Experiment No.6
Perform POS tagging on the given English and Indian Language Text
Date of Performance:
Date of Submission:

Aim: Perform POS tagging on the given English and Indian Language Text

Objective: To study POS Tagging and tag the part of speech for given input in english and an Indian Language.

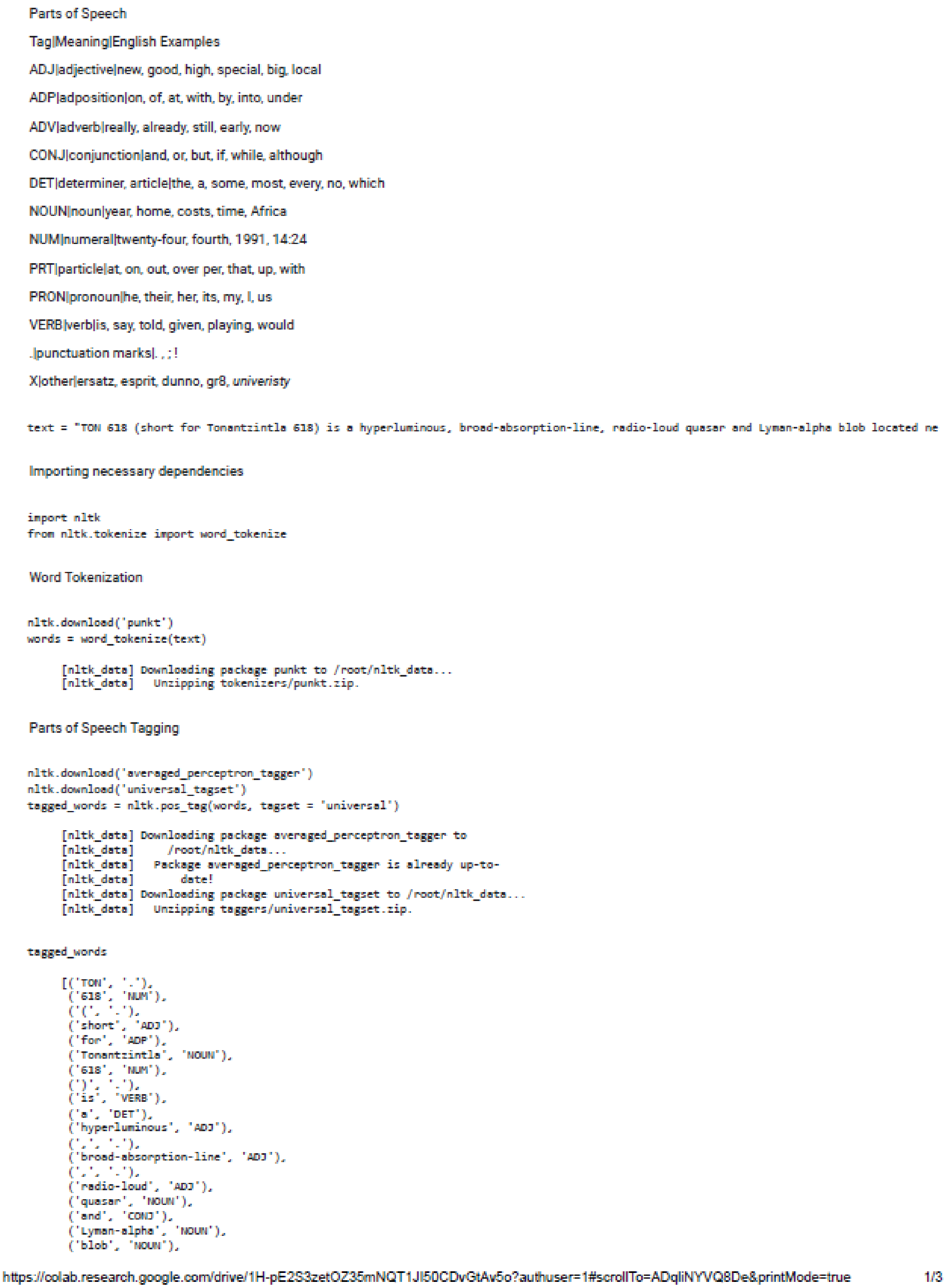
Theory:

The primary target of Part-of-Speech (POS) tagging is to identify the grammatical group of a given word. Whether it is a NOUN, PRONOUN, ADJECTIVE, VERB, ADVERBS, etc. based on the context. POS Tagging looks for relationships within the sentence and assigns a corresponding tag to the word.

POS Tagging (Parts of Speech Tagging) is a process to mark up the words in text format for a particular part of a speech based on its definition and context. It is responsible for text reading in a language and assigning some specific token (Parts of Speech) to each word. It is also called grammatical tagging.

Steps Involved in the POS tagging example:

- Tokenize text (word_tokenize)
- apply pos_tag to above step that is nltk.pos_tag(tokenize_text)



Parts of Speech

Tag	Meaning	English Examples
ADJ	adjective	new, good, high, special, big, local
ADP	adposition	on, of, at, with, by, into, under
ADV	adverb	really, already, still, early, now
CONJ	conjunction	and, or, but, if, while, although
DET	determiner, article	the, a, some, most, every, no, which
NOUN	noun	year, home, costs, time, Africa
NUM	numeral	twenty-four, fourth, 1991, 14:24
PRT	particle	at, on, out, over per, that, up, with
PRON	pronoun	he, their, her, its, my, I, us
VERB	verb	is, say, told, given, playing, would
.	punctuation marks	. , ; !
X	other	ersatz, esprit, dunno, gr8, *univeristy*

```
text = "TON 618 (short for Tonantzintla 618) is a hyperluminous, broad-absorption-line, radio-loud quasar and Lyman-alpha blob located ne
```

Importing necessary dependencies

```
import nltk
from nltk.tokenize import word_tokenize
```

Word Tokenization

```
nltk.download('punkt')
words = word_tokenize(text)

    [nltk_data] Downloading package punkt to /root/nltk_data...
    [nltk_data]   Unzipping tokenizers/punkt.zip.
```

Parts of Speech Tagging

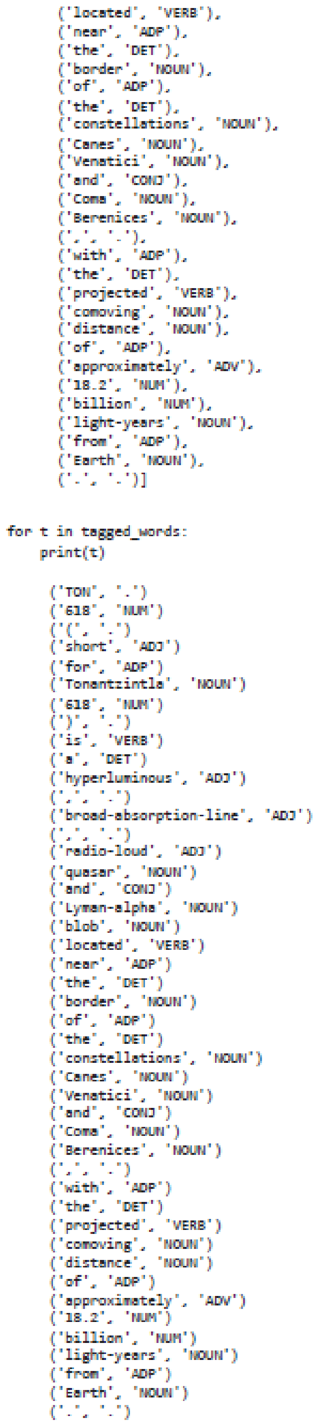
```
nltk.download('averaged_perceptron_tagger')
nltk.download('universal_tagset')
tagged_words = nltk.pos_tag(words, tagset = 'universal')

    [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 universal_tagset to /root/nltk_data...
    [nltk_data]   Unzipping taggers/universal_tagset.zip.
```

```
tagged_words

    [('TON', '.'),
     ('618', 'NUM'),
     ('(', '.'),
     ('short', 'ADJ'),
     ('for', 'ADP'),
     ('Tonantzintla', 'NOUN'),
     ('618', 'NUM'),
     (')', '.'),
     ('is', 'VERB'),
     ('a', 'DET'),
     ('hyperluminous', 'ADJ'),
     (',', '.'),
     ('broad-absorption-line', 'ADJ'),
     (',', '.'),
     ('radio-loud', 'ADJ'),
     ('quasar', 'NOUN'),
     ('and', 'CONJ'),
     ('Lyman-alpha', 'NOUN'),
     ('blob', 'NOUN'),
```

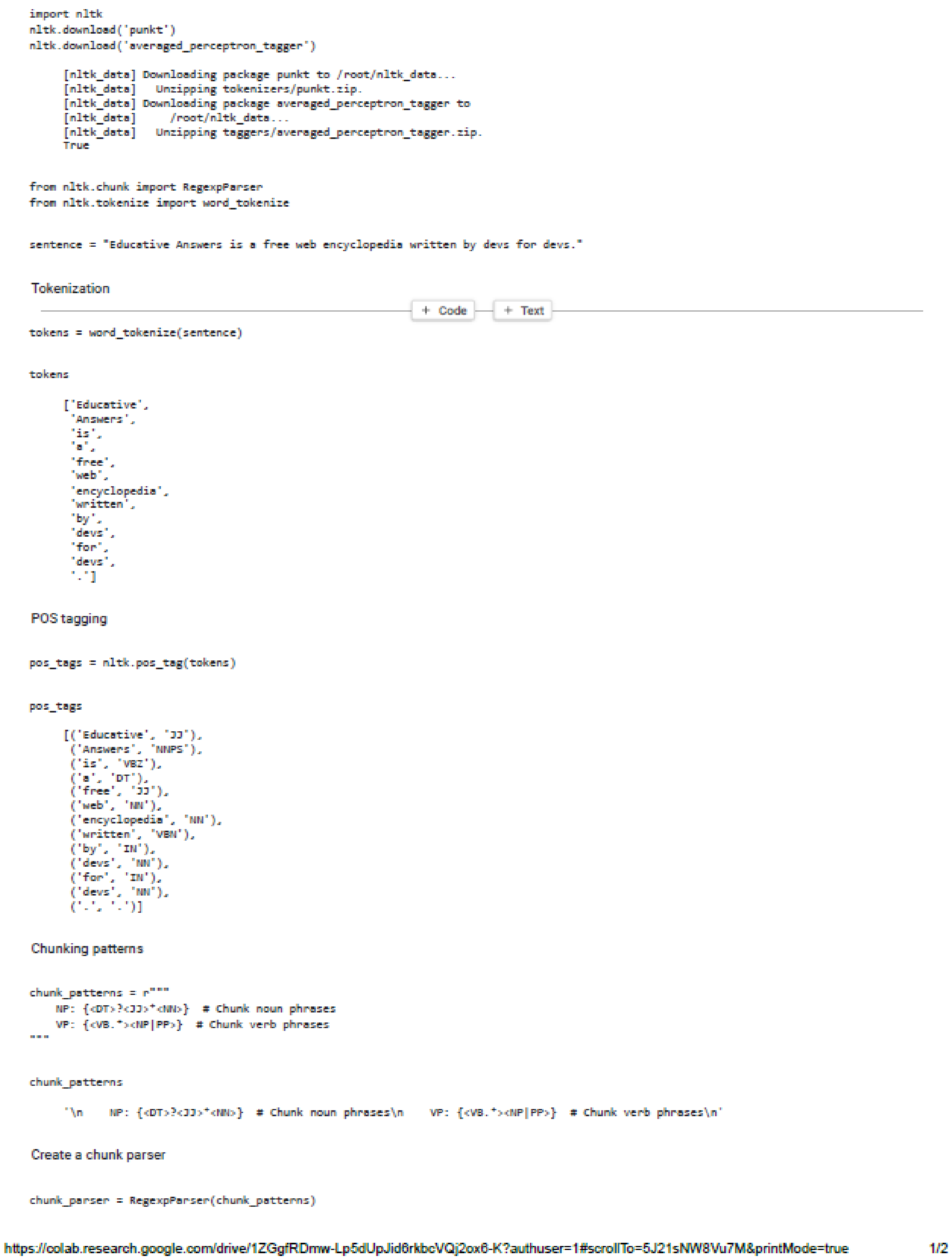
https://colab.research.google.com/drive/1H-pE2S3zetOZ35mNQT1JI50CDvGtAv5o?authuser=1#scrollTo=ADqliNYVQ8De&printMode=true 1/3



```
      ('located', 'VERB'),
      ('near', 'ADP'),
      ('the', 'DET'),
      ('border', 'NOUN'),
      ('of', 'ADP'),
      ('the', 'DET'),
      ('constellations', 'NOUN'),
      ('Canes', 'NOUN'),
      ('Venatici', 'NOUN'),
      ('and', 'CONJ'),
      ('Coma', 'NOUN'),
      ('Berenices', 'NOUN'),
      (',', '.'),
      ('with', 'ADP'),
      ('the', 'DET'),
      ('projected', 'VERB'),
      ('comoving', 'NOUN'),
      ('distance', 'NOUN'),
      ('of', 'ADP'),
      ('approximately', 'ADV'),
      ('18.2', 'NUM'),
      ('billion', 'NUM'),
      ('light-years', 'NOUN'),
      ('from', 'ADP'),
      ('Earth', 'NOUN'),
      ('.', '.')]


for t in tagged_words:
    print(t)

     ('TON', '.')
     ('618', 'NUM')
     ('(', '.')
     ('short', 'ADJ')
     ('for', 'ADP')
     ('Tonantzintla', 'NOUN')
     ('618', 'NUM')
     (')', '.')
     ('is', 'VERB')
     ('a', 'DET')
     ('hyperluminous', 'ADJ')
     (',', '.')
     ('broad-absorption-line', 'ADJ')
     (',', '.')
     ('radio-loud', 'ADJ')
     ('quasar', 'NOUN')
     ('and', 'CONJ')
     ('Lyman-alpha', 'NOUN')
     ('blob', 'NOUN')
     ('located', 'VERB')
     ('near', 'ADP')
     ('the', 'DET')
     ('border', 'NOUN')
     ('of', 'ADP')
     ('the', 'DET')
     ('constellations', 'NOUN')
     ('Canes', 'NOUN')
     ('Venatici', 'NOUN')
     ('and', 'CONJ')
     ('Coma', 'NOUN')
     ('Berenices', 'NOUN')
     (',', '.')
     ('with', 'ADP')
     ('the', 'DET')
     ('projected', 'VERB')
     ('comoving', 'NOUN')
     ('distance', 'NOUN')
     ('of', 'ADP')
     ('approximately', 'ADV')
     ('18.2', 'NUM')
     ('billion', 'NUM')
     ('light-years', 'NOUN')
     ('from', 'ADP')
     ('Earth', 'NOUN')
     ('.', '.')
```



```
import nltk
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
```

```
    [nltk_data] Downloading package punkt to /root/nltk_data...
    [nltk_data]   Unzipping tokenizers/punkt.zip.
    [nltk_data] Downloading package averaged_perceptron_tagger to
    [nltk_data]     /root/nltk_data...
    [nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
    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
"""
```

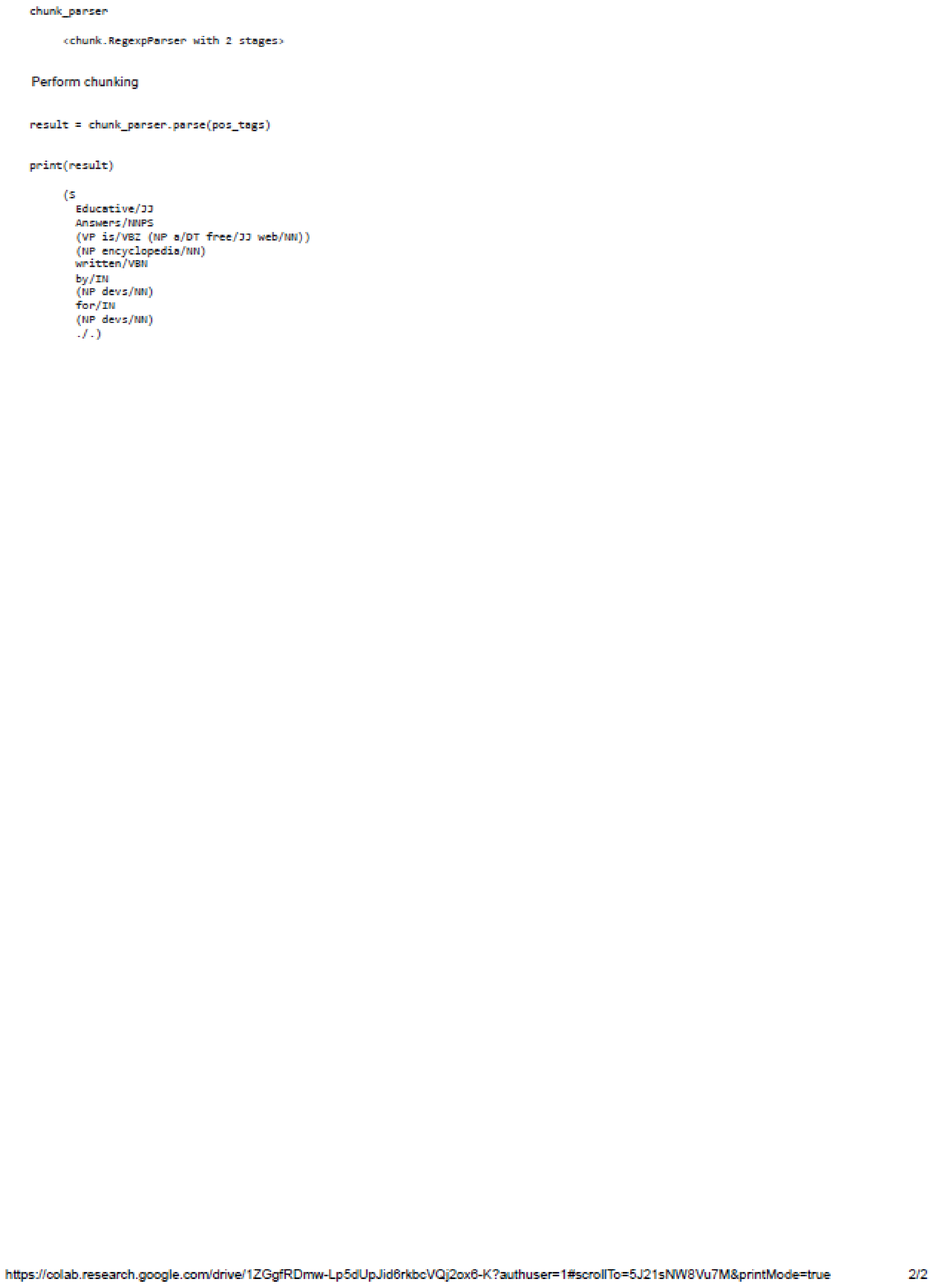
```
chunk_patterns
```

```
    '\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)
```

https://colab.research.google.com/drive/1ZGgfRDmw-Lp5dUpJid6rkbcVQj2ox6-K?authuser=1#scrollTo=5J21sNW8Vu7M&printMode=true 1/2



```
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)
      ./.)
```

https://colab.research.google.com/drive/1ZGgfRDmw-Lp5dUpJid6rkbcVQj2ox6-K?authuser=1#scrollTo=5J21sNW8Vu7M&printMode=true 2/2

Conclusion:

The results of part-of-speech (POS) tagging can significantly impact natural language processing tasks, and various techniques have been developed to tackle this problem. Rule-based methods use predefined patterns and heuristics, providing transparency but often lacking accuracy. Statistical methods, such as Hidden Markov Models and Conditional Random Fields, utilize probabilistic models to achieve better accuracy. Machine learning approaches, including deep learning models like recurrent neural networks (RNNs) and transformers, have recently demonstrated state-of-the-art performance by learning contextual information. In conclusion, the choice of POS tagging technique should consider the trade-off between accuracy and complexity, with machine learning methods offering the most promising results in recent years, especially when trained on large datasets and taking advantage of contextual information.