Applied Text Mining in Python

Advanced NLP tasks with NLTK

NLP Tasks

- Counting words, counting frequency of words
- Finding sentence boundaries
- Part of speech tagging
- Parsing the sentence structure
- Identifying semantic role labeling
- Named Entity Recognition
- Co-reference and pronoun resolution

Part-of-speech (POS) Tagging

- **Recall high school grammar: nouns, verbs, adjectives, …**
- **Many more tags or word classes than just these**

Tag	Word class	Tag	Word class	Tag	Word class
CC	Conjunction	JJ	Adjective	PRP	Pronoun
CD	Cardinal	MD	Modal	RB	Adverb
DT	Determiner	NN	Noun	SYM	Symbol
IN	Preposition	POS	Possessive	VB	Verb

…

```
>>> import nltk
>>> nltk.help.upenn_tagset('MD')
MD: modal auxiliary
    can cannot could couldn't dare may might must need ought
    shall should shouldn't will would
```

POS Tagging with NLTK

- **Recall splitting a sentence into words / tokens**

```
>>> text11 = "Children shouldn't drink a sugary drink
before bed."
>>> text13 = nltk.word_tokenize(text11)
```

- **NLTK's Tokenizer**

```
>>> nltk.pos_tag(text13)
[('Children', 'NNP'), ('should', 'MD'), ("n't", 'RB'),
('drink', 'VB'), ('a', 'DT'), ('sugary', 'JJ'), ('drink',
'NN'), ('before', 'IN'), ('bed', 'NN'), ('.', '.')]
```

Ambiguity in POS Tagging

- Ambiguity is common in English

Visiting aunts can be a nuisance

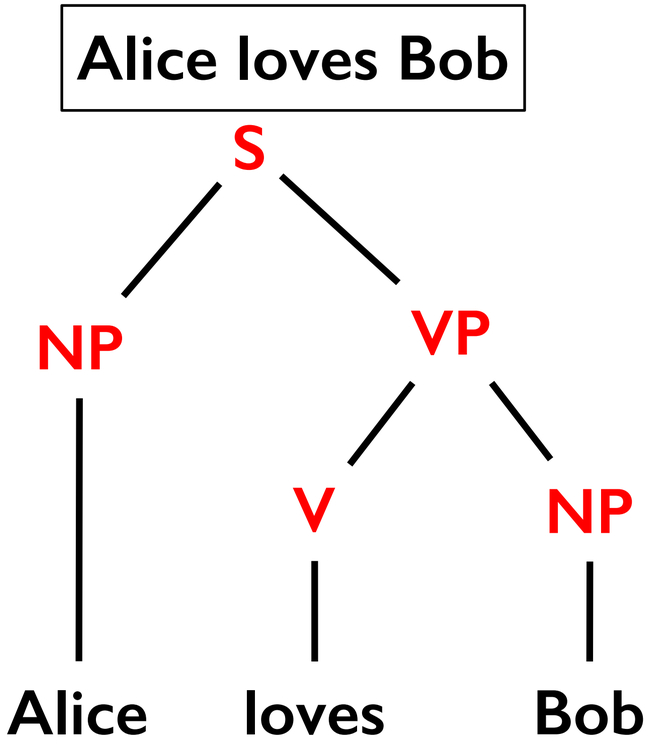
```
>>> text14 = nltk.word_tokenize("Visiting aunts can be a nuisance")
>>> nltk.pos_tag(text14)
[('Visiting', 'VBG'), ('aunts', 'NNS'), ('can', 'MD'), ('be', 'VB'), ('a', 'DT'),
('nuisance', 'NN')]
```

- Another alternative POS tagging

```
[('Visiting', 'JJ'), ('aunts', 'NNS'), ('can', 'MD'), ('be', 'VB'), ('a', 'DT'),
('nuisance', 'NN')]
```

Parsing Sentence Structure

- Making sense of sentences is easy if they follow a well-defined grammatical structure



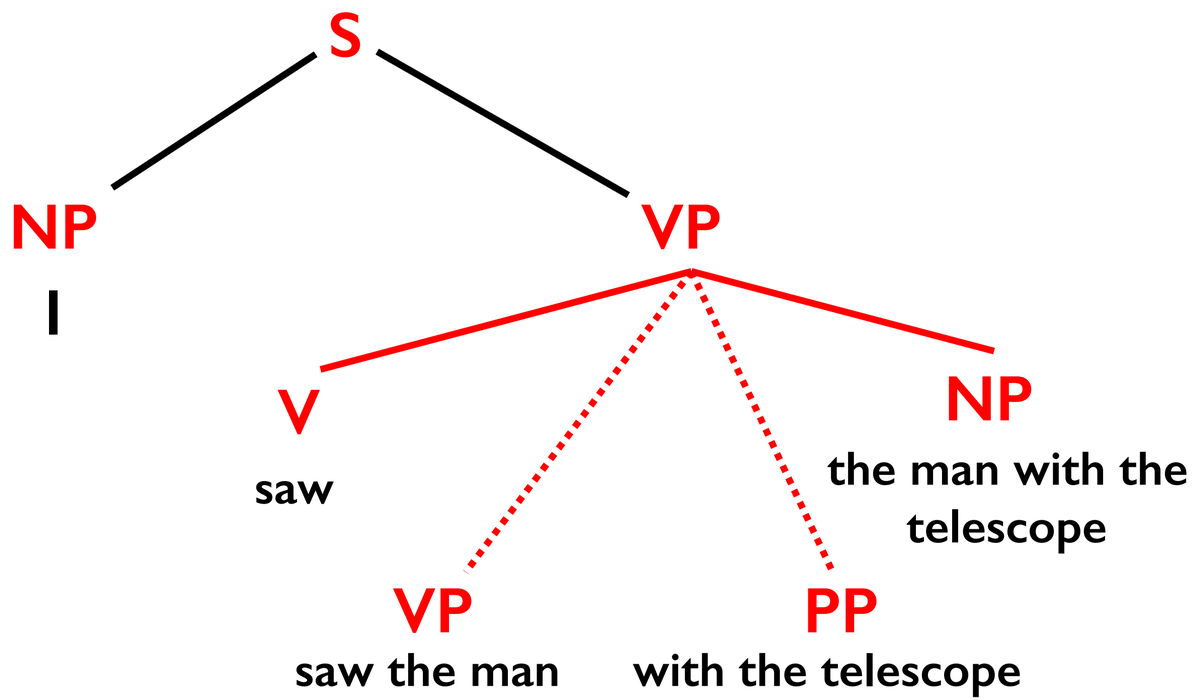

```
>>> text15 = nltk.word_tokenize("Alice loves Bob")
>>> grammar = nltk.CFG.fromstring("""
... S -> NP VP
... VP -> V NP
... NP -> 'Alice' | 'Bob'
... V -> 'loves'
... """)

>>> parser = nltk.ChartParser(grammar)
>>> trees = parser.parse_all(text15)
>>> for tree in trees:
...     print tree
...
(S (NP Alice) (VP (V loves) (NP Bob)))
```

Ambiguity in Parsing

- Ambiguity may exist even if sentences are grammatically correct!

I saw the man with a telescope

```
S -> NP VP
VP -> V NP | VP PP
PP -> P NP
NP -> DT N | DT N PP | 'I'
DT -> 'a' | 'the'
N -> 'man' | 'telescope'
V -> 'saw'
P -> 'with'
```



Ambiguity in Parsing

```
>>> text16 = nltk.word_tokenize("I saw the man with a telescope")
>>> grammar1 = nltk.data.load('mygrammar1.cfg')
>>> grammar1
<Grammar with 13 productions>
>>> parser = nltk.ChartParser(grammar1)
>>> trees = parser.parse_all(text16)
>>> for tree in trees:
...   print tree
...
(S
  (NP I)
  (VP
    (VP (V saw) (NP (DT the) (N man)))
    (PP (P with) (NP (DT a) (N telescope)))))
(S
  (NP I)
  (VP
    (V saw)
    (NP (DT the) (N man) (PP (P with) (NP (DT a) (N telescope))))))
```

```
S -> NP VP
VP -> V NP | VP PP
PP -> P NP
NP -> DT N | DT N PP | 'I'
DT -> 'a' | 'the'
N -> 'man' | 'telescope'
V -> 'saw'
P -> 'with'
```

mygrammar1.cfg

NLTK and Parse Tree Collection

```
>>> from nltk.corpus import treebank
>>> text17 = treebank.parsed_sents('wsj_0001.mrg')[0]
>>> print text17
(S
  (NP-SBJ
    (NP (NNP Pierre) (NNP Vinken))
    (, ,)
    (ADJP (NP (CD 61) (NNS years)) (JJ old))
    (, ,))
  (VP
    (MD will)
    (VP
      (VB join)
      (NP (DT the) (NN board))
      (PP-CLR (IN as) (NP (DT a) (JJ nonexecutive) (NN director)))
      (NP-TMP (NNP Nov.) (CD 29))))
  (. .))
```

POS Tagging & Parsing Complexity

- Uncommon usages of words

The old man the boat

```
>>> text18 = nltk.word_tokenize("The old man the boat")
>>> nltk.pos_tag(text19)
[('The', 'DT'), ('old', 'JJ'), ('man', 'NN'), ('the', 'DT'), ('boat', 'NN')]
```

- Well-formed sentences may still be meaningless!

Colorless green ideas sleep furiously

```
>>> text19 = nltk.word_tokenize("Colorless green ideas sleep furiously")
>>> nltk.pos_tag(text18)
[('Colorless', 'NNP'), ('green', 'JJ'), ('ideas', 'NNS'), ('sleep', 'VBP'),
('furiously', 'RB')]
```

Take Home Concepts

- POS tagging provides insights into the word classes / types in a sentence
- Parsing the grammatical structures helps derive meaning
- Both tasks are difficult, linguistic ambiguity increases the difficulty even more
- Better models could be learned with supervised training
- NLTK provides access to tools and data for training