

```
import nltk
```

4 major use of nltk tokenization stopwords removal stemming lemmatization

```
nltk.download("punkt_tab")
nltk.download("punkt")
nltk.download("stopwords")
nltk.download("wordnet")
```

```
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data] Unzipping tokenizers/punkt_tab.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
True
```

```
from nltk.tokenize import sent_tokenize, word_tokenize #tokenisation
from nltk.corpus import stopwords #remove stopwords
from nltk.stem import PorterStemmer #stemming

from nltk.stem import WordNetLemmatizer #lemmatization
```

```
text= "I am learning python programming, and it is very. helpfull! "
```

```
print("orig text: ", text)
```

```
orig text: I am learning python programming, and it is very. helpfull!
```

```
# lower case
text=text.lower()
print("Aft lowwcase: ",text)
```

```
Aft lowwcase: i am learning python programming, and it is very. helpfull!
```

```
#tokenizer
#text.split()
tokens=word_tokenize(text)
print("After tokenization: ", tokens)
```

```
After tokenization: ['i', 'am', 'learning', 'python', 'programming', ',', 'and', 'it', 'is', 'very', '.', 'helpfull', '!']
```

```
#remove puctuation
import string
punc=string.punctuation
print(punc)
```

```
!"#$%&'()*+,-./:;<=>?@[\\]^_`{|}~
```

```
punctuation_filter=[word for word in tokens if word not in punc]
print("Aft remo :",punctuation_filter)
```

```
Aft remo : ['i', 'am', 'learning', 'python', 'programming', 'and', 'it', 'is', 'very', 'helpfull']
```

```
#remove stopwords
english_stopwords=stopwords.words("english")
filter_tokens=[word for word in punctuation_filter if word not in english_stopwords]
print("Aft stopword: ", filter_tokens)
```

```
Aft stopword: ['learning', 'python', 'programming', 'helpfull']
```

```
print(stopwords.fileids())
```

```
['albanian', 'arabic', 'azerbaijani', 'basque', 'belarusian', 'bengali', 'catalan', 'chinese', 'danish', 'dutch', 'english',
```

Here are some additional sentences for text processing practice.

- Natural Language Processing is a fascinating field.
- It involves computers understanding human language.
- Many applications, like chatbots, rely on NLP.
- Learning NLP techniques opens up new possibilities.

```
#stemming in lemmetizaation
stem=PorterStemmer()
stem.stem("went")
```

```
'went'
```

Start coding or [generate](#) with AI.

```
wnet=WordNetLemmatizer()
wnet.lemmatize("went","v")#here v is verb
# wnet.lemmatize("buying","v")
```

```
'go'
```

```
lemmatized_words=[]
for word in filter_tokens:
    lemmatized_words.append(wnet.lemmatize(word,"v"))
print("Aft lemmatization: ",lemmatized_words)
```

```
Aft lemmatization:  ['learn', 'python', 'program', 'helpfull']
```

```
" ".join(lemmatized_words)
```

```
'learn python program helpfull'
```

```
sentences=["i am good",
           "I am happy",
           "i am sad",
           "i am bored"]
def preprocess_text(sentences):
    cleaned_sentences = []
    #logic for text processing
    for sentence in sentences:
        # Lower case
        sentence = sentence.lower()
        # Tokenization
        tokens = word_tokenize(sentence)
        # Remove punctuation
        punctuation_filter = [word for word in tokens if word not in punc]
        # Remove stopwords
        filter_tokens = [word for word in punctuation_filter if word not in english_stopwords]
        # Lemmatization
        lemmatized_words = [wnet.lemmatize(word, "v") for word in filter_tokens]
        #append
        cleaned_sentences.append(" ".join(lemmatized_words))

    return cleaned_sentences
cleaned_text=preprocess_text(sentences)
print(cleaned_text)
```

```
['good', 'happy', 'sad', 'bore']
```

Start coding or [generate](#) with AI.