In [2]:
```
# Step 0: Install and Import Required Libraries
import nltk
import string
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk import pos_tag
from nltk.stem import PorterStemmer, WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
import numpy as np
```

In [4]:
```
# Step 1: Download NLTK resources (only first time)
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('averaged_perceptron_tagger')
nltk.download('wordnet')
nltk.download('omw-1.4')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\piyus\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\piyus\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\piyus\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\piyus\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package omw-1.4 to
[nltk_data]     C:\Users\piyus\AppData\Roaming\nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
```

Out[4]: True

In [8]:
```
# Step 2: Sample Documents
documents = [
    "Natural Language Processing is a field of Artificial Intelligence.",
    "It deals with the interaction between computers and humans using natural la
    "NLP techniques are widely used in text analytics and sentiment analysis."
]
```

In [11]:
```
# Step 3: Text Preprocessing Function
def preprocess_text(text):
    # 3.1 Convert to lowercase
    text = text.lower()

    # 3.2 Tokenization
    tokens = word_tokenize(text)

    # 3.3 Remove punctuation
    tokens = [word for word in tokens if word not in string.punctuation]

    # 3.4 Remove Stop Words
    stop_words = set(stopwords.words('english'))
    filtered_tokens = [word for word in tokens if word not in stop_words]
```

```
    # 3.5 POS Tagging
    pos_tags = pos_tag(filtered_tokens)

    # 3.6 Stemming
    stemmer = PorterStemmer()
    stemmed_tokens = [stemmer.stem(word) for word in filtered_tokens]

    # 3.7 Lemmatization
    lemmatizer = WordNetLemmatizer()
    lemmatized_tokens = [lemmatizer.lemmatize(word) for word in filtered_tokens]

    return {
        'tokens': tokens,
        'filtered': filtered_tokens,
        'pos_tags': pos_tags,
        'stemmed': stemmed_tokens,
        'lemmatized': lemmatized_tokens
    }
```

In [20]:

```
# Step 4: Preprocess and Print Each Document
for i, doc in enumerate(documents):
    print(f"\n--- Document {i+1} ---")
    result = preprocess_text(doc)
    print("Original Tokens:     ", result['tokens'])
    print("After Stop Removal:  ", result['filtered'])
    print("POS Tags:            ", result['pos_tags'])
    print("After Stemming:      ", result['stemmed'])
    print("After Lemmatization: ", result['lemmatized'])
```

```
--- Document 1 ---
Original Tokens:      ['natural', 'language', 'processing', 'is', 'a', 'field',
'of', 'artificial', 'intelligence']
After Stop Removal:   ['natural', 'language', 'processing', 'field', 'artificia
l', 'intelligence']
POS Tags:             [('natural', 'JJ'), ('language', 'NN'), ('processing', 'N
N'), ('field', 'NN'), ('artificial', 'JJ'), ('intelligence', 'NN')]
After Stemming:       ['natur', 'languag', 'process', 'field', 'artifici', 'intel
lig']
After Lemmatization:  ['natural', 'language', 'processing', 'field', 'artificia
l', 'intelligence']

--- Document 2 ---
Original Tokens:      ['it', 'deals', 'with', 'the', 'interaction', 'between', 'c
omputers', 'and', 'humans', 'using', 'natural', 'language']
After Stop Removal:   ['deals', 'interaction', 'computers', 'humans', 'using', 'n
atural', 'language']
POS Tags:             [('deals', 'NNS'), ('interaction', 'VBP'), ('computers', 'N
NS'), ('humans', 'NNS'), ('using', 'VBG'), ('natural', 'JJ'), ('language', 'NN')]
After Stemming:       ['deal', 'interact', 'comput', 'human', 'use', 'natur', 'la
nguag']
After Lemmatization:  ['deal', 'interaction', 'computer', 'human', 'using', 'natu
ral', 'language']

--- Document 3 ---
Original Tokens:      ['nlp', 'techniques', 'are', 'widely', 'used', 'in', 'tex
t', 'analytics', 'and', 'sentiment', 'analysis']
After Stop Removal:   ['nlp', 'techniques', 'widely', 'used', 'text', 'analytic
s', 'sentiment', 'analysis']
POS Tags:             [('nlp', 'RB'), ('techniques', 'NNS'), ('widely', 'RB'),
('used', 'VBD'), ('text', 'JJ'), ('analytics', 'NNS'), ('sentiment', 'NN'), ('ana
lysis', 'NN')]
After Stemming:       ['nlp', 'techniqu', 'wide', 'use', 'text', 'analyt', 'senti
ment', 'analysi']
After Lemmatization:  ['nlp', 'technique', 'widely', 'used', 'text', 'analytics',
'sentiment', 'analysis']
```

In [21]:
```python
# Step 5: Term Frequency (TF) Matrix using CountVectorizer
print("\n--- Term Frequency (TF) Matrix ---")
tf_vectorizer = CountVectorizer(stop_words='english')
tf_matrix = tf_vectorizer.fit_transform(documents)
print("Vocabulary:", tf_vectorizer.get_feature_names_out())
print(tf_matrix.toarray())
```

```
--- Term Frequency (TF) Matrix ---
Vocabulary: ['analysis' 'analytics' 'artificial' 'computers' 'deals' 'field' 'hum
ans'
 'intelligence' 'interaction' 'language' 'natural' 'nlp' 'processing'
 'sentiment' 'techniques' 'text' 'used' 'using' 'widely']
[[0 0 1 0 0 1 0 1 0 1 1 0 1 0 0 0 0 0 0]
 [0 0 0 1 1 0 1 0 1 1 1 0 0 0 0 0 0 1 0]
 [1 1 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 0 1]]
```

In [22]:
```python
# Step 6: TF-IDF Matrix using TfidfVectorizer
print("\n--- TF-IDF Matrix ---")
tfidf_vectorizer = TfidfVectorizer(stop_words='english')
tfidf_matrix = tfidf_vectorizer.fit_transform(documents)
print("Vocabulary:", tfidf_vectorizer.get_feature_names_out())
print(np.round(tfidf_matrix.toarray(), 3))  # rounded for better readability
```

```
--- TF-IDF Matrix ---
Vocabulary: ['analysis' 'analytics' 'artificial' 'computers' 'deals' 'field' 'hum
ans'
 'intelligence' 'interaction' 'language' 'natural' 'nlp' 'processing'
 'sentiment' 'techniques' 'text' 'used' 'using' 'widely']
[[0.    0.    0.44  0.    0.    0.44  0.    0.44  0.    0.335 0.335 0.
  0.44  0.    0.    0.    0.    0.    0.    0.   ]
 [0.    0.    0.    0.403 0.403 0.    0.403 0.    0.403 0.307 0.307 0.
  0.    0.    0.    0.    0.    0.403 0.    0.   ]
 [0.354 0.354 0.    0.    0.    0.    0.    0.    0.    0.    0.    0.354
  0.    0.354 0.354 0.354 0.354 0.    0.354]]
```

In [ ]: