```
import nltk
nltk.download("punkt")
nltk.download("stopwords")
nltk.download("wordnet")
nltk.download("averaged_perceptron_tagger")

from nltk import word_tokenize, sent_tokenize

corpus = "Sachin was the GOAT of the previous generation. Virat is the GOAT of this generation.
Shubman will be the GOAT of the next generation"

print(word_tokenize(corpus))
print(sent_tokenize(corpus))

from nltk import pos_tag

tokens = word_tokenize(corpus)
print(pos_tag(tokens))

from nltk.corpus import stopwords
stop_words = set(stopwords.words("english"))

tokens = word_tokenize(corpus)
cleaned_tokens = []
for token in tokens:
  if (token not in stop_words):
    cleaned_tokens.append(token)
print(cleaned_tokens)

from nltk.stem import PorterStemmer
```

```
stemmer = PorterStemmer()

stemmed_tokens = []
for token in cleaned_tokens:
  stemmed = stemmer.stem(token)
  stemmed_tokens.append(stemmed)
print(stemmed_tokens)

from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()

lemmatized_tokens = []
for token in cleaned_tokens:
  lemmatized = lemmatizer.lemmatize(token)
  lemmatized_tokens.append(lemmatized)
print(lemmatized_tokens)

from sklearn.feature_extraction.text import TfidfVectorizer

corpus = [
    "Sachin was the GOAT of the previous generation",
    "Virat is the GOAT of the this generation",
    "Shubman will be the GOAT of the next generation"
]

vectorizer = TfidfVectorizer()

matrix = vectorizer.fit(corpus)
matrix.vocabulary_
```

```
tfidf_matrix = vectorizer.transform(corpus)
print(tfidf_matrix)


print(vectorizer.get_feature_names_out())
```