```
import nltk
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger_eng')
nltk.download('punkt_tab') # Added to download the missing resource
sentence = "Students are learning Natural Language Processing"
tokens = nltk.word_tokenize(sentence)
pos_tags = nltk.pos_tag(tokens)
print(pos_tags)
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger_eng to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger_eng is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
[('Students', 'NNS'), ('are', 'VBP'), ('learning', 'VBG'), ('Natural', 'NNP'), ('Language', 'NNP'), ('Processing', 'NNP')]
```

```
import spacy
nlp = spacy.load("en_core_web_sm")
doc = nlp("Students are learning Natural Language Processing")
for token in doc:
    print(token.text, token.pos_)
```

```
Students NOUN
are AUX
learning VERB
Natural PROPN
Language PROPN
Processing NOUN
```

```
import spacy
nlp = spacy.load("en_core_web_sm")
doc = nlp("Apple is looking at buying a startup in India.")
for token in doc:
  print(token.text, token.pos_, token.tag_)
```

```
Apple PROPN NNP
is AUX VBZ
looking VERB VBG
at ADP IN
buying VERB VBG
a DET DT
startup NOUN NN
in ADP IN
India PROPN NNP
. PUNCT .
```

```
import spacy
from collections import Counter
nlp = spacy.load("en_core_web_sm")
text = "Loving the new AI features ���� #AI #MachineLearning"
doc = nlp(text)
nouns = []
verbs = []
for token in doc:
  if token.pos_ in ["NOUN", "PROPN"]:
    nouns.append(token.text)
  elif token.pos_ == "VERB":
    verbs.append(token.text)
noun_freq = Counter(nouns)
verb_freq = Counter(verbs)
print("Noun Frequency:", noun_freq)
print("Verb Frequency:", verb_freq)
```

```
Noun Frequency: Counter({'�': 4, 'AI': 2, 'MachineLearning': 1})
Verb Frequency: Counter({'Loving': 1})
```

```
SRUniversity="""The SR University campus is located in Ananthasagar village of Hasanparthy Mandal in Warangal, Telangana, India
It is in 150 acres, with both separate hostel facilities for boys and girls.
There is a huge central library along with Indias largest Technology Business Incubator (TBI) in tier 2 cities."""
```

```
nltk.download('averaged_perceptron_tagger')
from nltk.tokenize import word_tokenize
words = word_tokenize(SRUniversity)
nltk.pos_tag(words)
```

```
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
[('The', 'DT'),
 ('SR', 'NNP'),
 ('University', 'NNP'),
 ('campus', 'NN'),
 ('is', 'VBZ'),
 ('located', 'VBN'),
 ('in', 'IN'),
 ('Ananthasagar', 'NNP'),
 ('village', 'NN'),
 ('of', 'IN'),
 ('Hasanparthy', 'NNP'),
 ('Mandal', 'NNP'),
 ('in', 'IN'),
 ('Warangal', 'NNP'),
 (',', ','),
 ('Telangana', 'NNP'),
 (',', ','),
 ('India', 'NNP'),
 ('.', '.'),
 ('It', 'PRP'),
 ('is', 'VBZ'),
 ('in', 'IN'),
 ('150', 'CD'),
 ('acres', 'NNS'),
 (',', ','),
 ('with', 'IN'),
 ('both', 'DT'),
 ('separate', 'JJ'),
 ('hostel', 'NN'),
 ('facilities', 'NNS'),
 ('for', 'IN'),
 ('boys', 'NNS'),
 ('and', 'CC'),
 ('girls', 'NNS'),
 ('.', '.'),
 ('There', 'EX'),
 ('is', 'VBZ'),
 ('a', 'DT'),
 ('huge', 'JJ'),
 ('central', 'JJ'),
 ('library', 'NN'),
 ('along', 'IN'),
 ('with', 'IN'),
 ('Indias', 'NNP'),
 ('largest', 'JJS'),
 ('Technology', 'NN'),
 ('Business', 'NNP'),
 ('Incubator', 'NNP'),
 ('(', '('),
 ('TBI', 'NNP'),
 (')', ')'),
 ('in', 'IN'),
 ('tier', '$'),
 ('2', 'CD'),
 ('cities', 'NNS'),
 ('.', '.')]
```

```
nltk.download('tagsets')
nltk.download('tagsets_json') # Download the missing resource
nltk.help.upenn_tagset()
```

```
[nltk_data] Downloading package tagsets to /root/nltk_data...
[nltk_data]   Package tagsets is already up-to-date!
[nltk_data] Downloading package tagsets_json to /root/nltk_data...
$: dollar
    $ -$ --$ A$ C$ HK$ M$ NZ$ S$ U.S.$ US$
'': closing quotation mark
    ' ''
(: opening parenthesis
    ( [ {
): closing parenthesis
    ) ] }
,: comma
    ,
--: dash
    --
```

```
.: sentence terminator
    . ! ?
:: colon or ellipsis
    : ; ...
CC: conjunction, coordinating
    & 'n and both but either et for less minus neither nor or plus so
    therefore times v. versus vs. whether yet
CD: numeral, cardinal
    mid-1890 nine-thirty forty-two one-tenth ten million 0.5 one forty-
    seven 1987 twenty '79 zero two 78-degrees eighty-four IX '60s .025
    fifteen 271,124 dozen quintillion DM2,000 ...
DT: determiner
    all an another any both del each either every half la many much nary
    neither no some such that the them these this those
EX: existential there
    there
FW: foreign word
    gemeinschaft hund ich jeux habeas Haementeria Herr K'ang-si vous
    lutihaw alai je jour objets salutaris fille quibusdam pas trop Monte
    terram fiche oui corporis ...
IN: preposition or conjunction, subordinating
    astride among uppon whether out inside pro despite on by throughout
    below within for towards near behind atop around if like until below
    next into if beside ...
JJ: adjective or numeral, ordinal
    third ill-mannered pre-war regrettable oiled calamitous first separable
    ectoplasmic battery-powered participatory fourth still-to-be-named
    multilingual multi-disciplinary ...
JJR: adjective, comparative
    bleaker braver breezier briefer brighter brisker broader bumper busier
    calmer cheaper choosier cleaner clearer closer colder commoner costlier
    cozier creamier crunchier cuter ...
JJS: adjective, superlative
    calmest cheapest choicest classiest cleanest clearest closest commonest
    corniest costliest crassest creepiest crudest cutest darkest deadliest
    dearest deepest densest dinkiest ...
LS: list item marker
    A A. B B. C C. D E F First G H I J K One SP-44001 SP-44002 SP-44005
    SP-44007 Second Third Three Two * a b c d first five four one six three
    two
MD: modal auxiliary
    can cannot could couldn't dare may might must need ought shall should
```

ASSIGNMENT -3

```
!pip install nltk spacy emoji
!python -m spacy download en_core_web_s
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.12/dist-packages (3.9.1)
Requirement already satisfied: spacy in /usr/local/lib/python3.12/dist-packages (3.8.11)
Collecting emoji
  Downloading emoji-2.15.0-py3-none-any.whl.metadata (5.7 kB)
Requirement already satisfied: click in /usr/local/lib/python3.12/dist-packages (from nltk) (8.3.1)
Requirement already satisfied: joblib in /usr/local/lib/python3.12/dist-packages (from nltk) (1.5.3)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.12/dist-packages (from nltk) (2025.11.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from nltk) (4.67.1)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.0.15)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.13)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.0.12)
Requirement already satisfied: thinc<8.4.0,>=8.3.4 in /usr/local/lib/python3.12/dist-packages (from spacy) (8.3.10)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.1.3)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.5.2)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.10)
Requirement already satisfied: weasel<0.5.0,>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (0.4.3)
Requirement already satisfied: typer-slim<1.0.0,>=0.3.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (0.21.1)
Requirement already satisfied: numpy>=1.19.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.2)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.32.4)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.1
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.1.6)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from spacy) (75.2.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (25.0)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1.8.1,<3.
Requirement already satisfied: pydantic-core==2.41.4 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1.8.1,<3.0
Requirement already satisfied: typing-extensions>=4.14.1 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1.8.1,
Requirement already satisfied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1.8.1,<
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (3.
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spac
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spac
Requirement already satisfied: blis<1.4.0,>=1.3.0 in /usr/local/lib/python3.12/dist-packages (from thinc<8.4.0,>=8.3.4->spacy) (
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.12/dist-packages (from thinc<8.4.0,>=8.3.4->sp
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.12/dist-packages (from weasel<0.5.0,>=0.4.2-
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in /usr/local/lib/python3.12/dist-packages (from weasel<0.5.0,>=0.4.2->s
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->spacy) (3.0.3)
Requirement already satisfied: wrapt in /usr/local/lib/python3.12/dist-packages (from smart-open<8.0.0,>=5.2.1->weasel<0.5.0,>=0
Downloading emoji-2.15.0-py3-none-any.whl (608 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 608.4/608.4 kB 9.8 MB/s eta 0:00:00
Installing collected packages: emoji
Successfully installed emoji-2.15.0

✗ No compatible package found for 'en_core_web_s' (spaCy v3.8.11)
```

```
import nltk
import spacy
import emoji
from collections import Counter
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
nltk.download('twitter_samples')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
[nltk_data] Downloading package twitter_samples to /root/nltk_data...
[nltk_data]   Package twitter_samples is already up-to-date!
True
```

```
file_path = "/content/enriched_posts.txt"
with open(file_path, "r", encoding="utf-8") as file:
  text_data = file.read()
print("Sample Text:\n")
print(text_data[:500])
```

```
Sample Text:

[
    {
        "text": "Just saw a LinkedIn Influencer with 'Organic Growth' written in the profile with 65K+ followers claiming that h
        "engagement": 90,
        "line_count": 3,
        "language": "English",
        "tags": [
            "Influencer",
            "Personal Growth"
        ],
        "tone": "Professional"
    },
    {
        "text": "Jobseekers, this one\u2019s for you.\n Every applic
```

```
sentences = nltk.sent_tokenize(text_data)
tokens = nltk.word_tokenize(text_data)
print("First 30 Tokens:")
print(tokens[:30])
```

```
First 30 Tokens:
['[', '{', '``', 'text', "''", ':', '``', 'Just', 'saw', 'a', 'LinkedIn', 'Influencer', 'with', "'Organic", 'Growth', "'", 'writ
```

```
nltk_pos_tags = nltk.pos_tag(tokens)
print("NLTK POS Tags (Sample):")
print(nltk_pos_tags[:20])
```

```
NLTK POS Tags (Sample):
[('[', 'RB'), ('{', '('), ('``', '``'), ('text', 'NN'), ("''", "''"), (':', ':'), ('``', '``'), ('Just', 'RB'), ('saw', 'VB'), (
```

```
nlp = spacy.load("en_core_web_sm")
doc = nlp(text_data)
print("spaCy POS Tags (Sample):")
for token in doc[:20]:
  print(token.text, token.pos_, token.tag_)
```

```
spaCy POS Tags (Sample):
[ X XX

     SPACE _SP
{ PUNCT -LRB-

         SPACE _SP
" PUNCT ``
text NOUN NN
" PUNCT ''
: PUNCT :
" PUNCT ``
Just ADV RB
saw VERB VBD
a DET DT
LinkedIn NOUN NN
Influencer NOUN NN
with ADP IN
' PUNCT ``
Organic PROPN NNP
Growth PROPN NNP
' PUNCT ''
written VERB VBN
```

```
nouns = []
verbs = []
for token in doc:
  if token.pos_ in ["NOUN", "PROPN"]:
   nouns.append(token.text.lower())
  elif token.pos_ == "VERB":
   verbs.append(token.text.lower())
   noun_freq = Counter(nouns)
verb_freq = Counter(verbs)
print("Top 10 Nouns:")
print(noun_freq.most_common(10))
print("\nTop 10 Verbs:")
print(verb_freq.most_common(10))
```

```
Top 10 Nouns:
[('text', 60), ('engagement', 60), ('language', 59), ('tags', 59), ('tone', 59), ('job', 17), ('t', 15), ('work', 11), ('growth'

Top 10 Verbs:
[('line_count', 60), ('know', 6), ('need', 6), ('have', 5), ('come', 4), ('feel', 4), ('get', 4), ('do', 4), ('let', 3), ('tell'
```

```
hashtags = [token.text for token in doc if token.text.startswith("#")]
emojis = [char for char in text_data if char in emoji.EMOJI_DATA]
print("Hashtags Found:", hashtags[:10])
print("Emojis Found:", emojis[:10])
from nltk.corpus import twitter_samples
tweets = twitter_samples.strings('positive_tweets.json')
tweet_tokens = nltk.word_tokenize(tweets[0])
tweet_pos = nltk.pos_tag(tweet_tokens)
print("Sample Tweet:")
print(tweets[0])
print("\nPOS Tags:")
print(tweet_pos)
```

```
Hashtags Found: []
Emojis Found: []
Sample Tweet:
#FollowFriday @France_Inte @PKuchly57 @Milipol_Paris for being top engaged members in my community this week :)

POS Tags:
[('#', '#'), ('FollowFriday', 'NNP'), ('@', 'NNP'), ('France_Inte', 'NNP'), ('@', 'NNP'), ('PKuchly57', 'NNP'), ('@', 'NNP'), ('
```