LAB 3 : Perform the following text operation with the help of suitable paragraph of your own choice

a. Lowercasing
b. Tokenization
c. Stemming
d. Punctuation removal
e. Stop words removal
f. Lammatization

1. **Lowercasing**

   - **Definition:** Lowercasing converts all letters in the text to lowercase. This is often done to ensure consistency in text analysis and processing.
   - **Example:** Original text: "The Quick Brown Fox Jumps Over The Lazy Dog" Lowercased text: "the quick brown fox jumps over the lazy dog"
2. **Tokenization**

   - **Definition:** Tokenization involves splitting a text into individual words or tokens.
   - **Example:** Original text: "The Quick Brown Fox Jumps Over The Lazy Dog" Tokens: ["The", "Quick", "Brown", "Fox", "Jumps", "Over", "The", "Lazy", "Dog"]
3. **Stemming**

   - **Definition:** Stemming reduces words to their root or base form. It removes suffixes from words.
   - **Example:** For the root word "like": - "likes" - "liked" - "likely" - "liking"
4. **Punctuation Removal**

   - **Definition:** Punctuation removal involves removing all punctuation marks from the text.
   - **Example:** Original text: "The Quick, Brown Fox Jumps Over The Lazy Dog!" Text after punctuation removal: "The Quick Brown Fox Jumps Over The Lazy Dog"
5. **Stop Words Removal**

   - **Definition:** Stop words removal involves removing common words (e.g., 'is', 'and', 'the') that do not carry much meaning.
   - **Example:** Original text: "The Quick Brown Fox Jumps Over The Lazy Dog" Text after stop words removal: "Quick Brown Fox Jumps Lazy Dog"
6. **Lemmatization**

   - **Definition:** Lemmatization involves reducing words to their base or dictionary form (lemma).
   - **Example:** Original text: "The Quick Brown Fox Jumps Over The Lazy Dog" Lemmatized text: "The Quick Brown Fox Jumps Over The Lazy Dog"

Prime Minister Pushpa Kamal Dahal, who is also chancellor of the Tribhuvan University, on Thursday appointed Professor Keshar Jung Baral as the vice-chancellor of the university. The former Pokhara University vice-chancellor will serve in the position for four years. Baral holds a PhD in Capital Structure and Cost of Capital from Delhi University. He is a professor of finance at the Tribhuvan University. He is the 20th person to assume the office since the university's establishment in 1959. The varsity's search committee had called for applications from aspirants to lead the country's largest university last month. From a total of 43 applicants, the committee had short-listed 14 and made their

names public earlier this month. Professor Baral was one of the three candidates—including Chitra Bahadur Budhathoki and Tankanath Dhamala— recommended by the search committee headed by

In [1]:
```
import nltk
```

In [2]:
```
paragraph = input("Enter the paragraph of text: ")
```

```
Enter the paragraph of text: Prime Minister Pushpa Kamal Dahal, who is also
chancellor of the Tribhuvan University, on Thursday appointed Professor Kes
har Jung Baral as the vice-chancellor of the university. The former Pokhara
University vice-chancellor will serve in the position for four years. Baral
holds a PhD in Capital Structure and Cost of Capital from Delhi University.
He is a professor of finance at the Tribhuvan University. He is the 20th pe
rson to assume the office since the university's establishment in 1959. The
varsity's search committee had called for applications from aspirants to le
ad the country's largest university last month. From a total of 43 applican
ts, the committee had short-listed 14 and made their names public earlier t
his month. Professor Baral was one of the three candidates—including Chitra
Bahadur Budhathoki and Tankanath Dhamala— recommended by the search committ
ee headed by Minister for Education Ashok Kumar Rai.
```

In [3]:
```
from nltk.corpus import stopwords
nltk.download('stopwords', quiet=True)
from nltk.stem import WordNetLemmatizer
```

1. Lowercasing

Lowercasing converts all letters in the text to lowercase. This is often done to ensure consistency in text analysis and processing.

In [4]:
```
lowercase_paragraph = paragraph.lower()

print("Lowercased paragraph:")
print(lowercase_paragraph)
```

```
Lowercased paragraph:
prime minister pushpa kamal dahal, who is also chancellor of the tribhuvan
university, on thursday appointed professor keshar jung baral as the vice-c
hancellor of the university. the former pokhara university vice-chancellor
will serve in the position for four years. baral holds a phd in capital str
ucture and cost of capital from delhi university. he is a professor of fina
nce at the tribhuvan university. he is the 20th person to assume the office
since the university's establishment in 1959. the varsity's search committe
e had called for applications from aspirants to lead the country's largest
university last month. from a total of 43 applicants, the committee had sho
rt-listed 14 and made their names public earlier this month. professor bara
l was one of the three candidates—including chitra bahadur budhathoki and t
ankanath dhamala— recommended by the search committee headed by minister fo
r education ashok kumar rai.
```

2. Tokenization

Tokenization involves splitting a text into individual words or tokens.

In [5]:
```
import nltk

# Tokenization
tokens = nltk.word_tokenize(paragraph)

print("\nTokens:")
print(tokens)
```

```
Tokens:
['Prime', 'Minister', 'Pushpa', 'Kamal', 'Dahal', ',', 'who', 'is', 'also',
'chancellor', 'of', 'the', 'Tribhuvan', 'University', ',', 'on', 'Thursda
y', 'appointed', 'Professor', 'Keshar', 'Jung', 'Baral', 'as', 'the', 'vice
-chancellor', 'of', 'the', 'university', '.', 'The', 'former', 'Pokhara',
'University', 'vice-chancellor', 'will', 'serve', 'in', 'the', 'position',
'for', 'four', 'years', '.', 'Baral', 'holds', 'a', 'PhD', 'in', 'Capital',
'Structure', 'and', 'Cost', 'of', 'Capital', 'from', 'Delhi', 'University',
'.', 'He', 'is', 'a', 'professor', 'of', 'finance', 'at', 'the', 'Tribhuva
n', 'University', '.', 'He', 'is', 'the', '20th', 'person', 'to', 'assume',
'the', 'office', 'since', 'the', 'university', '’', 's', 'establishment',
'in', '1959', '.', 'The', 'varsity', '’', 's', 'search', 'committee', 'ha
d', 'called', 'for', 'applications', 'from', 'aspirants', 'to', 'lead', 'th
e', 'country', '’', 's', 'largest', 'university', 'last', 'month', '.', 'Fr
om', 'a', 'total', 'of', '43', 'applicants', ',', 'the', 'committee', 'ha
d', 'short-listed', '14', 'and', 'made', 'their', 'names', 'public', 'earli
er', 'this', 'month', '.', 'Professor', 'Baral', 'was', 'one', 'of', 'the',
'three', 'candidates—including', 'Chitra', 'Bahadur', 'Budhathoki', 'and',
'Tankanath', 'Dhamala—', 'recommended', 'by', 'the', 'search', 'committee',
'headed', 'by', 'Minister', 'for', 'Education', 'Ashok', 'Kumar', 'Rai',
'.']
```

3. Stemming

Stemming reduces words to their root or base form. It removes suffixes from words.

In [6]:
```
from nltk.stem import PorterStemmer

# Initialize stemmer
stemmer = PorterStemmer()

# Stemming
stemmed_words = [stemmer.stem(token) for token in tokens]

print("\nStemmed words:")
print(stemmed_words)
```

```
Stemmed words:
['prime', 'minist', 'pushpa', 'kamal', 'dahal', ',', 'who', 'is', 'also',
'chancellor', 'of', 'the', 'tribhuvan', 'univers', ',', 'on', 'thursday',
'appoint', 'professor', 'keshar', 'jung', 'baral', 'as', 'the', 'vice-chanc
ellor', 'of', 'the', 'univers', '.', 'the', 'former', 'pokhara', 'univers',
'vice-chancellor', 'will', 'serv', 'in', 'the', 'posit', 'for', 'four', 'ye
ar', '.', 'baral', 'hold', 'a', 'phd', 'in', 'capit', 'structur', 'and', 'c
ost', 'of', 'capit', 'from', 'delhi', 'univers', '.', 'he', 'is', 'a', 'pro
fessor', 'of', 'financ', 'at', 'the', 'tribhuvan', 'univers', '.', 'he', 'i
s', 'the', '20th', 'person', 'to', 'assum', 'the', 'offic', 'sinc', 'the',
'univers', '’', 's', 'establish', 'in', '1959', '.', 'the', 'varsiti', '’',
's', 'search', 'committe', 'had', 'call', 'for', 'applic', 'from', 'aspir',
'to', 'lead', 'the', 'countri', '’', 's', 'largest', 'univers', 'last', 'mo
nth', '.', 'from', 'a', 'total', 'of', '43', 'applic', ',', 'the', 'committ
e', 'had', 'short-list', '14', 'and', 'made', 'their', 'name', 'public', 'e
arlier', 'thi', 'month', '.', 'professor', 'baral', 'wa', 'one', 'of', 'th
e', 'three', 'candidates-includ', 'chitra', 'bahadur', 'budhathoki', 'and',
'tankanath', 'dhamala-', 'recommend', 'by', 'the', 'search', 'committe', 'h
ead', 'by', 'minist', 'for', 'educ', 'ashok', 'kumar', 'rai', '.']
```

4. Punctuation Removal

Punctuation removal involves removing all punctuation marks from the text.

In [7]:
```
import string

# Punctuation removal
cleaned_tokens = [token for token in tokens if token not in string.punctuati

print("\nTokens after punctuation removal:")
print(cleaned_tokens)
```

```
Tokens after punctuation removal:
['Prime', 'Minister', 'Pushpa', 'Kamal', 'Dahal', 'who', 'is', 'also', 'cha
ncellor', 'of', 'the', 'Tribhuvan', 'University', 'on', 'Thursday', 'appoin
ted', 'Professor', 'Keshar', 'Jung', 'Baral', 'as', 'the', 'vice-chancello
r', 'of', 'the', 'university', 'The', 'former', 'Pokhara', 'University', 'v
ice-chancellor', 'will', 'serve', 'in', 'the', 'position', 'for', 'four',
'years', 'Baral', 'holds', 'a', 'PhD', 'in', 'Capital', 'Structure', 'and',
'Cost', 'of', 'Capital', 'from', 'Delhi', 'University', 'He', 'is', 'a', 'p
rofessor', 'of', 'finance', 'at', 'the', 'Tribhuvan', 'University', 'He',
'is', 'the', '20th', 'person', 'to', 'assume', 'the', 'office', 'since', 't
he', 'university', '’', 's', 'establishment', 'in', '1959', 'The', 'varsit
y', '’', 's', 'search', 'committee', 'had', 'called', 'for', 'application
s', 'from', 'aspirants', 'to', 'lead', 'the', 'country', '’', 's', 'larges
t', 'university', 'last', 'month', 'From', 'a', 'total', 'of', '43', 'appli
cants', 'the', 'committee', 'had', 'short-listed', '14', 'and', 'made', 'th
eir', 'names', 'public', 'earlier', 'this', 'month', 'Professor', 'Baral',
'was', 'one', 'of', 'the', 'three', 'candidates—including', 'Chitra', 'Baha
dur', 'Budhathoki', 'and', 'Tankanath', 'Dhamala—', 'recommended', 'by', 't
he', 'search', 'committee', 'headed', 'by', 'Minister', 'for', 'Education',
'Ashok', 'Kumar', 'Rai']
```

5. Stop Words Removal

Stop words removal involves removing common words (e.g., 'is', 'and', 'the') that do not carry much meaning.

In [8]:
```
# Get stopwords list
stop_words = set(stopwords.words('english'))

# Stop words removal
filtered_tokens = [token for token in cleaned_tokens if token.lower() not in

print("\nTokens after stop words removal:")
print(filtered_tokens)
```

```
Tokens after stop words removal:
['Prime', 'Minister', 'Pushpa', 'Kamal', 'Dahal', 'also', 'chancellor', 'Tr
ibhuvan', 'University', 'Thursday', 'appointed', 'Professor', 'Keshar', 'Ju
ng', 'Baral', 'vice-chancellor', 'university', 'former', 'Pokhara', 'Univer
sity', 'vice-chancellor', 'serve', 'position', 'four', 'years', 'Baral', 'h
olds', 'PhD', 'Capital', 'Structure', 'Cost', 'Capital', 'Delhi', 'Universi
ty', 'professor', 'finance', 'Tribhuvan', 'University', '20th', 'person',
'assume', 'office', 'since', 'university', '’', 'establishment', '1959', 'v
arsity', '’', 'search', 'committee', 'called', 'applications', 'aspirants',
'lead', 'country', '’', 'largest', 'university', 'last', 'month', 'total',
'43', 'applicants', 'committee', 'short-listed', '14', 'made', 'names', 'pu
blic', 'earlier', 'month', 'Professor', 'Baral', 'one', 'three', 'candidate
s—including', 'Chitra', 'Bahadur', 'Budhathoki', 'Tankanath', 'Dhamala–',
'recommended', 'search', 'committee', 'headed', 'Minister', 'Education', 'A
shok', 'Kumar', 'Rai']
```

6. Lemmatization

Lemmatization involves reducing words to their base or dictionary form (lemma).

In [9]:
```
# Lemmatization
lemmatizer = WordNetLemmatizer()
lemmatized_words = [lemmatizer.lemmatize(token) for token in filtered_tokens

# Show lemmatized words
print("\nLemmatized words:")
print(lemmatized_words)
```

```
Lemmatized words:
['Prime', 'Minister', 'Pushpa', 'Kamal', 'Dahal', 'also', 'chancellor', 'Tr
ibhuvan', 'University', 'Thursday', 'appointed', 'Professor', 'Keshar', 'Ju
ng', 'Baral', 'vice-chancellor', 'university', 'former', 'Pokhara', 'Univer
sity', 'vice-chancellor', 'serve', 'position', 'four', 'year', 'Baral', 'ho
ld', 'PhD', 'Capital', 'Structure', 'Cost', 'Capital', 'Delhi', 'Universit
y', 'professor', 'finance', 'Tribhuvan', 'University', '20th', 'person', 'a
ssume', 'office', 'since', 'university', '’', 'establishment', '1959', 'var
sity', '’', 'search', 'committee', 'called', 'application', 'aspirant', 'le
ad', 'country', '’', 'largest', 'university', 'last', 'month', 'total', '4
3', 'applicant', 'committee', 'short-listed', '14', 'made', 'name', 'publi
c', 'earlier', 'month', 'Professor', 'Baral', 'one', 'three', 'candidates—i
ncluding', 'Chitra', 'Bahadur', 'Budhathoki', 'Tankanath', 'Dhamala–', 'rec
ommended', 'search', 'committee', 'headed', 'Minister', 'Education', 'Asho
k', 'Kumar', 'Rai']
```