```
!pip install nltk
import pandas as pd
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
nltk.download('punkt')
nltk.download('stopwords')
```

```
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) (2023.6.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk) (4.66.1)
[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.
True
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

## Loading Dataset

```
import pandas as pd
df = pd.read_csv('/content/test-data.csv', names=['text', 'aspect'], sep='|', encoding='utf-8')
df
```

```
<ipython-input-3-345425df101b>:2: ParserWarning: Falling back to the 'python' engine because the separator encoded in utf-8 is > 1 char
  df = pd.read_csv('/content/test-data.csv', names=['text', 'aspect'], sep='|', encoding='utf-8')
```

| | text | aspect |
|---|---|---|
| 0 | रियर कैमरा , डुअल टोन एलईडी फ्लैश और फिंगरप्रि... | स्पेसिफिकेशन |
| 1 | हालांकि , इस बार शाओमी ने फोन में स्पीकर ग्रिल... | डिज़ाइन |
| 2 | निचले हिस्से में चार्जिंग और डेटा ट्रांसफर के ... | डिज़ाइन |
| 3 | टॉप में आपको 3.5 एमएम ऑडियो जैक के साथ इंफ्रार... | स्पेसिफिकेशन |
| 4 | पावर और वॉल्यूम बटन दायीं तरफ हैं और इन तक ऊंग... | डिज़ाइन |
| ... | ... | ... |
| 195 | इसके अलावा हमने देखा कि कम रोशनी में पी2 स्मार... | परफॉर्मेंस |
| 196 | कैमरे से सही कलर रीप्रोड्यूस होते हैं लेकिन सि... | कैमरा |
| 197 | लेनोवो पी2 में कैमरा ऐप के6 पावर और के6 नोट की... | कैमरा |
| 198 | सिर्फ एक टैप से ही कैमरा को एक्सेस किया जा सकत... | कैमरा |
| 199 | पी2 से शानदार क्वालिटी के साथ फुल - एचडी वीडिय... | कैमरा |

200 rows × 2 columns

```
!pip install spacy
!python -m spacy download xx_ent_wiki_sm
```

```
Requirement already satisfied: spacy in /usr/local/lib/python3.10/dist-packages (3.6.1)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.10/dist-packages (from spacy) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (1.0.10)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.10/dist-packages (from spacy) (2.0.8)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.10/dist-packages (from spacy) (3.0.9)
Requirement already satisfied: thinc<8.2.0,>=8.1.8 in /usr/local/lib/python3.10/dist-packages (from spacy) (8.1.12)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.10/dist-packages (from spacy) (1.1.2)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.10/dist-packages (from spacy) (2.4.8)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.10/dist-packages (from spacy) (2.0.10)
Requirement already satisfied: typer<0.10.0,>=0.3.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (0.9.0)
Requirement already satisfied: pathy>=0.10.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (0.10.3)
Requirement already satisfied: smart-open<7.0.0,>=5.2.1 in /usr/local/lib/python3.10/dist-packages (from spacy) (6.4.0)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (4.66.1)
```

```
Requirement already satisfied: numpy>=1.15.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (1.23.5)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (2.31.0)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.10/dist-packages (from spacy) (1.10.13)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from spacy) (3.1.2)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from spacy) (67.7.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (23.2)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in /usr/local/lib/python3.10/dist-packages (from spacy) (3.3.0)
Requirement already satisfied: typing-extensions>=4.2.0 in /usr/local/lib/python3.10/dist-packages (from pydantic!=1.8,!=1.8.1,<3.0.0
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0,>=2.13.0->spa
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (2
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (2
Requirement already satisfied: blis<0.8.0,>=0.7.8 in /usr/local/lib/python3.10/dist-packages (from thinc<8.2.0,>=8.1.8->spacy) (0.7.1
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.10/dist-packages (from thinc<8.2.0,>=8.1.8->spacy)
Requirement already satisfied: click<9.0.0,>=7.1.1 in /usr/local/lib/python3.10/dist-packages (from typer<0.10.0,>=0.3.0->spacy) (8.1
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->spacy) (2.1.3)
2023-12-13 03:12:02.038338: E tensorflow/compiler/xla/stream_executor/cuda/cuda_dnn.cc:9342] Unable to register cuDNN factory: Attemp
2023-12-13 03:12:02.038432: E tensorflow/compiler/xla/stream_executor/cuda/cuda_fft.cc:609] Unable to register cuFFT factory: Attempt
2023-12-13 03:12:02.038491: E tensorflow/compiler/xla/stream_executor/cuda/cuda_blas.cc:1518] Unable to register cuBLAS factory: Atte
2023-12-13 03:12:02.065838: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2023-12-13 03:12:07.728559: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
Collecting xx-ent-wiki-sm==3.6.0
  Downloading https://github.com/explosion/spacy-models/releases/download/xx_ent_wiki_sm-3.6.0/xx_ent_wiki_sm-3.6.0-py3-none-any.whl
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 11.1/11.1 MB 13.1 MB/s eta 0:00:00
Requirement already satisfied: spacy<3.7.0,>=3.6.0 in /usr/local/lib/python3.10/dist-packages (from xx-ent-wiki-sm==3.6.0) (3.6.1)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->xx-e
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->xx-e
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->xx-ent
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->xx-ent-wiki-
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->xx-ent-wik
Requirement already satisfied: thinc<8.2.0,>=8.1.8 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->xx-ent-wiki-
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->xx-ent-wiki
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->xx-ent-wiki-
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->xx-ent-w
Requirement already satisfied: typer<0.10.0,>=0.3.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->xx-ent-wiki
Requirement already satisfied: pathy>=0.10.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->xx-ent-wiki-sm==3.
Requirement already satisfied: smart-open<7.0.0,>=5.2.1 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->xx-ent-
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->xx-ent-wiki-
Requirement already satisfied: numpy>=1.15.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->xx-ent-wiki-sm==3.
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.6.0->xx-ent-w
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.10/dist-packages (from spacy<3.7.0,>=3.
```

## Remove Stop words - store into preprocessed.csv

```
import spacy


nlp = spacy.load("xx_ent_wiki_sm")

with open('/content/hindi_stopwords.txt', 'r', encoding='utf-8') as stopwords_file:
    custom_stopwords = stopwords_file.read().split()


def preprocess_text(text):

    doc = nlp(text)


    tokens = [token.text for token in doc if token.text.lower() not in custom_stopwords and not token.is_punct]


    cleaned_text = " ".join(tokens)

    return cleaned_text

df['preprocessed_text'] = df['text'].apply(preprocess_text)

df.to_csv('/content/preprocessed_train.csv', index=False)
df
```

| | text | aspect | preprocessed_text |
|---|---|---|---|
| 0 | रियर कैमरा , डुअल टोन एलईडी फ्लैश और फिंगरप्रि... | स्पेसिफिकेशन | रियर कैमरा डुअल टोन एलईडी फ्लैश फिंगरप्रिंट स्... |
| 1 | हालांकि , इस बार शाओमी ने फोन में स्पीकर ग्रिल... | डिज़ाइन | हालांकि बार शाओमी फोन स्पीकर ग्रिल निचले हिस्स... |
| 2 | निचले हिस्से में चार्जिंग और डेटा ट्रांसफर के ... | डिज़ाइन | निचले हिस्से चार्जिंग डेटा ट्रांसफर यूएसबी पोर्ट |
| 3 | टॉप में आपको 3.5 एमएम ऑडियो जैक के साथ इंफ्रार... | स्पेसिफिकेशन | टॉप आपको 3.5 एमएम ऑडियो जैक इंफ्रारेड एमिटर मि... |
| 4 | पावर और वॉल्यूम बटन दायीं तरफ हैं और इन तक ऊंग... | डिज़ाइन | पावर वॉल्यूम बटन दायीं तरफ ऊंगलियों पहुंचना आसान |
| ... | ... | ... | ... |
| 195 | इसके अलावा हमने देखा कि कम रोशनी में पी2 स्मार... | परफॉर्मेंस | अलावा हमने देखा कम रोशनी पी2 स्मार्टफोन सब्जेक... |
| 196 | कैमरे से सही कलर रीप्रोड्यूस होते हैं लेकिन सि... | कैमरा | कैमरे सही कलर रीप्रोड्यूस सिर्फ तस्वीरो अच्छी ... |
| 197 | लेनोवो पी2 में कैमरा ऐप के6 पावर और के6 नोट की... | कैमरा | लेनोवो पी2 कैमरा ऐप के6 पावर के6 नोट इस्तेमाल ... |

## Tokenize

```
from nltk.tokenize import word_tokenize

def preprocess_and_tokenize(text):
    return word_tokenize(text)

df['tokens'] = df['preprocessed_text'].apply(preprocess_and_tokenize)
```

## Store into final.csv

```
df.to_csv("/content/final.csv")
```

## POS tagging using CRF

```
pip install sklearn-crfsuite
```

```
Collecting sklearn-crfsuite
  Downloading sklearn_crfsuite-0.3.6-py2.py3-none-any.whl (12 kB)
Collecting python-crfsuite>=0.8.3 (from sklearn-crfsuite)
  Downloading python_crfsuite-0.9.9-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (993 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 993.5/993.5 kB 5.9 MB/s eta 0:00:00
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from sklearn-crfsuite) (1.16.0)
Requirement already satisfied: tabulate in /usr/local/lib/python3.10/dist-packages (from sklearn-crfsuite) (0.9.0)
Requirement already satisfied: tqdm>=2.0 in /usr/local/lib/python3.10/dist-packages (from sklearn-crfsuite) (4.66.1)
Installing collected packages: python-crfsuite, sklearn-crfsuite
Successfully installed python-crfsuite-0.9.9 sklearn-crfsuite-0.3.6
```

```
import nltk
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
```

```
!pip install --force-reinstall --no-dependencies "scikit-learn==0.24.2"
```

```
Collecting scikit-learn==0.24.2
  Downloading scikit-learn-0.24.2.tar.gz (7.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.5/7.5 MB 20.0 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  error: subprocess-exited-with-error

  × Preparing metadata (pyproject.toml) did not run successfully.
  │ exit code: 1
```

```
  ╰─> See above for output.

  note: This error originates from a subprocess, and is likely not a problem with pip.
  Preparing metadata (pyproject.toml) ... error
error: metadata-generation-failed

× Encountered error while generating package metadata.
╰─> See above for output.

note: This is an issue with the package mentioned above, not pip.
hint: See above for details.
```

```
import pandas as pd
import nltk
from sklearn import metrics
from nltk.tokenize import word_tokenize
from nltk import pos_tag
from sklearn_crfsuite import CRF
from sklearn.model_selection import train_test_split
from sklearn_crfsuite.metrics import flat_classification_report

df['pos_tags'] = df['tokens'].apply(pos_tag)

def word2features(sent, i):
    word = sent[i][0]

    features = {
        'bias': 1.0,
        'word.lower()': word.lower(),
        'word[-3:]': word[-3:],
        'word[-2:]': word[-2:],
        'word.isupper()': word.isupper(),
        'word.istitle()': word.istitle(),
        'word.isdigit()': word.isdigit(),
    }
    if i > 0:
        word1 = sent[i-1][0]
        features.update({
            '-1:word.lower()': word1.lower(),
            '-1:word.istitle()': word1.istitle(),
            '-1:word.isupper()': word1.isupper(),
        })
    else:
        features['BOS'] = True

    if i < len(sent)-1:
        word1 = sent[i+1][0]
        features.update({
            '+1:word.lower()': word1.lower(),
            '+1:word.istitle()': word1.istitle(),
            '+1:word.isupper()': word1.isupper(),
        })
    else:
        features['EOS'] = True

    return features

def sent2features(sent):
    return [word2features(sent, i) for i in range(len(sent))]

def sent2labels(sent):
    return [label for token, label in sent]

X = [sent2features(sent) for sent in df['pos_tags']]
y = [sent2labels(sent) for sent in df['pos_tags']]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


crf = CRF(algorithm='lbfgs', c1=0.1, c2=0.1, max_iterations=100)
try:
    crf.fit(X_train, y_train)
except AttributeError:
    pass
predictions = crf.predict(X_test)

# Print words and labels
for sent, true_labels, pred_labels in zip(X_test, y_test, predictions):
    for word, true_label, pred_label in zip(sent, true_labels, pred_labels):
        print(f'Word: {word["word.lower()"]}, True Label: {true_label}, Predicted Label: {pred_label}')
    print('\n---\n')
```

```
Word: हमारा, True Label: JJ, Predicted Label: JJ
Word: मानना, True Label: NNP, Predicted Label: NNP
Word: रेडमी, True Label: NNP, Predicted Label: NNP
Word: नोट, True Label: VBD, Predicted Label: VBD
Word: 4, True Label: CD, Predicted Label: CD
```

```
Word: सेगमेंट, True Label: NNP, Predicted Label: NNP
Word: थोड़े, True Label: NNP, Predicted Label: NNP
Word: महंगे, True Label: NNP, Predicted Label: NNP
Word: रेंज, True Label: NNP, Predicted Label: NNP
Word: स्मार्टफोन, True Label: NNP, Predicted Label: NNP
Word: मज़बूत, True Label: NNP, Predicted Label: NNP
Word: चुनौती, True Label: NNP, Predicted Label: NNP
Word: देगा, True Label: NN, Predicted Label: NN

---

Word: 5.5, True Label: CD, Predicted Label: CD
Word: इंच, True Label: JJ, Predicted Label: NNP
Word: फुल, True Label: NNP, Predicted Label: NNP
Word: एचडी, True Label: NNP, Predicted Label: NNP
Word: आईपीएस, True Label: NNP, Predicted Label: NNP
Word: डिस्प्ले, True Label: NNP, Predicted Label: NNP
Word: रिज़ॉल्यूशन, True Label: VBD, Predicted Label: NNP
Word: 1080x1920, True Label: CD, Predicted Label: NNP
Word: पिक्सल, True Label: NN, Predicted Label: NN

---

Word: ऐसा, True Label: JJ, Predicted Label: JJ
Word: लगता, True Label: NNP, Predicted Label: NNP
Word: कंपनी, True Label: NNP, Predicted Label: NNP
Word: रेडमी, True Label: NNP, Predicted Label: NNP
Word: नोट, True Label: VBD, Predicted Label: VBD
Word: 3, True Label: CD, Predicted Label: CD
Word: सीमित, True Label: JJ, Predicted Label: $
Word: 32, True Label: CD, Predicted Label: CD
Word: जीबी, True Label: JJ, Predicted Label: JJ
Word: माइक्रोएसडी, True Label: NNP, Predicted Label: NNP
Word: कार्ड, True Label: NNP, Predicted Label: NNP
Word: सपोर्ट, True Label: NNP, Predicted Label: NNP
Word: क्षमता, True Label: NNP, Predicted Label: NNP
Word: शिकायत, True Label: NNP, Predicted Label: NNP
Word: गंभीरता, True Label: NNP, Predicted Label: NNP
Word: लिया, True Label: NN, Predicted Label: NN

---

Word: लेनोवो, True Label: JJ, Predicted Label: JJ
Word: के6, True Label: NNP, Predicted Label: NNP
Word: पावर, True Label: NNP, Predicted Label: NNP
Word: के6, True Label: NNP, Predicted Label: NNP
Word: नोट, True Label: NNP, Predicted Label: NNP
Word: कैपेसिटिव, True Label: NNP, Predicted Label: NNP
Word: बटन, True Label: NNP, Predicted Label: NNP
Word: बैकलिट, True Label: NNP, Predicted Label: NNP
Word: वजह, True Label: NNP, Predicted Label: NNP
Word: आई, True Label: NNP, Predicted Label: NNP
```

```
df
```

| | text | aspect | preprocessed_text | tokens | pos_tags |
|---|---|---|---|---|---|
| **0** | रियर कैमरा , डुअल टोन एलईडी फ्लैश और फिंगरप्रि... | स्पेसिफिकेशन | रियर कैमरा डुअल टोन एलईडी फ्लैश फिंगरप्रिंट स्... | [रियर, कैमरा, डुअल, टोन, एलईडी, फ्लैश, फिंगरप्... | [(रियर, JJ), (कैमरा, NNP), (डुअल, NNP), (टोन, ... |
| **1** | हालांकि , इस बार शाओमी ने फोन में स्पीकर ग्रिल... | डिज़ाइन | हालांकि बार शाओमी फोन स्पीकर ग्रिल निचले हिस्स... | [हालांकि, बार, शाओमी, फोन, स्पीकर, ग्रिल, निचल... | [(हालांकि, JJ), (बार, NNP), (शाओमी, NNP), (फोन... |
| **2** | निचले हिस्से में चार्जिंग और डेटा ट्रांसफर के ... | डिज़ाइन | निचले हिस्से चार्जिंग डेटा ट्रांसफर यूएसबी पोर्ट | [निचले, हिस्से, चार्जिंग, डेटा, ट्रांसफर, यूएस... | [(निचले, JJ), (हिस्से, NNP), (चार्जिंग, NNP), ... |
| **3** | टॉप में आपको 3.5 एमएम ऑडियो जैक के साथ इंफ्रार... | स्पेसिफिकेशन | टॉप आपको 3.5 एमएम ऑडियो जैक इंफ्रारेड एमिटर मि... | [टॉप, आपको, 3.5, एमएम, ऑडियो, जैक, इंफ्रारेड, ... | [(टॉप, JJ), (आपको, $), (3.5, CD), (एमएम, NNP),... |
| **4** | पावर और वॉल्यूम बटन दायीं तरफ हैं और इन तक उंग... | डिज़ाइन | पावर वॉल्यूम बटन दायीं तरफ ऊंगलियों पहुंचना आसान | [पावर, वॉल्यूम, बटन, दायीं, तरफ, ऊंगलियों, पहु... | [(पावर, JJ), (वॉल्यूम, NNP), (बटन, NNP), (दायी... |

## POS tagging using HMM - Viterbi Algorithm

```
output_file = 'output_file.txt'

sentences = df['tokens'].apply(lambda tokens: ' '.join(tokens)).str.cat(sep='. ').replace(' .', '.')
with open(output_file, 'w', encoding='utf-8') as file:
    file.write(sentences)
```

```
import sys
import math
from decimal import *
import codecs

tag_list = set()
tag_count = {}
word_set = set()


def parse_traindata():
    fin = "/content/train_data.txt"
    output_file = "/content/hmmmodel.txt"
    wordtag_list = []

    try:
        input_file = codecs.open(fin, mode = 'r', encoding="utf-8")
        lines = input_file.readlines()
        for line in lines:
            line = line.strip('\n')
            data = line.split(" ")
            wordtag_list.append(data)

        input_file.close()
        return wordtag_list

    except IOError:
        fo = codecs.open(output_file,mode = 'w',encoding="utf-8")
        fo.write("File not found: {}".format(fin))
        fo.close()
        sys.exit()


def transition_count(train_data):
    global tag_list
    global word_set
    transition_dict = {}
    global tag_count
    for value in train_data:
        previous = "start"
        for data in value:
            i = data[::-1]
            word = data[:-i.find("/") - 1]
            word_set.add(word.lower())
            data = data.split("/")
            tag = data[-1]
            tag_list.add(tag)


            if tag in tag_count:
                tag_count[tag] += 1
            else:
                tag_count[tag] = 1


            if (previous + "~tag~" + tag) in transition_dict:
                transition_dict[previous + "~tag~" + tag] += 1
                previous = tag
            else:
                transition_dict[previous + "~tag~" + tag] = 1
                previous = tag

    return transition_dict


def transition_probability(train_data):
    count_dict = transition_count(train_data)
    prob_dict = {}
    for key in count_dict:
        den = 0
        val = key.split("~tag~")[0]
        for key_2 in count_dict:
            if key_2.split("~tag~")[0] == val:
                den += count_dict[key_2]
        prob_dict[key] = Decimal(count_dict[key])/(den)
    return prob_dict
```

```
def transition_smoothing(train_data):
    transition_prob = transition_probability(train_data)
    for tag in tag_list:
        if "start" + tag not in  transition_prob:
            transition_prob[("start" + "~tag~" + tag)] = Decimal(1) / Decimal(len(word_set) + tag_count[tag])
    for tag1 in tag_list:
        for tag2 in tag_list:
            if (tag1 +"~tag~" + tag2) not in transition_prob:
                transition_prob[(tag1+"~tag~"+tag2)] = Decimal(1)/Decimal(len(word_set) + tag_count[tag1])
    return transition_prob


def emission_count(train_data):
    count_word = {}
    for value in train_data:
        for data in value:
            i = data[::-1]
            word = data[:-i.find("/") - 1]
            tag = data.split("/")[-1]
            if word.lower() + "/" + tag in count_word:
                count_word[word.lower() + "/" + tag] +=1
            else:
                count_word[word.lower() + "/" + tag] = 1
    return count_word


def emission_probability(train_data):
    global tag_count
    word_count = emission_count(train_data)
    emission_prob_dict = {}
    for key in word_count:
        emission_prob_dict[key] = Decimal(word_count[key])/tag_count[key.split("/")[-1]]
    return emission_prob_dict



train_data = parse_traindata()
transition_model = transition_smoothing(train_data)
emission_model = emission_probability(train_data)

fout = codecs.open("/content/hmmmodel.txt", mode ='w', encoding="utf-8")
for key, value in transition_model.items():
    fout.write('%s:%s\n' % (key, value))

fout.write(u'Emission Model\n')
for key, value in emission_model.items():
    fout.write('%s:%s\n' % (key, value))
```

```
from decimal import Decimal
from pathlib import Path

tag_set = set()
word_set = set()

def parse_traindata():
    fin = "/content/hmmmodel.txt"
    output_file = "/content/hmmmoutput.txt"
    transition_prob = {}
    emission_prob = {}
    tag_list = []
    tag_count = {}
    global tag_set

    try:
        input_file = Path(fin).read_text(encoding="utf-8").splitlines()
        flag = False

        for line in input_file:
            line = line.strip('\n')

            if line != "Emission Model":
                i = line[::-1]
                key_insert = line[:-i.find(":")-1]
                value_insert = line.split(":")[-1]

                if flag == False:
                    transition_prob[key_insert] = value_insert
                    if (key_insert.split("~tag~")[0] not in tag_list) and (key_insert.split("~tag~")[0] != "start"):
                        tag_list.append(key_insert.split("~tag~")[0])
                else:
                    emission_prob[key_insert] = value_insert
                    val = key_insert.split("/")[-1]
                    j = key_insert[::-1]
                    word = key_insert[:-j.find("/")-1].lower()
                    word_set.add(word)
                    if val in tag_count:
                        tag_count[val] += 1
                    else:
                        tag_count[val] = 1
                    tag_set.add(val)
            else:
                flag = True
                continue

        return tag_list, transition_prob, emission_prob, tag_count, word_set

    except FileNotFoundError:
        output_file = "/content/hmmmoutput.txt"
        Path(output_file).write_text(f"File not found: {fin}")
        sys.exit()


def viterbi_algorithm(sentence, tag_list, transition_prob, emission_prob, tag_count, word_set):
    global tag_set
    sentence = sentence.strip("\n")
    word_list = sentence.split(" ")
    current_prob = {}

    for tag in tag_list:
        tp = Decimal(0)
        em = Decimal(0)
        if "start~tag~"+tag in transition_prob:
            tp = Decimal(transition_prob["start~tag~"+tag])

        if word_list[0].lower() in word_set:
            if (word_list[0].lower()+"/"+tag) in emission_prob:
                em = Decimal(emission_prob[word_list[0].lower()+"/"+tag])
                current_prob[tag] = tp * em
        else:
            em = Decimal(1) / (tag_count[tag] + len(word_set))
            current_prob[tag] = tp

    if len(word_list) == 1:
        max_path = max(current_prob, key=current_prob.get)
        return max_path
```

```
        else:
            for i in range(1, len(word_list)):
                previous_prob = current_prob
                current_prob = {}
                locals()['dict{}'.format(i)] = {}
                previous_tag = ""

                for tag in tag_list:
                    if word_list[i].lower() in word_set:
                        if word_list[i].lower()+"/"+tag in emission_prob:
                            em = Decimal(emission_prob[word_list[i].lower()+"/"+tag])
                            max_prob, previous_state = max(
                                (Decimal(previous_prob[previous_tag]) * Decimal(transition_prob[previous_tag + "~tag~" + tag]) * em, previous_ta
                            current_prob[tag] = max_prob
                            locals()['dict{}'.format(i)][previous_state + "~" + tag] = max_prob
                            previous_tag = previous_state
                        else:
                            em = Decimal(1) / (tag_count[tag] + len(word_set))
                            max_prob, previous_state = max((Decimal(previous_prob[previous_tag]) * Decimal(transition_prob[previous_tag+"~tag~"+tag]
                            current_prob[tag] = max_prob
                            locals()['dict{}'.format(i)][previous_state + "~" + tag] = max_prob
                            previous_tag = previous_state

                if i == len(word_list)-1:
                    max_path = ""
                    last_tag = max(current_prob, key=current_prob.get)
                    max_path = max_path + last_tag + " " + previous_tag
                    for j in range(len(word_list)-1, 0, -1):
                        for key in locals()['dict{}'.format(j)]:
                            data = key.split("~")
                            if data[-1] == previous_tag:
                                max_path = max_path + " " + data[0]
                                previous_tag = data[0]
                                break
                    result = max_path.split()
                    result.reverse()
                    return " ".join(result)


    tag_list, transition_model, emission_model, tag_count, word_set = parse_traindata()
    fin = "/content/output_file.txt"
    input_file = Path(fin).read_text(encoding="utf-8").splitlines()
    fout = Path("/content/hmmoutput.txt").open(mode='w', encoding="utf-8")

    for sentence in input_file:
        path = viterbi_algorithm(sentence, tag_list, transition_model, emission_model, tag_count, word_set)
        sentence = sentence.strip("\n")
        word = sentence.split(" ")
        tag = path.split(" ")
        for j in range(0, len(word)):
            if j == len(word)-1:
                fout.write(word[j] + "/" + tag[j] + u'\n')
            else:
                fout.write(word[j] + "/" + tag[j] + " ")

    predicted = Path("/content/hmmoutput.txt").read_text(encoding="utf-8").splitlines()
    expected = Path("/content/test_tagged.txt").read_text(encoding="utf-8").splitlines()

    c = 0
    total = 0

    for line in predicted:
        u = line.split(" ")
        total += len(u)
        a = expected.pop(0).split(" ")
        for i in range(min(len(u), len(a))):
            if(a[i] != u[i]):
                c += 1

    print("Wrong Predictions =", c)
    print("Total Predictions =", total)
    print("Accuracy is =", 100 - (c/total * 100), "%")


     Wrong Predictions = 14
     Total Predictions = 1738
     Accuracy is = 99.19447640966628 %
```

```
file_path = '/content/hmmoutput.txt'

with open(file_path, 'r', encoding='utf-8') as file:
    file_contents = file.read()

print(file_contents)
```

```
रियर/QFC कैमरा/QF डुअल/NN टोन/PSP एलईडी/NN फ्लैश/UNK फिंगरप्रिंट/VM स्कैनर/VAUX सेटअप/VAUX रेडमी/SYM नोट/NN 3/QC वाला./NN हालांकि/CC बार/NN शाओमी/PS
```

## NER using polyglot

```
!pip install polyglot
```

```
Collecting polyglot
  Downloading polyglot-16.7.4.tar.gz (126 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 126.3/126.3 kB 1.1 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: polyglot
  Building wheel for polyglot (setup.py) ... done
  Created wheel for polyglot: filename=polyglot-16.7.4-py2.py3-none-any.whl size=52558 sha256=16e8d3bbb0321b5250133a96e1eb11b96b062a73bf
  Stored in directory: /root/.cache/pip/wheels/aa/92/4a/b172589446ba537db3bdb9a1f2204f27fe71217981c14ac368
Successfully built polyglot
Installing collected packages: polyglot
Successfully installed polyglot-16.7.4
```

```
# pip install polyglot
!pip install PyICU
!pip install pycld2
```

```
Collecting PyICU
  Downloading PyICU-2.12.tar.gz (260 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 260.0/260.0 kB 1.8 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Building wheels for collected packages: PyICU
  Building wheel for PyICU (pyproject.toml) ... done
  Created wheel for PyICU: filename=PyICU-2.12-cp310-cp310-linux_x86_64.whl size=1754545 sha256=b236ec7d002e14c7b1cd20f97e1d3664170bd25f
  Stored in directory: /root/.cache/pip/wheels/74/60/95/66d97ac2fdc8be8e526c4254047405fe77feaf064282d1ad07
Successfully built PyICU
Installing collected packages: PyICU
Successfully installed PyICU-2.12
Collecting pycld2
  Downloading pycld2-0.41.tar.gz (41.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 41.4/41.4 MB 17.5 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: pycld2
  Building wheel for pycld2 (setup.py) ... done
  Created wheel for pycld2: filename=pycld2-0.41-cp310-cp310-linux_x86_64.whl size=9904070 sha256=d0f7be10fab89864859df12144ac653f134c61
  Stored in directory: /root/.cache/pip/wheels/be/81/31/240c89c845e008a93d98542325270007de595bfd356eb0b06c
Successfully built pycld2
Installing collected packages: pycld2
Successfully installed pycld2-0.41
```

```
!pip install polyglot numpy morfessor pycld2 pyicu
```

```
Requirement already satisfied: polyglot in /usr/local/lib/python3.10/dist-packages (16.7.4)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (1.23.5)
Collecting morfessor
  Downloading Morfessor-2.0.6-py3-none-any.whl (35 kB)
Requirement already satisfied: pycld2 in /usr/local/lib/python3.10/dist-packages (0.41)
Requirement already satisfied: pyicu in /usr/local/lib/python3.10/dist-packages (2.12)
Installing collected packages: morfessor
Successfully installed morfessor-2.0.6
```

```
!polyglot download embeddings2.hi
```

```
[polyglot_data] Downloading package embeddings2.hi to
[polyglot_data]     /root/polyglot_data...
```

```
!polyglot download ner2.hi
```

```
[polyglot_data] Downloading package ner2.hi to /root/polyglot_data...
```

```
from polyglot.text import Text
import pandas as pd
```

```
df['ner_results'] = df['text'].apply(lambda text: Text(text).entities)

for index, row in df.iterrows():
    print(f'Index: {index}')
    for entity in row['ner_results']:
        print(f'Entity: {entity[0]}, Label: {entity.tag}')
    print('\n---\n')
```

```
Index: 0

---

Index: 1

---

Index: 2

---

Index: 3

---

Index: 4

---

Index: 5

---

Index: 6

---

Index: 7

---

Index: 8

---

Index: 9

---

Index: 10

---

Index: 11
Entity: चीन, Label: I-LOC

---

Index: 12
Entity: भारत, Label: I-LOC
Entity: ग्रे, Label: I-LOC

---

Index: 13

---
```

```
df
```

| | text | aspect | preprocessed_text | tokens | pos_tags | ner_results |
|---|---|---|---|---|---|---|
| 0 | रियर कैमरा , डुअल टोन एलईडी फ्लैश और फिंगरप्रि... | स्पेसिफिकेशन | रियर कैमरा डुअल टोन एलईडी फ्लैश फिंगरप्रिंट स्... | [रियर, कैमरा, डुअल, टोन, एलईडी, फ्लैश, फिंगरप्... | [(रियर, JJ), (कैमरा, NNP), (डुअल, NNP), (टोन, ... | [] |
| 1 | हालांकि , इस बार शाओमी ने फोन में स्पीकर ग्रिल... | डिज़ाइन | हालांकि बार शाओमी फोन स्पीकर ग्रिल निचले हिस्स... | [हालांकि, बार, शाओमी, फोन, स्पीकर, ग्रिल, निचल... | [(हालांकि, JJ), (बार, NNP), (शाओमी, NNP), (फोन... | [] |
| 2 | निचले हिस्से में चार्जिंग और डेटा ट्रांसफर के ... | डिज़ाइन | निचले हिस्से चार्जिंग डेटा ट्रांसफर यूएसबी पोर्ट | [निचले, हिस्से, चार्जिंग, डेटा, ट्रांसफर, यूएस... | [(निचले, JJ), (हिस्से, NNP), (चार्जिंग, NNP), ... | [] |
| 3 | टॉप में आपको 3.5 एमएम ऑडियो जैक के साथ इंफ्रार... | स्पेसिफिकेशन | टॉप आपको 3.5 एमएम ऑडियो जैक इंफ्रारेड एमिटर मि... | [टॉप, आपको, 3.5, एमएम, ऑडियो, जैक, इंफ्रारेड, ... | [(टॉप, JJ), (आपको, $), (3.5, CD), (एमएम, NNP),... | [] |
| 4 | पावर और वॉल्यूम बटन दायीं तरफ हैं और इन तक ऊंग... | डिज़ाइन | पावर वॉल्यूम बटन दायीं तरफ ऊंगलियों पहुंचना आसान | [पावर, वॉल्यूम, बटन, दायीं, तरफ, ऊंगलियों, पहु... | [(पावर, JJ), (वॉल्यूम, NNP), (बटन, NNP), (दायी... | [] |

## NER using IndicNER

```
!pip3 install transformers
!pip3 install datasets
!pip3 install sentencepiece
!pip3 install seqeval
```

```
Requirement already satisfied: transformers in /usr/local/lib/python3.10/dist-packages (4.35.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from transformers) (3.13.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.16.4 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.19.4)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (1.23.5)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from transformers) (23.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (6.0.1)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (2023.6.3)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from transformers) (2.31.0)
Requirement already satisfied: tokenizers<0.19,>=0.14 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.15.0)
Requirement already satisfied: safetensors>=0.3.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.4.1)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packages (from transformers) (4.66.1)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,>=0.16.4->transf
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,>=0.16
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (3.3
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (2023.11.1
Collecting datasets
  Downloading datasets-2.15.0-py3-none-any.whl (521 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 521.2/521.2 kB 2.2 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from datasets) (1.23.5)
Requirement already satisfied: pyarrow>=8.0.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (10.0.1)
Collecting pyarrow-hotfix (from datasets)
  Downloading pyarrow_hotfix-0.6-py3-none-any.whl (7.9 kB)
Collecting dill<0.3.8,>=0.3.0 (from datasets)
  Downloading dill-0.3.7-py3-none-any.whl (115 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 115.3/115.3 kB 10.5 MB/s eta 0:00:00
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from datasets) (1.5.3)
Requirement already satisfied: requests>=2.19.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (2.31.0)
Requirement already satisfied: tqdm>=4.62.1 in /usr/local/lib/python3.10/dist-packages (from datasets) (4.66.1)
Requirement already satisfied: xxhash in /usr/local/lib/python3.10/dist-packages (from datasets) (3.4.1)
Collecting multiprocess (from datasets)
  Downloading multiprocess-0.70.15-py310-none-any.whl (134 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 134.8/134.8 kB 15.1 MB/s eta 0:00:00
Requirement already satisfied: fsspec[http]<=2023.10.0,>=2023.1.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (2023.6.
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-packages (from datasets) (3.9.1)
```

```
Requirement already satisfied: huggingface-hub>=0.18.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (0.19.4)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from datasets) (23.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from datasets) (6.0.1)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (23.1.0)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (6.0.4)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.9.3)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.4.0)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.3.1)
Requirement already satisfied: async-timeout<5.0,>=4.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (4.0.3)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.18.0->datasets) (3.13.1)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.18.0->d
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->datasets)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->datasets) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->datasets) (2.0.7
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->datasets) (2023.
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2023.3.post1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.1->pandas->datasets) (1
Installing collected packages: pyarrow-hotfix, dill, multiprocess, datasets
Successfully installed datasets-2.15.0 dill-0.3.7 multiprocess-0.70.15 pyarrow-hotfix-0.6
Collecting sentencepiece
```

```
# Import all the necessary classes and initialize the tokenizer and model.
from transformers import AutoTokenizer, AutoModelForTokenClassification
import torch

tokenizer = AutoTokenizer.from_pretrained("ai4bharat/IndicNER")

model = AutoModelForTokenClassification.from_pretrained("ai4bharat/IndicNER")
```

```
tokenizer_config.json: 100%  346/346 [00:00<00:00, 18.5kB/s]
vocab.txt: 100%  872k/872k [00:00<00:00, 12.6MB/s]
tokenizer.json: 100%  1.72M/1.72M [00:00<00:00, 27.2MB/s]
special_tokens_map.json: 100%  112/112 [00:00<00:00, 2.97kB/s]
config.json: 100%  1.19k/1.19k [00:00<00:00, 35.5kB/s]
pytorch_model.bin: 100%  667M/667M [00:10<00:00, 81.3MB/s]
```

```
def get_predictions( sentence, tokenizer, model ):
  # Let us first tokenize the sentence - split words into subwords
  tok_sentence = tokenizer(sentence, return_tensors='pt')

  with torch.no_grad():
    # we will send the tokenized sentence to the model to get predictions
    logits = model(**tok_sentence).logits.argmax(-1)

    # We will map the maximum predicted class id with the class label
    predicted_tokens_classes = [model.config.id2label[t.item()] for t in logits[0]]

    predicted_labels = []

    previous_token_id = 0
    # we need to assign the named entity label to the head word and not the following sub-words
    word_ids = tok_sentence.word_ids()
    for word_index in range(len(word_ids)):
        if word_ids[word_index] == None:
            previous_token_id = word_ids[word_index]
        elif word_ids[word_index] == previous_token_id:
            previous_token_id = word_ids[word_index]
        else:
            predicted_labels.append( predicted_tokens_classes[ word_index ] )
            previous_token_id = word_ids[word_index]

    return predicted_labels
```

```
df
```

| | text | aspect | preprocessed_text | tokens | pos_tags | ner_results |
|---|---|---|---|---|---|---|
| 0 | रियर कैमरा , डुअल टोन एलईडी फ्लैश और फिंगरप्रि... | स्पेसिफिकेशन | रियर कैमरा डुअल टोन एलईडी फ्लैश फिंगरप्रिंट स्... | [रियर, कैमरा, डुअल, टोन, एलईडी, फ्लैश, फिंगरप्... | [(रियर, JJ), (कैमरा, NNP), (डुअल, NNP), (टोन, ... | [] |
| 1 | हालांकि , इस बार शाओमी ने फोन में स्पीकर ग्रिल... | डिज़ाइन | हालांकि बार शाओमी फोन स्पीकर ग्रिल निचले हिस्स... | [हालांकि, बार, शाओमी, फोन, स्पीकर, ग्रिल, निचल... | [(हालांकि, JJ), (बार, NNP), (शाओमी, NNP), (फोन... | [] |
| 2 | निचले हिस्से में चार्जिंग और डेटा ट्रांसफर के ... | डिज़ाइन | निचले हिस्से चार्जिंग डेटा ट्रांसफर यूएसबी पोर्ट | [निचले, हिस्से, चार्जिंग, डेटा, ट्रांसफर, यूएस... | [(निचले, JJ), (हिस्से, NNP), (चार्जिंग, NNP), ... | [] |
| 3 | टॉप में आपको 3.5 एमएम ऑडियो जैक के साथ इंफ्रार... | स्पेसिफिकेशन | टॉप आपको 3.5 एमएम ऑडियो जैक इंफ्रारेड एमिटर मि... | [टॉप, आपको, 3.5, एमएम, ऑडियो, जैक, इंफ्रारेड, ... | [(टॉप, JJ), (आपको, $), (3.5, CD), (एमएम, NNP),... | [] |

```
combined_tokens = ' '.join([' '.join(tokens) for tokens in df['tokens']])

print(combined_tokens)
```

```
रियर कैमरा डुअल टोन एलईडी फ्लैश फिंगरप्रिंट स्कैनर सेटअप रेडमी नोट 3 वाला हालांकि बार शाओमी फोन स्पीकर ग्रिल निचले हिस्से जगह दी निचले हिस्से चार्जिंग डेटा ट्रांसफर
```

```
sentence = 'निचले हिस्से जगह दी निचले हिस्से चार्जिंग डेटा ट्रांसफर'



predicted_labels = get_predictions(sentence=sentence,
                                   tokenizer=tokenizer,
                                   model=model
                                   )

for index in range(len(sentence.split(' '))):
  print( sentence.split(' ')[index] + '\t' + predicted_labels[index] )
```

```
निचले	O
हिस्से	O
जगह	O
दी	O
निचले	O
हिस्से	O
चार्जिंग	O
डेटा	O
ट्रांसफर	O
```

## LLMs

```
import numpy as np
import pandas as pd
import nltk


!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) (2023.6.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk) (4.66.1)
```

```
data=pd.read_csv("/content/final.csv")
```

```
data.head(15)
```

| | Unnamed: 0 | text | aspect | preprocessed_text | tokens |
|---|---|---|---|---|---|
| **0** | 0 | रियर कैमरा , डुअल टोन एलईडी फ्लैश और फिंगरप्रि... | स्पेसिफिकेशन | रियर कैमरा डुअल टोन एलईडी फ्लैश फिंगरप्रिंट स्... | ['रियर', 'कैमरा', 'डुअल', 'टोन', 'एलईडी', 'फ्ल... |
| **1** | 1 | हालांकि , इस बार शाओमी ने फोन में स्पीकर ग्रिल... | डिज़ाइन | हालांकि बार शाओमी फोन स्पीकर ग्रिल निचले हिस्स... | ['हालांकि', 'बार', 'शाओमी', 'फोन', 'स्पीकर', '... |
| **2** | 2 | निचले हिस्से में चार्जिंग और डेटा ट्रांसफर के ... | डिज़ाइन | निचले हिस्से चार्जिंग डेटा ट्रांसफर यूएसबी पोर्ट | ['निचले', 'हिस्से', 'चार्जिंग', 'डेटा', 'ट्रां... |
| **3** | 3 | टॉप में आपको 3.5 एमएम ऑडियो जैक के साथ इंफ्रार... | स्पेसिफिकेशन | टॉप आपको 3.5 एमएम ऑडियो जैक इंफ्रारेड एमिटर मि... | ['टॉप', 'आपको', '3.5', 'एमएम', 'ऑडियो', 'जैक',... |
| **4** | 4 | पावर और वॉल्यूम बटन दायीं तरफ हैं और इन तक ऊंग... | डिज़ाइन | पावर वॉल्यूम बटन दायीं तरफ ऊंगलियों पहुंचना आसान | ['पावर', 'वॉल्यूम', 'बटन', 'दायीं', 'तरफ', 'ऊं... |
| **5** | 5 | 8.3 मिलीमीटर मोटाई वाला रेडमी नोट 4 पुराने वेर... | डिज़ाइन | 8.3 मिलीमीटर मोटाई वाला रेडमी नोट 4 पुराने वेर... | ['8.3', 'मिलीमीटर', 'मोटाई', 'वाला', 'रेडमी', ... |
| **6** | 6 | लेकिन वज़न 1 ग्राम ज़्यादा है | डिज़ाइन | वज़न 1 ग्राम ज़्यादा | ['वज़न', '1', 'ग्राम', 'ज़्यादा'] |
| | | हमें रेडमी नोट 4 को | | | |

## BERT Model

```
import pandas as pd
from transformers import BertTokenizer, BertModel
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report
import torch
```

```
tokenizer = BertTokenizer.from_pretrained('bert-base-multilingual-cased')
model = BertModel.from_pretrained('bert-base-multilingual-cased')
```

```
def get_bert_embeddings(text):
    tokens = tokenizer(text, return_tensors='pt', truncation=True, padding=True)
    outputs = model(**tokens)
    embeddings = outputs.pooler_output
    return embeddings

data['bert_embeddings'] = data['preprocessed_text'].apply(get_bert_embeddings)

data['bert_embeddings'] = data['bert_embeddings'].apply(lambda x: x.flatten().detach().numpy())
```

```
tokenizer_config.json: 100%  29.0/29.0 [00:00<00:00, 507B/s]
vocab.txt: 100%  996k/996k [00:00<00:00, 8.47MB/s]
tokenizer.json: 100%  1.96M/1.96M [00:00<00:00, 30.6MB/s]
config.json: 100%  625/625 [00:00<00:00, 8.71kB/s]
model.safetensors: 100%  714M/714M [00:05<00:00, 134MB/s]
```

```
data = data.dropna()
X_train, X_test, y_train, y_test = train_test_split(data['bert_embeddings'].tolist(), data['aspect'], test_size=0.2, random_state=42)

model_lr = LogisticRegression()
model_lr.fit(X_train, y_train)

y_pred = model_lr.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")


print("Classification Report:\n", classification_report(y_test, y_pred))
```

```
Accuracy: 0.5
Classification Report:
                precision    recall  f1-score   support

          NULL       0.40      0.50      0.44         4
        कैमरा        1.00      0.25      0.40         4
      डिज़ाइन        0.00      0.00      0.00         5
    परफॉर्मेंस       0.50      0.75      0.60        12
  स्पेसिफिकेशन       0.50      0.53      0.52        15

      accuracy                           0.50        40
     macro avg       0.48      0.41      0.39        40
  weighted avg       0.48      0.50      0.46        40

/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-d
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-d
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-d
  _warn_prf(average, modifier, msg_start, len(result))
```

## DistilBERT

```
import pandas as pd
from transformers import DistilBertTokenizer, DistilBertModel
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report
import torch
```

```
distilbert_tokenizer = DistilBertTokenizer.from_pretrained('distilbert-base-multilingual-cased')
distilbert_model = DistilBertModel.from_pretrained('distilbert-base-multilingual-cased')

def get_distilbert_embeddings(text):
    tokens = distilbert_tokenizer(text, return_tensors='pt', truncation=True, padding=True)
    outputs = distilbert_model(**tokens)
    embeddings = outputs.last_hidden_state.mean(dim=1)
    return embeddings

data['distilbert_embeddings'] = data['preprocessed_text'].apply(get_distilbert_embeddings)

data['distilbert_embeddings'] = data['distilbert_embeddings'].apply(lambda x: x.flatten().detach().numpy())
```

```
tokenizer_config.json: 100%  29.0/29.0 [00:00<00:00, 1.10kB/s]
vocab.txt: 100%  996k/996k [00:00<00:00, 9.93MB/s]
tokenizer.json: 100%  1.96M/1.96M [00:00<00:00, 25.9MB/s]
config.json: 100%  466/466 [00:00<00:00, 19.3kB/s]
```

```
X_train_distilbert, X_test_distilbert, y_train_distilbert, y_test_distilbert = train_test_split(data['distilbert_embeddings'].tolist(), data

model_lr_distilbert = LogisticRegression()
model_lr_distilbert.fit(X_train_distilbert, y_train_distilbert)

y_pred_distilbert = model_lr_distilbert.predict(X_test_distilbert)

accuracy_distilbert = accuracy_score(y_test_distilbert, y_pred_distilbert)
print("DistilBERT Model Accuracy:", accuracy_distilbert)
print("Classification Report for DistilBERT:\n", classification_report(y_test_distilbert, y_pred_distilbert))
```

```
DistilBERT Model Accuracy: 0.475
Classification Report for DistilBERT:
                precision    recall  f1-score   support

          NULL       0.20      0.25      0.22         4
        कैमरा        0.00      0.00      0.00         4
      डिज़ाइन        0.00      0.00      0.00         5
    परफॉर्मेंस       0.50      0.50      0.50        12
  स्पेसिफिकेशन       0.63      0.80      0.71        15

     accuracy                           0.48        40
    macro avg       0.27      0.31      0.29        40
 weighted avg       0.41      0.47      0.44        40

/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-d
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-d
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-d
  _warn_prf(average, modifier, msg_start, len(result))
```

## RoBERTa

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, f1_score
from sklearn.preprocessing import LabelEncoder
from transformers import RobertaTokenizer, RobertaForSequenceClassification, AdamW
from torch.utils.data import DataLoader, TensorDataset, random_split
import torch
from tqdm import tqdm

data=pd.read_csv("/content/final.csv")

X, y = data["preprocessed_text"], data["aspect"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

label_encoder = LabelEncoder()
y_train_encoded = label_encoder.fit_transform(y_train)
y_test_encoded = label_encoder.transform(y_test)

tokenizer = RobertaTokenizer.from_pretrained('roberta-base')
model = RobertaForSequenceClassification.from_pretrained('roberta-base', num_labels=len(label_encoder.classes_))

X_train_tokens = tokenizer(X_train.tolist(), padding=True, truncation=True, return_tensors='pt')
X_test_tokens = tokenizer(X_test.tolist(), padding=True, truncation=True, return_tensors='pt')
```