step 1: install and import libriers

```
# install NLTK (run once in colab)
!pip install nltk
# import requried modules
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import twitter_samples
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.12/dist-packages (3.9.1)
Requirement already satisfied: click in /usr/local/lib/python3.12/dist-packages (from nltk) (8.3.
Requirement already satisfied: joblib in /usr/local/lib/python3.12/dist-packages (from nltk) (1.5
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.12/dist-packages (from n
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from nltk) (4.67.
```

step 2: download required NLTK resourses

```
# Download datasets and modules
nltk.download('twitter_samples')
nltk.download('punkt')
#nltk download ('average_perceptron_tagger')
nltk.download('averaged_perceptron_tagger_eng')
```

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

step 3: Load sample tweets

```
#load sample tweets
tweets = twitter_samples.strings('positive_tweets.json')

#Display simple tweets
for i in range(3):
  print("tweet", i+1)
  print(tweets[i])
  print()
```

```
tweet 1
#FollowFriday @France_Inte @PKuchly57 @Milipol_Paris for being top engaged members in my communit

tweet 2
@Lamb2ja Hey James! How odd :/ Please call our Contact Centre on 02392441234 and we will be able

tweet 3
@DespiteOfficial we had a listen last night :) As You Bleed is an amazing track. When are you in
```

step 4: tokenization using tweettokenizer

```
#tweettokenizer handles emojies,hashtags,abberviations
tokenizer = TweetTokenizer(
```

```
    preserve_case = False,
    strip_handles = True,
    reduce_len = True
)

#tokenize first 5 tweets
tokenized_tweets = [tokenizer.tokenize(tweet) for tweet in tweets[:5]]

#display tokens
for i, tokens in enumerate(tokenized_tweets):
  print("tweet", i+1, "tokens:")
  print(tokens)
  print()
```

```
tweet 1 tokens:
['#followfriday', 'for', 'being', 'top', 'engaged', 'members', 'in', 'my', 'community', 'this', '

tweet 2 tokens:
['hey', 'james', '!', 'how', 'odd', ':/', 'please', 'call', 'our', 'contact', 'centre', 'on', '02

tweet 3 tokens:
['we', 'had', 'a', 'listen', 'last', 'night', ':)', 'as', 'you', 'bleed', 'is', 'an', 'amazing',

tweet 4 tokens:
['congrats', ':)']

tweet 5 tokens:
['yeaaah', 'yipppy', '!', '!', '!', 'my', 'accnt', 'verified', 'rqst', 'has', 'succeed', 'got', '
```

step 5: POS tagging using NLTK

```
#apply pos tagging
pos_tagged_tweets = [nltk.pos_tag(tokens) for tokens in tokenized_tweets]

#display pos tags
for i, pos_tags in enumerate(pos_tagged_tweets):
  print("tweet", i+1, "pos tags:")
  print(pos_tags)
  print()
```

```
tweet 1 pos tags:
[('#followfriday', 'NN'), ('for', 'IN'), ('being', 'VBG'), ('top', 'JJ'), ('engaged', 'VBN'), ('m

tweet 2 pos tags:
[('hey', 'NN'), ('james', 'NNS'), ('!', '.'), ('how', 'WRB'), ('odd', 'JJ'), (':/', 'JJ'), ('plea

tweet 3 pos tags:
[('we', 'PRP'), ('had', 'VBD'), ('a', 'DT'), ('listen', 'VBN'), ('last', 'JJ'), ('night', 'NN'),

tweet 4 pos tags:
[('congrats', 'NNS'), (':)', 'VBP')]

tweet 5 pos tags:
[('yeaaah', 'NN'), ('yipppy', 'JJ'), ('!', '.'), ('!', '.'), ('!', '.'), ('my', 'PRP$'), ('accnt'
```

step 6: POS Tagging on custom noisy text

```
# Example informal text
text = "OMG I luv this phone 😍 #awesome #AI"

# Tokenize
tokens = tokenizer.tokenize(text)
# POS tagging
```

```
tags = nltk.pos_tag(tokens)
print("Original Text:", text)
print("Tokens:", tokens)
print("POS Tags:", tags)
```

```
Original Text: OMG I luv this phone 😍 #awesome #AI
Tokens: ['omg', 'i', 'luv', 'this', 'phone', '😍', '#awesome', '#ai']
POS Tags: [('omg', 'NN'), ('i', 'NN'), ('luv', 'VBP'), ('this', 'DT'), ('phone', 'NN'), ('😍', 'V
```

```
# Extract nouns and verbs
nouns = []
verbs = []
for word, tag in tags:
    if tag.startswith('NN'):   # Nouns
         nouns.append(word)
    elif tag.startswith('VB'): # Verbs
         verbs.append(word)

print("Nouns:", nouns)
print("Verbs:", verbs)
```

```
Nouns: ['omg', 'i', 'phone', '#awesome', '#ai']
Verbs: ['luv', '😍']
```

Install and Load spaCy

```
# Install spaCy (run once in Colab)
!pip install spacy
# Download English language model
!python -m spacy download en_core_web_sm
```

```
Requirement already satisfied: spacy in /usr/local/lib/python3.12/dist-packages (3.8.11)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.12/dist-pack
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.12/dist-pack
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.12/dist-packag
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.12/dist-packages (fr
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.12/dist-packages (
Requirement already satisfied: thinc<8.4.0,>=8.3.4 in /usr/local/lib/python3.12/dist-packages (fr
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.12/dist-packages (f
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.12/dist-packages (fr
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: weasel<0.5.0,>=0.4.2 in /usr/local/lib/python3.12/dist-packages (f
Requirement already satisfied: typer-slim<1.0.0,>=0.3.0 in /usr/local/lib/python3.12/dist-package
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.12/dist-packages (fr
Requirement already satisfied: numpy>=1.19.0 in /usr/local/lib/python3.12/dist-packages (from spa
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.12/
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from spacy)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from s
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: pydantic-core==2.41.4 in /usr/local/lib/python3.12/dist-packages (
Requirement already satisfied: typing-extensions>=4.14.1 in /usr/local/lib/python3.12/dist-packag
Requirement already satisfied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12/dist-package
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-package
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requ
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (fro
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (fro
Requirement already satisfied: blis<1.4.0,>=1.3.0 in /usr/local/lib/python3.12/dist-packages (fro
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.12/dist-package
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.12/dist-packages (from type
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.12/dist-packa
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in /usr/local/lib/python3.12/dist-package
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from j
Requirement already satisfied: wrapt in /usr/local/lib/python3.12/dist-packages (from smart-open<
Collecting en-core-web-sm==3.8.0
```

```
    Downloading https://github.com/explosion/spacy-models/releases/download/en_core_web_sm-3.8.0/en
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.8/12.8 MB 48.4 MB/s eta 0:00:00
✓ Download and installation successful
You can now load the package via spacy.load('en_core_web_sm')
⚠ Restart to reload dependencies
If you are in a Jupyter or Colab notebook, you may need to restart Python in
order to load all the package's dependencies. You can do this by selecting the
'Restart kernel' or 'Restart runtime' option.
```

Import Libraries and Load Model

```
# Import spaCy
import spacy
# Load English model
nlp = spacy.load("en_core_web_sm")
```

Sample Informal Text (Tweet / Caption)

```
# Example noisy text
text = "OMG I luv this phone 😍 #awesome #AI!!! Can't wait to try it 😊"

print("Original Text:")
print(text)
```

```
Original Text:
OMG I luv this phone 😍 #awesome #AI!!! Can't wait to try it 😊
```

Tokenization and POS Tagging

```
# Process text using spaCy pipeline
doc = nlp(text)
# Display tokens with POS tags
print("\nToken\t\tPOS Tag\t\tDetailed Tag")
print("-"*50)
for token in doc:
    print(f"{token.text:12}\t{token.pos_:10}\t{token.tag_}")
```

```
Token           POS Tag         Detailed Tag
--------------------------------------------------
OMG             PROPN           NNP
I               PRON            PRP
luv             PROPN           NNP
this            DET             DT
phone           NOUN            NN
😍              NOUN            NNS
#               SYM             $
awesome         ADV             RB
#               SYM             $
AI              NOUN            NN
```

Extract Nouns and Verbs

```
# Extract nouns and verbs
nouns = []
verbs = []

for token in doc:
    if token.pos_ in ["NOUN", "PROPN"]:
        nouns.append(token.text)
```

```
        elif token.pos_ == "VERB":
            verbs.append(token.text)

print("\nExtracted Nouns:", nouns)
print("Extracted Verbs:", verbs)
```

```
Extracted Nouns: ['OMG', 'luv', 'phone', '😍', 'AI']
Extracted Verbs: []
```

Analyze Multiple Tweets (Optional for Lab)

```
# List of noisy tweets

tweets = [
    "Love this camera!!! 😍#photography #awesome",
    "OMG battery life sucks =👺totally disappointed",
    "Just bought a new laptop =💻#tech #AI",
    "LOL this update broke everything "
]

for i, tweet in enumerate(tweets, 1):
    doc = nlp(tweet)

print(f"\nTweet {i}: {tweet}")
print("Tokens and POS:")

for token in doc:
    print(token.text, "→", token.pos_)
```

```
Tweet 4: LOL this update broke everything
Tokens and POS:
LOL → PROPN
this → DET
update → NOUN
broke → VERB
everything → PRON
```