NLP EXPERIMENT NO : 06

Aim - Study the Different POS Taggers & Perform POS Tagging on the Given Text.

Pareenita Atul Shirsath BE AI & DS Roll.No:57

```
!pip install stanza
```

```
Collecting stanza
  Downloading stanza-1.10.1-py3-none-any.whl.metadata (13 kB)
Collecting emoji (from stanza)
  Downloading emoji-2.14.1-py3-none-any.whl.metadata (5.7 kB)
Requirement already satisfied: numpy in /usr/local/lib/python3.12/dist-packages (from stanza) (2.0.2)
Requirement already satisfied: protobuf>=3.15.0 in /usr/local/lib/python3.12/dist-packages (from stanza) (5.29.5)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from stanza) (2.32.4)
Requirement already satisfied: networkx in /usr/local/lib/python3.12/dist-packages (from stanza) (3.5)
Requirement already satisfied: torch>=1.3.0 in /usr/local/lib/python3.12/dist-packages (from stanza) (2.8.0+cu126)
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from stanza) (4.67.1)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from torch>=1.3.0->stanza) (3.19.1)
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.12/dist-packages (from torch>=1.3.0->stanza) (4.1
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from torch>=1.3.0->stanza) (75.2.0)
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-packages (from torch>=1.3.0->stanza) (1.13.3)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from torch>=1.3.0->stanza) (3.1.6)
Requirement already satisfied: fsspec in /usr/local/lib/python3.12/dist-packages (from torch>=1.3.0->stanza) (2025.3.0)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=1.3.0->stanza
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=1.3.0->star
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in /usr/local/lib/python3.12/dist-packages (from torch>=1.3.0->stanza
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in /usr/local/lib/python3.12/dist-packages (from torch>=1.3.0->stanza)
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in /usr/local/lib/python3.12/dist-packages (from torch>=1.3.0->stanza)
Requirement already satisfied: nvidia-cufft-cu12==11.3.0.4 in /usr/local/lib/python3.12/dist-packages (from torch>=1.3.0->stanza) (1
Requirement already satisfied: nvidia-curand-cu12==10.3.7.77 in /usr/local/lib/python3.12/dist-packages (from torch>=1.3.0->stanza)
Requirement already satisfied: nvidia-cusolver-cu12==11.7.1.2 in /usr/local/lib/python3.12/dist-packages (from torch>=1.3.0->stanza
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in /usr/local/lib/python3.12/dist-packages (from torch>=1.3.0->stanza
Requirement already satisfied: nvidia-cusparselt-cu12==0.7.1 in /usr/local/lib/python3.12/dist-packages (from torch>=1.3.0->stanza)
Requirement already satisfied: nvidia-nccl-cu12==2.27.3 in /usr/local/lib/python3.12/dist-packages (from torch>=1.3.0->stanza) (2.27
Requirement already satisfied: nvidia-nvtx-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=1.3.0->stanza) (12
Requirement already satisfied: nvidia-nvjitlink-cu12==12.6.85 in /usr/local/lib/python3.12/dist-packages (from torch>=1.3.0->stanza
Requirement already satisfied: nvidia-cufile-cu12==1.11.1.6 in /usr/local/lib/python3.12/dist-packages (from torch>=1.3.0->stanza)
Requirement already satisfied: triton==3.4.0 in /usr/local/lib/python3.12/dist-packages (from torch>=1.3.0->stanza) (3.4.0)
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests->stanza) (3.4.3)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests->stanza) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests->stanza) (2.5.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests->stanza) (2025.8.3)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.12/dist-packages (from sympy>=1.13.3->torch>=1.3.0->star
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->torch>=1.3.0->stanza) (3.0.2
Downloading stanza-1.10.1-py3-none-any.whl (1.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.1/1.1 MB 23.2 MB/s eta 0:00:00
Downloading emoji-2.14.1-py3-none-any.whl (590 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 590.6/590.6 kB 36.7 MB/s eta 0:00:00
Installing collected packages: emoji, stanza
Successfully installed emoji-2.14.1 stanza-1.10.1
```

```
# pos_taggers_demo.py

import nltk
import spacy

# Download required NLTK data
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
nltk.download('punkt_tab') # Download the missing resource
nltk.download('averaged_perceptron_tagger_eng') # Download the missing resource

# Load spaCy models
# English model
try:
    nlp_en = spacy.load("en_core_web_sm")
except:
    !python -m spacy download en_core_web_sm
    nlp_en = spacy.load("en_core_web_sm")


# Hindi / Marathi / Sanskrit → Use multilingual spaCy model
# Install via: pip install spacy-langdetect stanza
import stanza
stanza.download('hi')   # Hindi
stanza.download('mr')   # Marathi
stanza.download('sa')  # Sanskrit

nlp_hi = stanza.Pipeline('hi')  # Hindi
nlp_mr = stanza.Pipeline('mr')  # Marathi
nlp_sa = stanza.Pipeline('sa')  # Sanskrit

…
```

```
# ------------------------------
# Example Sentences
# ------------------------------
sentences = {
    "English": "Paree is wearing a beautiful saree and standing near the decorated flowers.",
    "Hindi": "परी एक सुंदर साड़ी पहन रही है और फूलों के पास खड़ी है।",
    "Marathi": "परी एक सुंदर साडी घालत आहे आणि फुलांच्या जवळ उभी आहे.",
    "Sanskrit": "परी सुन्दरीं साड़ीं धारयति पुष्पानां समीपे तिष्ठति।"
}

# ------------------------------
# Rule-Based Tagger (NLTK regex)
# ------------------------------
from nltk.tag import RegexpTagger

patterns = [
    (r'.*ing$', 'VBG'),              # gerunds
    (r'.*ed$', 'VBD'),               # past tense verbs
    (r'.*es$', 'VBZ'),               # 3rd person singular verbs
    (r'^-?[0-9]+$', 'CD'),           # cardinal numbers
    (r'.*', 'NN')                    # default noun
]
rule_based_tagger = RegexpTagger(patterns)

print("\n=== Rule-Based Tagger (English only) ===")
tokens = nltk.word_tokenize(sentences["English"])
print(rule_based_tagger.tag(tokens))

# ------------------------------
# Statistical Tagger (NLTK)
# ------------------------------
print("\n=== Statistical Tagger (NLTK Averaged Perceptron) ===")
print(nltk.pos_tag(tokens))

# ------------------------------
# Hybrid Tagger (Rule + Statistical)
# ------------------------------
from nltk.tag import UnigramTagger

# Train unigram tagger on first 100 sentences of Brown corpus
nltk.download('brown')
from nltk.corpus import brown
train_sents = brown.tagged_sents(categories='news')[:100]
unigram_tagger = UnigramTagger(train_sents, backoff=rule_based_tagger)

print("\n=== Hybrid Tagger (Unigram + Rule-Based) ===")
print(unigram_tagger.tag(tokens))

# ------------------------------
# Neural Tagger (spaCy + Stanza)
# ------------------------------
print("\n=== Neural Tagger with spaCy (English) ===")
doc = nlp_en(sentences["English"])
for token in doc:
    print(token.text, "→", token.pos_, "(", token.tag_, ")")

print("\n=== Neural Tagger with Stanza (Hindi) ===")
doc_hi = nlp_hi(sentences["Hindi"])
for sent in doc_hi.sentences:
    for word in sent.words:
        print(word.text, "→", word.upos)

print("\n=== Neural Tagger with Stanza (Marathi) ===")
doc_mr = nlp_mr(sentences["Marathi"])
for sent in doc_mr.sentences:
    for word in sent.words:
        print(word.text, "→", word.upos)

print("\n=== Neural Tagger with Stanza (Sanskrit) ===")
doc_sa = nlp_sa(sentences["Sanskrit"])
for sent in doc_sa.sentences:
    for word in sent.words:
        print(word.text, "→", word.upos)
```

```
[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 punkt_tab to /root/nltk_data...
[nltk_data]   Package punkt_tab 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!
```

Downloading https://raw.githubusercontent.com/stanfordnlp/stanza- 434k/? [00:00<00:00, 15.6MB/s]

resources/main/resources_1.10.0.json:

```
INFO:stanza:Downloaded file to /root/stanza_resources/resources.json
INFO:stanza:Downloading default packages for language: hi (Hindi) ...
INFO:stanza:File exists: /root/stanza_resources/hi/default.zip
INFO:stanza:Finished downloading models and saved to /root/stanza_resources
```

Downloading https://raw.githubusercontent.com/stanfordnlp/stanza- 434k/? [00:00<00:00, 26.4MB/s]

resources/main/resources_1.10.0.json:

```
INFO:stanza:Downloaded file to /root/stanza_resources/resources.json
INFO:stanza:Downloading default packages for language: mr (Marathi) ...
INFO:stanza:File exists: /root/stanza_resources/mr/default.zip
INFO:stanza:Finished downloading models and saved to /root/stanza_resources
```

Downloading https://raw.githubusercontent.com/stanfordnlp/stanza- 434k/? [00:00<00:00, 21.3MB/s]

resources/main/resources_1.10.0.json:

```
INFO:stanza:Downloaded file to /root/stanza_resources/resources.json
INFO:stanza:Downloading default packages for language: sa (Sanskrit) ...
INFO:stanza:File exists: /root/stanza_resources/sa/default.zip
INFO:stanza:Finished downloading models and saved to /root/stanza_resources
INFO:stanza:Checking for updates to resources.json in case models have been updated.  Note: this behavior can be turned off with d
```

Downloading https://raw.githubusercontent.com/stanfordnlp/stanza- 434k/? [00:00<00:00, 28.8MB/s]

resources/main/resources_1.10.0.json:

```
INFO:stanza:Downloaded file to /root/stanza_resources/resources.json
INFO:stanza:Loading these models for language: hi (Hindi):
============================
| Processor | Package      |
----------------------------
| tokenize  | hdtb         |
| pos       | hdtb_charlm  |
| lemma     | hdtb_nocharlm |
| depparse  | hdtb_charlm  |
| ner       | ilner_charlm |
============================

INFO:stanza:Using device: cpu
INFO:stanza:Loading: tokenize
INFO:stanza:Loading: pos
INFO:stanza:Loading: lemma
INFO:stanza:Loading: depparse
INFO:stanza:Loading: ner
INFO:stanza:Done loading processors!
INFO:stanza:Checking for updates to resources.json in case models have been updated.  Note: this behavior can be turned off with d
```

Downloading https://raw.githubusercontent.com/stanfordnlp/stanza- 434k/? [00:00<00:00, 22.3MB/s]

resources/main/resources_1.10.0.json:

```
INFO:stanza:Downloaded file to /root/stanza_resources/resources.json
INFO:stanza:Loading these models for language: mr (Marathi):
============================
| Processor | Package      |
----------------------------
| tokenize  | ufal         |
| mwt       | ufal         |
| pos       | ufal_charlm  |
| lemma     | ufal_nocharlm |
| depparse  | ufal_charlm  |
| sentiment | l3cube_charlm |
| ner       | l3cube       |
============================

INFO:stanza:Using device: cpu
INFO:stanza:Loading: tokenize
INFO:stanza:Loading: mwt
INFO:stanza:Loading: pos
INFO:stanza:Loading: lemma
INFO:stanza:Loading: depparse
INFO:stanza:Loading: sentiment
INFO:stanza:Loading: ner
INFO:stanza:Done loading processors!
INFO:stanza:Checking for updates to resources.json in case models have been updated.  Note: this behavior can be turned off with d
```

Downloading https://raw.githubusercontent.com/stanfordnlp/stanza- 434k/? [00:00<00:00, 18.4MB/s]

resources/main/resources_1.10.0.json:

```
INFO:stanza:Downloaded file to /root/stanza_resources/resources.json
INFO:stanza:Loading these models for language: sa (Sanskrit):
==============================
| Processor | Package        |
------------------------------
| tokenize  | vedic          |
| pos       | vedic_nocharlm |
| lemma     | vedic_nocharlm |
| depparse  | vedic_nocharlm |
==============================

INFO:stanza:Using device: cpu
INFO:stanza:Loading: tokenize
INFO:stanza:Loading: pos
INFO:stanza:Loading: lemma
INFO:stanza:Loading: depparse
INFO:stanza:Done loading processors!
[nltk_data] Downloading package brown to /root/nltk_data...
[nltk_data]   Package brown is already up-to-date!

=== Rule-Based Tagger (English only) ===
[('Paree', 'NN'), ('is', 'NN'), ('wearing', 'VBG'), ('a', 'NN'), ('beautiful', 'NN'), ('saree', 'NN'), ('and', 'NN'), ('standing',

=== Statistical Tagger (NLTK Averaged Perceptron) ===
[('Paree', 'NNP'), ('is', 'VBZ'), ('wearing', 'VBG'), ('a', 'DT'), ('beautiful', 'JJ'), ('saree', 'NN'), ('and', 'CC'), ('standing

=== Hybrid Tagger (Unigram + Rule-Based) ===
[('Paree', 'NN'), ('is', 'BEZ'), ('wearing', 'VBG'), ('a', 'AT'), ('beautiful', 'NN'), ('saree', 'NN'), ('and', 'CC'), ('standing'

=== Neural Tagger with spaCy (English) ===
Paree → PROPN ( NNP )
is → AUX ( VBZ )
wearing → VERB ( VBG )
a → DET ( DT )
beautiful → ADJ ( JJ )
saree → NOUN ( NN )
and → CCONJ ( CC )
standing → VERB ( VBG )
near → ADP ( IN )
the → DET ( DT )
decorated → VERB ( VBN )
flowers → NOUN ( NNS )
. → PUNCT ( . )

=== Neural Tagger with Stanza (Hindi) ===
परी → NOUN
एक → NUM
सुंदर → ADJ
साड़ी → NOUN
पहन → VERB
रही → AUX
है → AUX
और → CCONJ
फूलों → NOUN
के → ADP
पास → ADP
खड़ी → ADJ
है → AUX
। → PUNCT

=== Neural Tagger with Stanza (Marathi) ===
परी → NOUN
एक → DET
सुंदर → ADJ
साडी → NOUN
घालत → VERB
आहे → AUX
आणि → CCONJ
फुला → NOUN
च्या → PART
जवळ → ADP
उभी → ADJ
आहे → AUX
. → PUNCT

=== Neural Tagger with Stanza (Sanskrit) ===
परी → NOUN
सुन्दरीं → ADJ
साड़ीं → NOUN
धारयति → ADJ
पुष्पानां → NOUN
```