```
!pip install nltk
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages (3.8.1)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk) (8.1.7)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from nltk) (1.3.2)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk) (2023.12.25)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk) (4.66.2)
```

```
import nltk
```

```
nltk.download("punkt")
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

```
nltk.download("stopwords")
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
from nltk import sent_tokenize, word_tokenize
from nltk.corpus import stopwords
```

```
stopwords = stopwords.words("english")
```

```
sent = "I will walk 500 miles and I would walk 500 more , just to be the man who walks a thousand miles to fall down at your door"
```

```
words = word_tokenize(sent)
print(words)
```

```
['I', 'will', 'walk', '500', 'miles', 'and', 'I', 'would', 'walk', '500', 'more', ',', 'just', 'to', 'be', 'the', 'man', 'who', 'walks'
```

```
sentences = sent_tokenize(sent)
print(sentences)
```

```
['I will walk 500 miles and I would walk 500 more , just to be the man who walks a thousand miles to fall down at your door']
```

```
print(stopwords)
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself'
```

```
for word in words:
    if word in stopwords:
        words.remove(word)
print(words)
```

```
['I', 'walk', '500', 'miles', 'I', 'would', 'walk', '500', ',', 'the', 'man', 'walks', 'thousand', 'miles', 'to', 'fall', 'at', 'door']
```

```
from nltk import PorterStemmer
stemmer = PorterStemmer()
```

```
print(words)
```

```
['I', 'walk', '500', 'miles', 'I', 'would', 'walk', '500', ',', 'the', 'man', 'walks', 'thousand', 'miles', 'to', 'fall', 'at', 'door']
```

```
stemmed_words = [stemmer.stem(word) for word in words]
print(stemmed_words)
```

```
['i', 'walk', '500', 'mile', 'i', 'would', 'walk', '500', ',', 'the', 'man', 'walk', 'thousand', 'mile', 'to', 'fall', 'at', 'door']
```

```
from nltk.stem import SnowballStemmer
```

```
snowball_stemmer = SnowballStemmer("english")

snowball_stemming_result = [snowball_stemmer.stem(word) for word in words]
print(snowball_stemming_result)
```

```
['i', 'walk', '500', 'mile', 'i', 'would', 'walk', '500', ',', 'the', 'man', 'walk', 'thousand', 'mile', 'to', 'fall', 'at', 'door']
```

```
from collections import Counter
from nltk import pos_tag


nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
True
```

```
count_dict = {}

for word in stemmed_words:
    if word in count_dict:
        count_dict[word] += 1
    else:
        count_dict[word] = 1

print(count_dict)
```

```
{'i': 2, 'walk': 3, '500': 2, 'mile': 2, 'would': 1, ',': 1, 'the': 1, 'man': 1, 'thousand': 1, 'to': 1, 'fall': 1, 'at': 1, 'door': 1}
```

```
pos_tagged = pos_tag(stemmed_words)
print(pos_tagged)
```

```
[('i', 'NN'), ('walk', 'VBP'), ('500', 'CD'), ('mile', 'NN'), ('i', 'NN'), ('would', 'MD'), ('walk', 'VB'), ('500', 'CD'), (',', ','),
```

```
count = Counter(tag for _ , tag in pos_tagged)
print(count)
```

```
Counter({'NN': 7, 'VBP': 2, 'CD': 2, 'VB': 2, 'MD': 1, ',': 1, 'DT': 1, 'TO': 1, 'IN': 1})
```