Basic NLP Tasks with NLTK

```
import nltk
nltk.download()
from nltk.book import *
```

→ `len(set(text7))`

→ `list(set(text7))[0:10]`

→ u' ' → utf-encoded

→ Frequency

→ `dist = FreqDist(text7)`
`len(dist)`

→ `vocab1 = dist.keys()`

→ `dist['four']`
20 → count

→ `freqwords = [w for w in vocab1`
`if len(w)>5 and dist[w]>100]`

(length > 5 and count > 100)

Normalisation and Stemming

"List listed lists listing listings"

```
text.lower().split()
```

```
porter = nltk.PorterStemmer()
[porter.stem(t) for t in words]
```

Lemmatization

```
lemma = nltk.WordNetLemmatizer()
[lemma.lemmatize(t) for t in words]
```

Tokenization

```
text1.split(' ')
nltk.word_tokenize(text1)
```

Sentence Splitting

```
nltk.sent_tokenize(text1)
```

Advanced NLP Tasks with NLTK

Counting words, frequency, sentence boundaries,

→ POS Tagging.

```
text2 = nltk.word_tokenize(text1)
nltk.pos_tag(text2)
```

Ambiguity in POS Tagging

Visiting aunts can be a nuisance

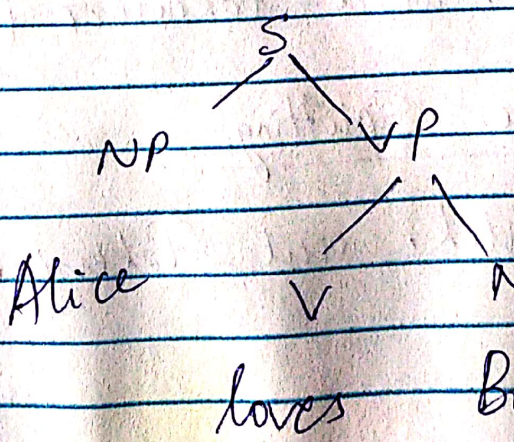
→ Aunts who are visiting

(or)

→ We are visiting aunts

Parsing Sentence Structure

Alice loves Bob.



```
parser = nltk.ChartParser(grammar)
trees = parser.parse_all(text1)
for tree in trees:
    print(tree)
```

Ambiguity in Parsing

I saw the man with a telescope.

→ man holding a telescope
→ saw the man using a telescope

```
parser = nltk.ChartParser(grammar1)
grammar1 = nltk.data.load('mygrammar1.cfg')
```

Tree Bank

```
from nltk.corpus import treebank
text1 = treebank.parsed_sents('wsj_0001.mrg')[0]
print tree structure (parsed)
```

→ Uncommon usages of words

Ex :- the old man the boat

→ Well-formed sentences may still be meaningless

Ex :- Colorless green ideas sleep furiously.