import nltk

nltk.download('punkt')

nltk.download('averaged\_perceptron\_tagger')

[nltk\_data] Downloading package punkt to

[nltk\_data] C:\Users\Admin\AppData\Roaming\nltk\_data... [nltk\_data] Unzipping tokenizers\punkt.zip.

[nltk\_data] Downloading package averaged\_perceptron\_tagger to

[nltk\_data] C:\Users\Admin\AppData\Roaming\nltk\_data...

[nltk\_data] Package averaged\_perceptron\_tagger is already up-to- [nltk\_data] date!

True

from nltk.chunk import RegexpParser

from nltk.tokenize import word\_tokenize

sentence = "Educative Answers is a free web encyclopedia written by devs for devs."

# Tokenization

tokens = word\_tokenize(sentence)

tokens

['Educative', 'Answers', 'is',

'a',

'free',

'web',

'encyclopedia', 'written',

'by',

'devs',

'for',

'devs',

'.']

# POS tagging

pos\_tags = nltk.pos\_tag(tokens)

pos\_tags

[('Educative', 'JJ'),

('Answers', 'NNPS'),

('is', 'VBZ'),

('a', 'DT'),

('free', 'JJ'),

('web', 'NN'),

('encyclopedia', 'NN'),

('written', 'VBN'),

('by', 'IN'),

('devs', 'NN'),

('for', 'IN'),

('devs', 'NN'),

('.', '.')]

# Chunking patterns

chunk\_patterns = r"""

NP: {<DT>?<JJ>\*<NN>} # Chunk noun phrases VP: {<VB.\*><NP|PP>} # Chunk verb phrases

"""

chunk\_patterns

'\n NP: {<DT>?<JJ>\*<NN>} # Chunk noun phrases\n VP: {<VB.\*><NP|PP>} # Chunk verb phrases\n'

# Create a chunk parser

chunk\_parser = RegexpParser(chunk\_patterns)

chunk\_parser

<chunk.RegexpParser with 2 stages>

# Perform chunking

result = chunk\_parser.parse(pos\_tags)

print(result)

(S

Educative/JJ Answers/NNPS

(VP is/VBZ (NP a/DT free/JJ web/NN))

(NP encyclopedia/NN) written/VBN

by/IN

(NP devs/NN) for/IN

(NP devs/NN)

./.)