Basic Morphological Analysis

In [1]:
```
import nltk
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
from sklearn.feature_extraction.text import CountVectorizer
import re
```

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

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\ASUS\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Out[2]: True

In [3]:
```
text = "The cats are running and jumping over the fence."
text = text.lower()
text = re.sub(r'\W', ' ', text)
text = re.sub(r'\s+', ' ', text)
```

In [4]:
```
words = word_tokenize(text)

print("Tokenized words: ", words)
```

```
Tokenized words:  ['the', 'cats', 'are', 'running', 'and', 'jumping', 'over', 'the', 'fence']
```

In [5]:
```
stemmer = PorterStemmer()
```

In [6]:
```
stemmed_words = [stemmer.stem(word) for word in words]
print("Stemmed Words: ", stemmed_words)
```

```
Stemmed Words:  ['the', 'cat', 'are', 'run', 'and', 'jump', 'over', 'the', 'fenc']
```

In [7]:
```
# BoW method
vectorizer = CountVectorizer()
X = vectorizer.fit_transform([text])
print("Vocabulary:", vectorizer.get_feature_names_out())
print("BoW Matrix:", X.toarray())
```

```
Vocabulary: ['and' 'are' 'cats' 'fence' 'jumping' 'over' 'running' 'the']
BoW Matrix: [[1 1 1 1 1 1 1 2]]
```

In [8]:
```
# Bag of Words method (optional)
word2count = {}
for word in words:
        if word not in word2count.keys():
                word2count[word] = 1
        else:
                word2count[word] += 1

print(word2count)
```

```
{'the': 2, 'cats': 1, 'are': 1, 'running': 1, 'and': 1, 'jumping': 1, 'over': 1, 'fence': 1}
```

In [ ]:
```
```