FOR EDUCATIONAL USE

Sundaram

churk Types

Category part

	Churk Type	Tag Name	, <u></u>		
0	Nous	NP			
0	verb	VP			
3	Adverb	ADUP			
(3)	Adjectival	ADJP	5, 1 ^k "		
5	prepoutional	PP	2	- 9 <i>-</i>	2.
\sim					

Tokens Pos Churk -Tags:

1	He	PRB	B-NP
	are	VBD	B-NP
	an	DT	B-NP
1	apple	~~	I-NP
	To	0	B-VP
1	satiate	V B	I-VP
	his	B	B-NP
10.5	hunger	~~	I-NP
+	., 53,19		

(Sundaram)

FOR EDUCATIONAL USE

#	Chunk Types	
	Туре	pefinition
	characters	individual characters within text
	words	words seperated by any amount of white spaces (spaces, tabs) within text
	Lines	Paragraphs seperated by the any of several standard line endings (CR, LF, CRLF, etc.)
	Text items	rainons of text sererated by commas
	Cist items	The individual item in a list
9	Lytes	the bytes within Linary data
	occurences	The text matches of defined pattern
	matches	The text matches and text range of a defined pattern and its capture groups
	O LANCESCO DE LA COMPANSIONA DEL COMPANSIONA DE LA COMPANSIONA DEL COMPANSIONA DE LA	

FOR EDUCATIONAL USE

(Sumtaram)

	Condusión:
	Herce we understood too process of
	Herce, we understood the process of durking and its types and performed the
	chunking on the given input.
fr	
0	
	FOR EDUCATIONAL USE
Sundaram	FOR EDUCATIONAL USE

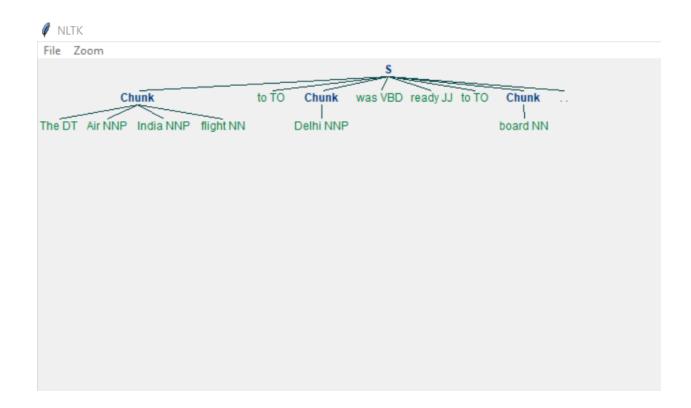
Code

```
# Importing the required libraries
import nltk
from nltk import pos_tag
from nltk import word_tokenize
from nltk import RegexpParser
text = input("Enter Sentence")
# Splitiing the sentence into words
list_of_words = word_tokenize(text)
# Applying POS tagging
tagged_words = pos_tag(list_of_words)
# Extracting the Noun Phrases
chunk to be extracted = r'' Chunk: {<DT>*<NNP>*<NN>*} "
# Applying chunking to the text
chunkParser = nltk.chunk.RegexpParser(chunk_to_be_extracted)
chunked_sentence = chunkParser.parse(tagged_words)
# To view the NLTK tree
chunked_sentence.draw()
# To print the chunks extracted
print('Chunks obtained: \n')
for subtree in chunked_sentence.subtrees():
  if subtree.label() == 'Chunk':
    print(subtree)
```

Output

Giving in the text input

Enter Sentence The Air India flight to Delhi was ready to board.



Jupyter anish-exp7-chunking-nlp Last Checkpoint: 4 minutes ago (autosaved) Edit View Insert Cell Kernel Widgets Help ↑ ↓ | ▶ Run ■ C ▶ | Code **≈** 🖆 🖺 Enter SentenceThe Air India flight to Delhi was ready to board. In [14]: # Extracting the Noun Phrases chunk_to_be_extracted = r''' Chunk: {<DT>*<NNP>*<NN>*} ''' # Applying chunking to the text chunkParser = nltk.chunk.RegexpParser(chunk to be extracted) chunked sentence = chunkParser.parse(tagged words) In [15]: # To view the NLTK tree chunked_sentence.draw() In [16]: # To print the chunks extracted print('Chunks obtained: \n') for subtree in chunked_sentence.subtrees(): if subtree.label() == 'Chunk': print(subtree) Chunks obtained: (Chunk The/DT Air/NNP India/NNP flight/NN) (Chunk Delhi/NNP) (Chunk board/NN) In []: