In [1]:
```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer, WordNetLemmatizer
from nltk import pos_tag
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [2]:
```
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
nltk.download('stopwords')
nltk.download('wordnet')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Atul\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\Atul\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping taggers\averaged_perceptron_tagger.zip.
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Atul\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\Atul\AppData\Roaming\nltk_data...
```
Out[2]:
```
True
```

In [3]:
```
# Load the sample document
document = "This is a sample document. It includes various words and sentences."
```

In [4]:
```
# Tokenization
tokens = word_tokenize(document)
```

In [5]:
```
print(tokens)
```

```
['This', 'is', 'a', 'sample', 'document', '.', 'It', 'includes', 'various', 'words',
'and', 'sentences', '.']
```

In [6]:
```
# POS Tagging
pos_tags = pos_tag(tokens)
print(pos_tags)
# Stop Words Removal
stop_words = set(stopwords.words('english'))
print(stop_words)
filtered_tokens = [token for token in tokens if token.lower() not in stop_words]
```

```
[('This', 'DT'), ('is', 'VBZ'), ('a', 'DT'), ('sample', 'JJ'), ('document', 'NN'),
('.', '.'), ('It', 'PRP'), ('includes', 'VBZ'), ('various', 'JJ'), ('words', 'NNS'),
('and', 'CC'), ('sentences', 'NNS'), ('.', '.')]
{'that', 'only', 'same', 'd', 'yourselves', 'your', 'ourselves', 'if', 'very', 'hims
elf', 'i', 'which', 'too', 'has', 'when', 'her', 'having', 'now', "you'd", "does
n't", "weren't", 'herself', 'them', 'where', 'each', 'by', 'under', 'doing', 'what',
'nor', 'myself', 'how', 'had', 'be', 'yourself', 'any', 'we', "should've", 'you', 't
hese', 'both', 'before', 'because', 'hadn', 'don', 'are', 'wasn', 'then', 'their',
't', 'about', 'ours', 'our', "wouldn't", "shan't", 'those', 'up', 'all', 'will', 'ag
ain', "you're", 'is', 'other', "didn't", 'mustn', 'with', 'she', "wasn't", 'into',
'to', 'just', 'it', 'once', 'down', 'few', "haven't", 've', 'y', 'at', 'have', 'your
s', 'weren', 'no', 'further', 'there', 'me', "hadn't", "isn't", "mightn't", 'his',
```

```
'until', 'for', 'o', 'themselves', 'been', 'off', 'more', 'do', "mustn't", 'betwee
n', 'not', "aren't", 'own', 'such', 'out', 'haven', 'him', 'being', 'ma', 'from', 'n
eedn', "that'll", 'mightn', "it's", 'doesn', 'should', "you've", 'this', 'aren', 'o
f', 'through', 'above', 'or', "hasn't", 'while', 'who', 'its', 'couldn', 'shouldn',
'and', 'here', "couldn't", 'than', "won't", 'can', 'll', "needn't", 'didn', 'does',
'but', 'whom', 'the', 'below', 'did', 'over', 'so', 'hers', "she's", 'a', 'an', "sho
uldn't", 'they', 'shan', 'in', 'why', 'was', "don't", 'wouldn', 'theirs', 'isn', 'o
n', 'he', 'during', 'against', 's', 'hasn', 'won', "you'll", 'some', 'after', 'm',
're', 'as', 'were', 'most', 'itself', 'my', 'ain', 'am'}
```

In [7]:

```
# Stemming
words = ["game","gaming","gamed","games"]
ps = PorterStemmer()
for word in words:
    print(ps.stem(word))
# Lemmatization
lemmatizer = WordNetLemmatizer()
lemmatized_tokens = [lemmatizer.lemmatize(token) for token in filtered_tokens]
print(lemmatized_tokens)
# Create TF-IDF representation
documents = [document]
vectorizer = TfidfVectorizer()
tfidf_matrix = vectorizer.fit_transform(documents)
```

```
game
game
game
game
['sample', 'document', '.', 'includes', 'various', 'word', 'sentence', '.']
```

In [ ]: