificial Intelligence & Data Science

#### **Conclusion:**

Comment on the tools used for tokenization of language input.

Tokenization is a key process in natural language processing (NLP), where text is divided into smaller units, such as words or sentences. In your code, several tools from the \*\*NLTK\*\* (Natural Language Toolkit) are used for tokenization:

### 1. sent tokenize

This function splits a text into sentences. It uses punctuation marks like periods, question marks, and exclamations to determine sentence boundaries.

Example:

Input: "I have 2 cats. They eat 3 times a day."

Output: ['I have 2 cats.', 'They eat 3 times a day.']

#### 2. word tokenize

This function breaks a sentence or text into individual words or tokens, including punctuation marks. It's useful for further processing like stemming, lemmatization, or frequency analysis.

Example:

Input: "I have 2 cats."

Output: ['I', 'have', '2', 'cats', '.']

### 3. wordpunct\_tokenize

This tokenization function splits text based on punctuation and whitespace. It treats punctuation separately from words, providing a more granular tokenization.

Example:

Input: "I have 2 cats."

Output: ['I', 'have', '2', 'cats', '.']

These tools are essential for transforming unstructured text into structured formats that are easier to analyze in NLP applications.