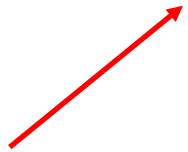
POS

Part-of-speech Specific lexical categories based on their syntactic context and role

POS Tagging A process of classifying and labeling POS tags for words

Consult - https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/Penn-Treebank-Tagset.pdf



4 Problematic cases

This section discusses difficult tagging decisions. Section 4.1 discusses parts of speech that are easily confused and guidelines on how to tag such cases. Section 4.2 contains an alphabetical list of specific problematic words and collocations.

4.1 Confusing parts of speech

This section discusses parts of speech that are easily confused and gives guidelines on how to tag such cases.

CC or DT

When they are the first members of the double conjunctions *both ... and*, *either ... or* and *neither ... nor*, *both*, *either* and *neither* are tagged as coordinating conjunctions (CC), not as determiners (DT).

EXAMPLES: Either/DT child could sing.

But:

Either/CC a boy could sing or/CC a girl could dance.
Either/CC a boy or/CC a girl could sing.
Either/CC a boy or/CC girl could sing.

POS Tagging

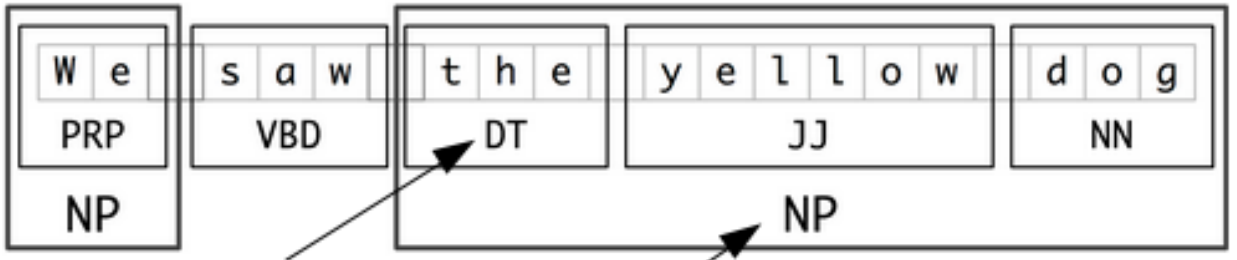
- Retain punctuation and upper cases
- sent_tokenize > word_tokenize

Transposing from vertical to horizontal

```
nltk_pos_tagged = nltk.pos_tag(words[15:40])
pd.DataFrame(nltk_pos_tagged, columns=['Word', 'POS tag']).T
```

	0	1	2	3	4	5	6	7	8	9	...	15	16	17	18	19	20	21	22	23	24
Word	the	Rabbit-Hole	Alice	was	beginning	to	get	very	tired	of	...	the	bank	,	and	of	having	nothing	to	do	:
POS tag	DT	JJ	NNP	VBD	VBG	TO	VB	RB	JJ	IN	...	DT	NN	,	CC	IN	VBG	NN	TO	VB	:

Chunking



POS

Chunk

Step 1

```
grammar = """
NP: {<DT>?<JJ>?<NN.*>}
ADJP: {<JJ>}
ADVP: {<RB.*>}
PP: {<IN>}
VP: {<MD>?<VB.*>+}
"""
```

Can you translate these regexes?

```
from nltk.chunk import RegexpParser
rc = RegexpParser(grammar)
c = rc.parse(nltk_pos_tagged)
print(c)
```

```
(S
  (ADVP once/RB)
  or/CC
  (VP twice/VB)
  she/PRP
  (VP had/VBD peeped/VBN)
  (PP into/IN)
  (NP the/DT book/NN)
  her/PRP$
  (NP sister/NN)
  (VP was/VBD reading/VBG))
```

Step 2

```
from nltk.chunk import RegexpParser
rc = RegexpParser(grammar)
c = rc.parse(nltk_pos_tagged)
```

Step 3

```
c.draw()
```