Page	Vo.		
Date			

Ain- To under stand the concept of chunking and get familior with the basic chunk torget

Theory - Chunk extraction or partial passing is a process of meaning. Full extracting short phrases from the sentence tagged with posts of speech. Chunks are made up of words of the kind of words are defined using the Postage, one can even define a pattern or words that con't be a port of a chunk of such words one known as chunks

Petining chunk patterns.

Chunk potterns are normal regular expression which are modified & designed to match the Postag designed to match sequence of part of speech trap. Angle brackets are used to specify on individual trag for example to match a noun trag

one can define multiple tags in the same way.

Steps to follow:

1) Converting thanks to Reg Ex pattern 2) Passing the sentence using the Regen

For eg. The book has many chapters let us convert the sentence into flat tree

HCOT booken has VBZ many JJ chapter MNS

Teacher's Sign.: _

	Page No. Date						
and the same of th							
	Step 2 - Forming chunes						
Control of the last of the las	Sentence						
the section has reported							
	Chunk						
and the state of t	V V						
	the DT booken has VBZ mong JJ enapter NPS						
	Step 3 - Splitting bigger chunks into smaller chunks						
0	Sentence						
	J						
	chunk has VB2 thunk						
	chunk has VB2 chunk						
	the DOT bookNP mony JJ . Chapter NNS						
	THE ODT BOOK FOR						
	Thus convert the chunktree into 2 subtree						
	5						
	NP has VBZ NP						
	the DT bookper mong JJ chapter mus						
	a. The chunks are						
	(the book, NP), (has, VP) (many chapter, NP)						
	time book, NIT, this , (many onaph)						
	Conduston: Thus we have successfully implemented						
	chunking using basic chunk target						
	Teacher's Sign.:						

Scanned with CamScanner

EXP 7: CHUNKING

PROGRAM:

```
import nltk
txt = "The dog barked at the cat"
grammar = ("
    NP: {<DT>?<JJ>*<NN>} # NP
    "")
chunkParser = nltk.RegexpParser(grammar)
tagged = nltk.pos_tag(nltk.word_tokenize(txt))
print(tagged)
tree = chunkParser.parse(tagged)
for subtree in tree.subtrees():
    print(subtree)
tree.draw()
```

OUTPUT:

```
hunking.py ====
[('The', 'DT'), ('dog', 'NN'), ('barked', 'VBD'), ('at', 'IN'),
('the', 'DT'), ('cat', 'NN')]
(S (NP The/DT dog/NN) barked/VBD at/IN (NP the/DT cat/NN))
(NP The/DT dog/NN)
(NP the/DT cat/NN)
```

