Experiment No.2
Apply Tokenization on given English and Indian Language Text
Date of Performance:
Date of Submission:

Aim: Apply Tokenization on given English and Indian Language Text

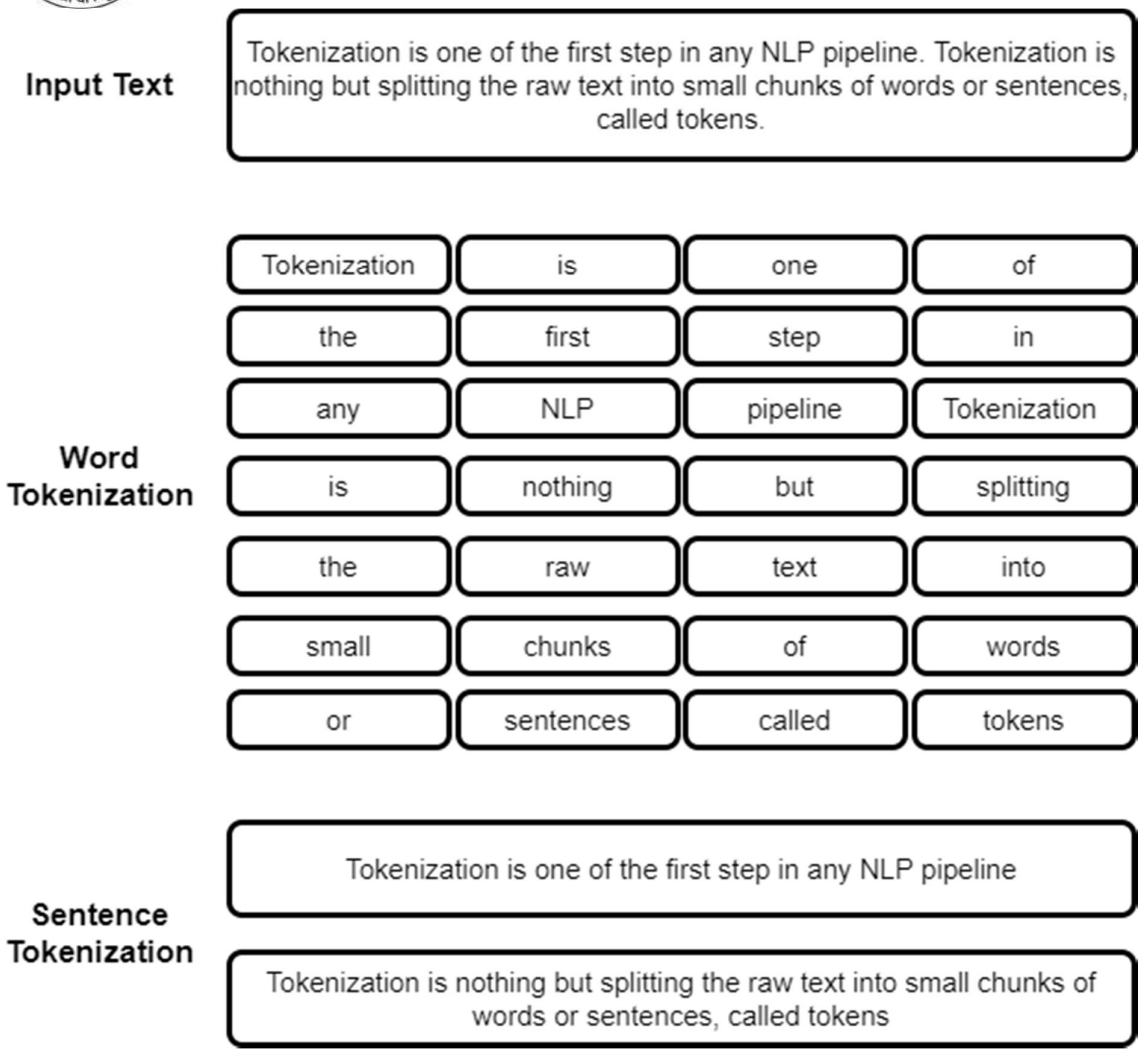
Objective: Able to perform sentence and word tokenization for the given input text for English and Indian Language.

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:

```
!pip install nltk
import nltk
nltk.download()
```

```
NLTK Downloader
---------------------------------------------------------------------------
    d) Download   l) List    u) Update   c) Config   h) Help   q) Quit
---------------------------------------------------------------------------
Downloader> d

Download which package (l=list; x=cancel)?
  Identifier> punkt
    Downloading package punkt to /root/nltk_data...
      Unzipping tokenizers/punkt.zip.
```

```
---------------------------------------------------------------------------
    d) Download   l) List   u) Update   c) Config   h) Help   q) Quit
---------------------------------------------------------------------------
Downloader> q
True
```

 # Sentence Tokenization

```
from nltk.tokenize import sent_tokenize
text = '''In probability, two events are independent if the incidence
of one event does not affect the probability of the other event. If the
incidence of one event does affect the probability of the other event,
then the events are dependent.'''
text
sentences = sent_tokenize (text)
sentences
```

```
['In probability, two events are independent if the incidence of one
event does not affect the probability of the other event.', 'If the
incidence of one event does affect the probability of the other event,
then the events are dependent.']
```

```
#Word Tokenization
```

```
from nltk.tokenize import word_tokenize
words = word_tokenize (text)
words
```

```
['In', 'probability', ',', 'two', 'events', 'are', 'independent', 'if',
'the', 'incidence', 'of', 'one', 'event', 'does', 'not', 'affect', 'the',
'probability', 'of', 'the', 'other', 'event', '.', 'If', 'the',
'incidence', 'of', 'one', 'event', 'does', 'affect', 'the',
'probability', 'of', 'the', 'other', 'event', ',', 'then', 'the',
'events', 'are', 'dependent', '.']
```

```
for w in words:
    print (w)
```

```
In
probability
,
two
events
are
independent
if
the
incidence
of
one
```

```
event
does
not
affect
the
probability
of
the
other
event
.
If
the
incidence
of
one
event
does
affect
the
probability
of
the
other
event
,
then
the
events
are
dependent
```

```
#Levels of Sentences Tokenization using Comprehension
```

```
sent_tokenize (text)
```

```
['In probability, two events are independent if the incidence of one
event does not affect the probability of the other event.', 'If the
incidence of one event does affect the probability of the other event,
then the events are dependent.']
```

```
[word_tokenize (text) for t in sent_tokenize(text)]
```

```
[['In', 'probability', ',', 'two', 'events', 'are', 'independent',
'if', 'the', 'incidence', 'of', 'one', 'event', 'does', 'not',
'affect', 'the', 'probability', 'of', 'the', 'other', 'event', '.',
'If', 'the', 'incidence', 'of', 'one', 'event', 'does', 'affect',
'the', 'probability', 'of', 'the', 'other', 'event', ',', 'then',
'the', 'events', 'are', 'dependent', '.'], ['In', 'probability', ',',
'two', 'events', 'are', 'independent', 'if', 'the', 'incidence', 'of',
'one', 'event', 'does', 'not', 'affect', 'the', 'probability', 'of',
'the', 'other', 'event', '.', 'If', 'the', 'incidence', 'of', 'one',
'event', 'does', 'affect', 'the', 'probability', 'of', 'the', 'other',
'event', ',', 'then', 'the', 'events', 'are', 'dependent', '.']]
```

```
from nltk.tokenize import wordpunct_tokenize
wordpunct_tokenize (text)
```

```
['In', 'probability', ',', 'two', 'events', 'are', 'independent', 'if',
'the', 'incidence', 'of', 'one', 'event', 'does', 'not', 'affect',
'the', 'probability', 'of', 'the', 'other', 'event', '.', 'If', 'the',
'incidence', 'of', 'one', 'event', 'does', 'affect', 'the',
'probability', 'of', 'the', 'other', 'event', ',', 'then', 'the',
'events', 'are', 'dependent', '.']
```

```
text.lower()
```

'in probability, two events are independent if the incidence of one event does not affect the probability of the other event. if the incidence of one event does affect the probability of the other event, then the events are dependent.'

```
text.upper()
```

```
'IN PROBABILITY, TWO EVENTS ARE INDEPENDENT IF THE INCIDENCE OF ONE
EVENT DOES NOT AFFECT THE PROBABILITY OF THE OTHER EVENT. IF THE
INCIDENCE OF ONE EVENT DOES AFFECT THE PROBABILITY OF THE OTHER EVENT,
THEN THE EVENTS ARE DEPENDENT.'
```

Conclusion:

Tokenization is a fundamental natural language processing (NLP) task that involves breaking a text into smaller units called tokens. These tokens can be words, sub words, or characters, depending on the level of granularity chosen for analysis. To perform tokenization on both English and an Indian language text, we need to consider the specific characteristics of each language.