Lab1-Assignment

This notebook describes the assignment for Lab 1 of the text mining course.

Points: each exercise is prefixed with the number of points you can obtain for the exercise.

We assume you have worked through the following notebooks:

- **Lab1.1-introduction**
- **Lab1.2-introduction-to-NLTK**
- **Lab1.3-introduction-to-spaCy**

In this assignment, you will process an English text (**Lab1-apple-samsung-example.txt**) with both NLTK and spaCy and discuss the similarities and differences.

Credits

The notebooks in this block have been originally created by Marten Postma. Adaptations were made by Filip Ilievski.

Tip: how to read a file from disk

Let's open the file **Lab1-apple-samsung-example.txt** from disk.

In [1]:

```
from pathlib import Path
```

In [2]:

```
cur_dir = Path().resolve() # this should provide you with the folder in which this no
path_to_file = Path.joinpath(cur_dir, 'Lab1-apple-samsung-example.txt')
print(path_to_file)
print('does path exist? ->', Path.exists(path_to_file))
```

```
/Users/owhy/Desktop/github/TextMining-VU-2024/Lab1-apple-samsung-example.txt
does path exist? -> True
```

If the output from the code cell above states that **does path exist? -> False**, please check that the file **Lab1-apple-samsung-example.txt** is in the same directory as this notebook.

In [4]:

```
with open(path_to_file) as infile:
    text = infile.read()

print('number of characters', len(text))
```

```
number of characters 1139
```

[total points: 4] Exercise 1: NLTK

In this exercise, we use NLTK to apply **Part-of-speech (POS) tagging**, **Named Entity Recognition (NER)**, and **Constituency parsing**. The following code snippet already performs sentence splitting and tokenization.

In [5]:

```
import nltk
from nltk.tokenize import sent_tokenize
from nltk import word_tokenize

nltk.download("punkt")
nltk.download("averaged_perceptron_tagger")
```

```
[nltk_data] Downloading package punkt to /Users/owhy/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /Users/owhy/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
```

Out[5]:

```
True
```

In [6]:

```
sentences_nltk = sent_tokenize(text)
```

In [7]:

```
tokens_per_sentence = []
for sentence_nltk in sentences_nltk:
    sent_tokens = word_tokenize(sentence_nltk)
    tokens_per_sentence.append(sent_tokens)
```

We will use lists to keep track of the output of the NLP tasks. We can hence inspect the output for each task using the index of the sentence.

In [8]:

```
sent_id = 1
print('SENTENCE', sentences_nltk[sent_id])
print('TOKENS', tokens_per_sentence[sent_id])
```

```
SENTENCE The six phones and tablets affected are the Galaxy S III, running the new Jel
ly Bean system, the Galaxy Tab 8.9 Wifi tablet, the Galaxy Tab 2 10.1, Galaxy Rugby Pr
o and Galaxy S III mini.
TOKENS ['The', 'six', 'phones', 'and', 'tablets', 'affected', 'are', 'the', 'Galaxy',
'S', 'III', ',', 'running', 'the', 'new', 'Jelly', 'Bean', 'system', ',', 'the', 'Gala
xy', 'Tab', '8.9', 'Wifi', 'tablet', ',', 'the', 'Galaxy', 'Tab', '2', '10.1', ',', 'G
alaxy', 'Rugby', 'Pro', 'and', 'Galaxy', 'S', 'III', 'mini', '.']
```

[point: 1] Exercise 1a: Part-of-speech (POS) tagging

Use `nltk.pos_tag` to perform part-of-speech tagging on each sentence.

Use `print` to **show** the output in the notebook (and hence also in the exported PDF!).

In [9]:

```
pos_tags_per_sentence = []
for tokens in tokens_per_sentence:
    pos_tags = nltk.pos_tag(tokens)
    pos_tags_per_sentence.append(pos_tags)
    print(pos_tags)
```

```
[('https', 'NN'), (':', ':'), ('//www.telegraph.co.uk/technology/apple/9702716/Apple-S
amsung-lawsuit-six-more-products-under-scrutiny.html', 'JJ'), ('Documents', 'NNS'),
('filed', 'VBN'), ('to', 'TO'), ('the', 'DT'), ('San', 'NNP'), ('Jose', 'NNP'), ('fede
ral', 'JJ'), ('court', 'NN'), ('in', 'IN'), ('California', 'NNP'), ('on', 'IN'), ('Nov
ember', 'NNP'), ('23', 'CD'), ('list', 'NN'), ('six', 'CD'), ('Samsung', 'NNP'), ('pro
ducts', 'NNS'), ('running', 'VBG'), ('the', 'DT'), ('``', '``'), ('Jelly', 'RB'), ('Be
an', 'NNP'), ("''", "''"), ('and', 'CC'), ('``', '``'), ('Ice', 'NNP'), ('Cream', 'NN
P'), ('Sandwich', 'NNP'), ("''", "''"), ('operating', 'VBG'), ('systems', 'NNS'),
(',', ','), ('which', 'WDT'), ('Apple', 'NNP'), ('claims', 'VBZ'), ('infringe', 'VB'),
('its', 'PRP$'), ('patents', 'NNS'), ('.', '.')]
[('The', 'DT'), ('six', 'CD'), ('phones', 'NNS'), ('and', 'CC'), ('tablets', 'NNS'),
('affected', 'VBN'), ('are', 'VBP'), ('the', 'DT'), ('Galaxy', 'NNP'), ('S', 'NNP'),
('III', 'NNP'), (',', ','), ('running', 'VBG'), ('the', 'DT'), ('new', 'JJ'), ('Jell
y', 'NNP'), ('Bean', 'NNP'), ('system', 'NN'), (',', ','), ('the', 'DT'), ('Galaxy',
'NNP'), ('Tab', 'NNP'), ('8.9', 'CD'), ('Wifi', 'NNP'), ('tablet', 'NN'), (',', ','),
('the', 'DT'), ('Galaxy', 'NNP'), ('Tab', 'NNP'), ('2', 'CD'), ('10.1', 'CD'), (',',
','), ('Galaxy', 'NNP'), ('Rugby', 'NNP'), ('Pro', 'NNP'), ('and', 'CC'), ('Galaxy',
'NNP'), ('S', 'NNP'), ('III', 'NNP'), ('mini', 'NN'), ('.', '.')]
[('Apple', 'NNP'), ('stated', 'VBD'), ('it', 'PRP'), ('had', 'VBD'), ('“', 'NNP'), ('a
cted', 'VBD'), ('quickly', 'RB'), ('and', 'CC'), ('diligently', 'RB'), ("''", "''"),
('in', 'IN'), ('order', 'NN'), ('to', 'TO'), ('``', '``'), ('determine', 'VB'), ('tha
t', 'IN'), ('these', 'DT'), ('newly', 'RB'), ('released', 'VBN'), ('products', 'NNS'),
('do', 'VBP'), ('infringe', 'VB'), ('many', 'JJ'), ('of', 'IN'), ('the', 'DT'), ('sam
e', 'JJ'), ('claims', 'NNS'), ('already', 'RB'), ('asserted', 'VBN'), ('by', 'IN'),
('Apple', 'NNP'), ('.', '.'), ("''", "''")]
[('In', 'IN'), ('August', 'NNP'), (',', ','), ('Samsung', 'NNP'), ('lost', 'VBD'),
('a', 'DT'), ('US', 'NNP'), ('patent', 'NN'), ('case', 'NN'), ('to', 'TO'), ('Apple',
'NNP'), ('and', 'CC'), ('was', 'VBD'), ('ordered', 'VBN'), ('to', 'TO'), ('pay', 'V
B'), ('its', 'PRP$'), ('rival', 'JJ'), ('$', '$'), ('1.05bn', 'CD'), ('(', '('), ('£0.
66bn', 'NN'), (')', ')'), ('in', 'IN'), ('damages', 'NNS'), ('for', 'IN'), ('copying',
'VBG'), ('features', 'NNS'), ('of', 'IN'), ('the', 'DT'), ('iPad', 'NN'), ('and', 'C
C'), ('iPhone', 'NN'), ('in', 'IN'), ('its', 'PRP$'), ('Galaxy', 'NNP'), ('range', 'N
N'), ('of', 'IN'), ('devices', 'NNS'), ('.', '.')]
[('Samsung', 'NNP'), (',', ','), ('which', 'WDT'), ('is', 'VBZ'), ('the', 'DT'), ('wor
ld', 'NN'), ("'s", 'POS'), ('top', 'JJ'), ('mobile', 'NN'), ('phone', 'NN'), ('maker',
'NN'), (',', ','), ('is', 'VBZ'), ('appealing', 'VBG'), ('the', 'DT'), ('ruling', 'N
N'), ('.', '.')]
[('A', 'DT'), ('similar', 'JJ'), ('case', 'NN'), ('in', 'IN'), ('the', 'DT'), ('UK',
'NNP'), ('found', 'VBD'), ('in', 'IN'), ('Samsung', 'NNP'), ("'s", 'POS'), ('favour',
'NN'), ('and', 'CC'), ('ordered', 'VBD'), ('Apple', 'NNP'), ('to', 'TO'), ('publish',
'VB'), ('an', 'DT'), ('apology', 'NN'), ('making', 'VBG'), ('clear', 'JJ'), ('that',
'IN'), ('the', 'DT'), ('South', 'JJ'), ('Korean', 'JJ'), ('firm', 'NN'), ('had', 'VB
D'), ('not', 'RB'), ('copied', 'VBN'), ('its', 'PRP$'), ('iPad', 'NN'), ('when', 'WR
B'), ('designing', 'VBG'), ('its', 'PRP$'), ('own', 'JJ'), ('devices', 'NNS'), ('.',
'.')]
```

```
In [10]: print(pos_tags_per_sentence)
```

```
[[('https', 'NN'), (':', ':'), ('//www.telegraph.co.uk/technology/apple/9702716/Apple-
Samsung-lawsuit-six-more-products-under-scrutiny.html', 'JJ'), ('Documents', 'NNS'),
('filed', 'VBN'), ('to', 'TO'), ('the', 'DT'), ('San', 'NNP'), ('Jose', 'NNP'), ('fede
ral', 'JJ'), ('court', 'NN'), ('in', 'IN'), ('California', 'NNP'), ('on', 'IN'), ('Nov
ember', 'NNP'), ('23', 'CD'), ('list', 'NN'), ('six', 'CD'), ('Samsung', 'NNP'), ('pro
ducts', 'NNS'), ('running', 'VBG'), ('the', 'DT'), ('``', '``'), ('Jelly', 'RB'), ('Be
an', 'NNP'), ("''", "''"), ('and', 'CC'), ('``', '``'), ('Ice', 'NNP'), ('Cream', 'NN
P'), ('Sandwich', 'NNP'), ("''", "''"), ('operating', 'VBG'), ('systems', 'NNS'),
(',', ','), ('which', 'WDT'), ('Apple', 'NNP'), ('claims', 'VBZ'), ('infringe', 'VB'),
('its', 'PRP$'), ('patents', 'NNS'), ('.', '.')], [('The', 'DT'), ('six', 'CD'), ('pho
nes', 'NNS'), ('and', 'CC'), ('tablets', 'NNS'), ('affected', 'VBN'), ('are', 'VBP'),
('the', 'DT'), ('Galaxy', 'NNP'), ('S', 'NNP'), ('III', 'NNP'), (',', ','), ('runnin
g', 'VBG'), ('the', 'DT'), ('new', 'JJ'), ('Jelly', 'NNP'), ('Bean', 'NNP'), ('syste
m', 'NN'), (',', ','), ('the', 'DT'), ('Galaxy', 'NNP'), ('Tab', 'NNP'), ('8.9', 'C
D'), ('Wifi', 'NNP'), ('tablet', 'NN'), (',', ','), ('the', 'DT'), ('Galaxy', 'NNP'),
('Tab', 'NNP'), ('2', 'CD'), ('10.1', 'CD'), (',', ','), ('Galaxy', 'NNP'), ('Rugby',
'NNP'), ('Pro', 'NNP'), ('and', 'CC'), ('Galaxy', 'NNP'), ('S', 'NNP'), ('III', 'NN
P'), ('mini', 'NN'), ('.', '.')], [('Apple', 'NNP'), ('stated', 'VBD'), ('it', 'PRP'),
('had', 'VBD'), ('“', 'NNP'), ('acted', 'VBD'), ('quickly', 'RB'), ('and', 'CC'), ('di
ligently', 'RB'), ("''", "''"), ('in', 'IN'), ('order', 'NN'), ('to', 'TO'), ('``', '`
`'), ('determine', 'VB'), ('that', 'IN'), ('these', 'DT'), ('newly', 'RB'), ('release
d', 'VBN'), ('products', 'NNS'), ('do', 'VBP'), ('infringe', 'VB'), ('many', 'JJ'),
('of', 'IN'), ('the', 'DT'), ('same', 'JJ'), ('claims', 'NNS'), ('already', 'RB'), ('a
sserted', 'VBN'), ('by', 'IN'), ('Apple', 'NNP'), ('.', '.'), ("''", "''")], [('In',
'IN'), ('August', 'NNP'), (',', ','), ('Samsung', 'NNP'), ('lost', 'VBD'), ('a', 'D
T'), ('US', 'NNP'), ('patent', 'NN'), ('case', 'NN'), ('to', 'TO'), ('Apple', 'NNP'),
('and', 'CC'), ('was', 'VBD'), ('ordered', 'VBN'), ('to', 'TO'), ('pay', 'VB'), ('it
s', 'PRP$'), ('rival', 'JJ'), ('$', '$'), ('1.05bn', 'CD'), ('(', '('), ('£0.66bn', 'N
N'), (')', ')'), ('in', 'IN'), ('damages', 'NNS'), ('for', 'IN'), ('copying', 'VBG'),
('features', 'NNS'), ('of', 'IN'), ('the', 'DT'), ('iPad', 'NN'), ('and', 'CC'), ('iPh
one', 'NN'), ('in', 'IN'), ('its', 'PRP$'), ('Galaxy', 'NNP'), ('range', 'NN'), ('of',
'IN'), ('devices', 'NNS'), ('.', '.')], [('Samsung', 'NNP'), (',', ','), ('which', 'WD
T'), ('is', 'VBZ'), ('the', 'DT'), ('world', 'NN'), ("'s", 'POS'), ('top', 'JJ'), ('mo
bile', 'NN'), ('phone', 'NN'), ('maker', 'NN'), (',', ','), ('is', 'VBZ'), ('appealin
g', 'VBG'), ('the', 'DT'), ('ruling', 'NN'), ('.', '.')], [('A', 'DT'), ('similar', 'J
J'), ('case', 'NN'), ('in', 'IN'), ('the', 'DT'), ('UK', 'NNP'), ('found', 'VBD'), ('i
n', 'IN'), ('Samsung', 'NNP'), ("'s", 'POS'), ('favour', 'NN'), ('and', 'CC'), ('order
ed', 'VBD'), ('Apple', 'NNP'), ('to', 'TO'), ('publish', 'VB'), ('an', 'DT'), ('apolog
y', 'NN'), ('making', 'VBG'), ('clear', 'JJ'), ('that', 'IN'), ('the', 'DT'), ('Sout
h', 'JJ'), ('Korean', 'JJ'), ('firm', 'NN'), ('had', 'VBD'), ('not', 'RB'), ('copied',
'VBN'), ('its', 'PRP$'), ('iPad', 'NN'), ('when', 'WRB'), ('designing', 'VBG'), ('it
s', 'PRP$'), ('own', 'JJ'), ('devices', 'NNS'), ('.', '.')]]
```

[point: 1] Exercise 1b: Named Entity Recognition (NER)

Use `nltk.chunk.ne_chunk` to perform Named Entity Recognition (NER) on each sentence.

Use `print` to **show** the output in the notebook (and hence also in the exported PDF!).

```
In [11]: ner_per_sentence = []
         for tokens in tokens_per_sentence:
             ner_tags = nltk.chunk.ne_chunk(nltk.pos_tag(tokens))
             ner_per_sentence.append(ner_tags)
             print(ner_tags)
```

```
(S
  https/NN
  :/:
  //www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more-produc
ts-under-scrutiny.html/JJ
  Documents/NNS
  filed/VBN
  to/TO
  the/DT
  (ORGANIZATION San/NNP Jose/NNP)
  federal/JJ
  court/NN
  in/IN
  (GPE California/NNP)
  on/IN
  November/NNP
  23/CD
  list/NN
  six/CD
  (ORGANIZATION Samsung/NNP)
  products/NNS
  running/VBG
  the/DT
  ``/``
  Jelly/RB
  (GPE Bean/NNP)
  ''/''
  and/CC
  ``/``
  Ice/NNP
  Cream/NNP
  Sandwich/NNP
  ''/''
  operating/VBG
  systems/NNS
  ,/,
  which/WDT
  (PERSON Apple/NNP)
  claims/VBZ
  infringe/VB
  its/PRP$
  patents/NNS
  ./.)
(S
  The/DT
  six/CD
  phones/NNS
  and/CC
  tablets/NNS
  affected/VBN
  are/VBP
  the/DT
  (ORGANIZATION Galaxy/NNP)
  S/NNP
  III/NNP
  ,/,
  running/VBG
  the/DT
  new/JJ
  (PERSON Jelly/NNP Bean/NNP)
  system/NN
  ,/,
  the/DT
  (ORGANIZATION Galaxy/NNP)
  Tab/NNP
  8.9/CD
  Wifi/NNP
  tablet/NN
  ,/,
  the/DT
  (ORGANIZATION Galaxy/NNP)
  Tab/NNP
  2/CD
  10.1/CD
  ,/,
  (PERSON Galaxy/NNP Rugby/NNP Pro/NNP)
  and/CC
  (PERSON Galaxy/NNP S/NNP)
  III/NNP
  mini/NN
  ./.)
(S
  (PERSON Apple/NNP)
  stated/VBD
  it/PRP
  had/VBD
  “/NNP
  acted/VBD
  quickly/RB
  and/CC
  diligently/RB
  ''/''
  in/IN
  order/NN
  to/TO
  ``/``
  determine/VB
  that/IN
  these/DT
  newly/RB
  released/VBN
  products/NNS
  do/VBP
  infringe/VB
  many/JJ
  of/IN
  the/DT
  same/JJ
  claims/NNS
  already/RB
  asserted/VBN
  by/IN
  (PERSON Apple/NNP)
  ./.
  ''/'')
(S
  In/IN
  (GPE August/NNP)
  ,/,
  (PERSON Samsung/NNP)
  lost/VBD
  a/DT
  (GSP US/NNP)
  patent/NN
  case/NN
  to/TO
  (GPE Apple/NNP)
  and/CC
  was/VBD
  ordered/VBN
  to/TO
  pay/VB
  its/PRP$
  rival/JJ
  $/$
  1.05bn/CD
  (/(
  £0.66bn/NN
  )/)
  in/IN
  damages/NNS
  for/IN
  copying/VBG
  features/NNS
  of/IN
  the/DT
  (ORGANIZATION iPad/NN)
  and/CC
  (ORGANIZATION iPhone/NN)
  in/IN
  its/PRP$
  (GPE Galaxy/NNP)
  range/NN
  of/IN
  devices/NNS
  ./.)
(S
  (GPE Samsung/NNP)
  ,/,
  which/WDT
  is/VBZ
  the/DT
  world/NN
  's/POS
  top/JJ
  mobile/NN
  phone/NN
  maker/NN
  ,/,
  is/VBZ
  appealing/VBG
  the/DT
  ruling/NN
  ./.)
(S
  A/DT
  similar/JJ
  case/NN
  in/IN
  the/DT
  (ORGANIZATION UK/NNP)
  found/VBD
  in/IN
  (GPE Samsung/NNP)
  's/POS
  favour/NN
  and/CC
  ordered/VBD
  (PERSON Apple/NNP)
```

```
to/TO
publish/VB
an/DT
apology/NN
making/VBG
clear/JJ
that/IN
the/DT
(LOCATION South/JJ Korean/JJ)
firm/NN
had/VBD
not/RB
copied/VBN
its/PRP$
iPad/NN
when/WRB
designing/VBG
its/PRP$
own/JJ
devices/NNS
./.)
```

In [12]:

```
print(ner_per_sentence)
```

```
[Tree('S', [('https', 'NN'), (':', ':'), ('//www.telegraph.co.uk/technology/apple/9702
716/Apple-Samsung-lawsuit-six-more-products-under-scrutiny.html', 'JJ'), ('Documents',
'NNS'), ('filed', 'VBN'), ('to', 'TO'), ('the', 'DT'), Tree('ORGANIZATION', [('San',
'NNP'), ('Jose', 'NNP')]), ('federal', 'JJ'), ('court', 'NN'), ('in', 'IN'), Tree('GP
E', [('California', 'NNP')]), ('on', 'IN'), ('November', 'NNP'), ('23', 'CD'), ('lis
t', 'NN'), ('six', 'CD'), Tree('ORGANIZATION', [('Samsung', 'NNP')]), ('products', 'NN
S'), ('running', 'VBG'), ('the', 'DT'), ('``', '``'), ('Jelly', 'RB'), Tree('GPE',
[('Bean', 'NNP')]), ("''", "''"), ('and', 'CC'), ('``', '``'), ('Ice', 'NNP'), ('Crea
m', 'NNP'), ('Sandwich', 'NNP'), ("''", "''"), ('operating', 'VBG'), ('systems', 'NN
S'), (',', ','), ('which', 'WDT'), Tree('PERSON', [('Apple', 'NNP')]), ('claims', 'VB
Z'), ('infringe', 'VB'), ('its', 'PRP$'), ('patents', 'NNS'), ('.', '.')]), Tree('S',
[('The', 'DT'), ('six', 'CD'), ('phones', 'NNS'), ('and', 'CC'), ('tablets', 'NNS'),
('affected', 'VBN'), ('are', 'VBP'), ('the', 'DT'), Tree('ORGANIZATION', [('Galaxy',
'NNP')]), ('S', 'NNP'), ('III', 'NNP'), (',', ','), ('running', 'VBG'), ('the', 'DT'),
('new', 'JJ'), Tree('PERSON', [('Jelly', 'NNP'), ('Bean', 'NNP')]), ('system', 'NN'),
(',', ','), ('the', 'DT'), Tree('ORGANIZATION', [('Galaxy', 'NNP')]), ('Tab', 'NNP'),
('8.9', 'CD'), ('Wifi', 'NNP'), ('tablet', 'NN'), (',', ','), ('the', 'DT'), Tree('ORG
ANIZATION', [('Galaxy', 'NNP')]), ('Tab', 'NNP'), ('2', 'CD'), ('10.1', 'CD'), (',',
','), Tree('PERSON', [('Galaxy', 'NNP'), ('Rugby', 'NNP'), ('Pro', 'NNP')]), ('and',
'CC'), Tree('PERSON', [('Galaxy', 'NNP'), ('S', 'NNP')]), ('III', 'NNP'), ('mini', 'N
N'), ('.', '.')]), Tree('S', [Tree('PERSON', [('Apple', 'NNP')]), ('stated', 'VBD'),
('it', 'PRP'), ('had', 'VBD'), ('“', 'NNP'), ('acted', 'VBD'), ('quickly', 'RB'), ('an
d', 'CC'), ('diligently', 'RB'), ("''", "''"), ('in', 'IN'), ('order', 'NN'), ('to',
'TO'), ('``', '``'), ('determine', 'VB'), ('that', 'IN'), ('these', 'DT'), ('newly',
'RB'), ('released', 'VBN'), ('products', 'NNS'), ('do', 'VBP'), ('infringe', 'VB'),
('many', 'JJ'), ('of', 'IN'), ('the', 'DT'), ('same', 'JJ'), ('claims', 'NNS'), ('alre
ady', 'RB'), ('asserted', 'VBN'), ('by', 'IN'), Tree('PERSON', [('Apple', 'NNP')]),
('.', '.'), ("''", "''")]), Tree('S', [('In', 'IN'), Tree('GPE', [('August', 'NNP')]),
(',', ','), Tree('PERSON', [('Samsung', 'NNP')]), ('lost', 'VBD'), ('a', 'DT'), Tree
('GSP', [('US', 'NNP')]), ('patent', 'NN'), ('case', 'NN'), ('to', 'TO'), Tree('GPE',
[('Apple', 'NNP')]), ('and', 'CC'), ('was', 'VBD'), ('ordered', 'VBN'), ('to', 'TO'),
('pay', 'VB'), ('its', 'PRP$'), ('rival', 'JJ'), ('$', '$'), ('1.05bn', 'CD'), ('(',
'('), ('£0.66bn', 'NN'), (')', ')'), ('in', 'IN'), ('damages', 'NNS'), ('for', 'IN'),
('copying', 'VBG'), ('features', 'NNS'), ('of', 'IN'), ('the', 'DT'), Tree('ORGANIZATI
ON', [('iPad', 'NN')]), ('and', 'CC'), Tree('ORGANIZATION', [('iPhone', 'NN')]), ('i
n', 'IN'), ('its', 'PRP$'), Tree('GPE', [('Galaxy', 'NNP')]), ('range', 'NN'), ('of',
'IN'), ('devices', 'NNS'), ('.', '.')]), Tree('S', [Tree('GPE', [('Samsung', 'NNP')]),
(',', ','), ('which', 'WDT'), ('is', 'VBZ'), ('the', 'DT'), ('world', 'NN'), ("'s", 'P
OS'), ('top', 'JJ'), ('mobile', 'NN'), ('phone', 'NN'), ('maker', 'NN'), (',', ','),
('is', 'VBZ'), ('appealing', 'VBG'), ('the', 'DT'), ('ruling', 'NN'), ('.', '.')]), Tr
ee('S', [('A', 'DT'), ('similar', 'JJ'), ('case', 'NN'), ('in', 'IN'), ('the', 'DT'),
Tree('ORGANIZATION', [('UK', 'NNP')]), ('found', 'VBD'), ('in', 'IN'), Tree('GPE',
[('Samsung', 'NNP')]), ("'s", 'POS'), ('favour', 'NN'), ('and', 'CC'), ('ordered', 'VB
D'), Tree('PERSON', [('Apple', 'NNP')]), ('to', 'TO'), ('publish', 'VB'), ('an', 'D
T'), ('apology', 'NN'), ('making', 'VBG'), ('clear', 'JJ'), ('that', 'IN'), ('the', 'D
T'), Tree('LOCATION', [('South', 'JJ'), ('Korean', 'JJ')]), ('firm', 'NN'), ('had', 'V
BD'), ('not', 'RB'), ('copied', 'VBN'), ('its', 'PRP$'), ('iPad', 'NN'), ('when', 'WR
B'), ('designing', 'VBG'), ('its', 'PRP$'), ('own', 'JJ'), ('devices', 'NNS'), ('.',
'.')])]
```

[points: 2] Exercise 1c: Constituency parsing

Use the `nltk.RegexpParser` to perform constituency parsing on each sentence.

Use `print` to **show** the output in the notebook (and hence also in the exported PDF!).

In [13]:

```
constituent_parser = nltk.RegexpParser('''
NP: {<DT>? <JJ>* <NN>*} # NP
P: {<IN>}           # Preposition
V: {<V.*>}          # Verb
PP: {<P> <NP>}      # PP -> P NP
VP: {<V> <NP|PP>*}  # VP -> V (NP|PP)*''')
```

In [14]:

```
constituency_output_per_sentence = []
for tokens in tokens_per_sentence:
    constituency_output = constituent_parser.parse(nltk.pos_tag(tokens))
    constituency_output_per_sentence.append(constituency_output)
    print(constituency_output)
```

```
(S
  (NP https/NN)
  :/:
  (NP
    //www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more-prod
ucts-under-scrutiny.html/JJ)
  Documents/NNS
  (VP (V filed/VBN))
  to/TO
  (NP the/DT)
  San/NNP
  Jose/NNP
  (NP federal/JJ court/NN)
  (P in/IN)
  California/NNP
  (P on/IN)
  November/NNP
  23/CD
  (NP list/NN)
  six/CD
  Samsung/NNP
  products/NNS
  (VP (V running/VBG) (NP the/DT))
  ``/``
  Jelly/RB
  Bean/NNP
  ''/''
  and/CC
  ``/``
  Ice/NNP
  Cream/NNP
  Sandwich/NNP
  ''/''
  (VP (V operating/VBG))
  systems/NNS
  ,/,
  which/WDT
  Apple/NNP
  (VP (V claims/VBZ))
  (VP (V infringe/VB))
  its/PRP$
  patents/NNS
  ./.)
(S
  (NP The/DT)
  six/CD
  phones/NNS
  and/CC
  tablets/NNS
  (VP (V affected/VBN))
  (VP (V are/VBP) (NP the/DT))
  Galaxy/NNP
  S/NNP
  III/NNP
  ,/,
  (VP (V running/VBG) (NP the/DT new/JJ))
  Jelly/NNP
  Bean/NNP
  (NP system/NN)
  ,/,
  (NP the/DT)
  Galaxy/NNP
  Tab/NNP
  8.9/CD
  Wifi/NNP
  (NP tablet/NN)
  ,/,
  (NP the/DT)
  Galaxy/NNP
  Tab/NNP
  2/CD
  10.1/CD
  ,/,
  Galaxy/NNP
  Rugby/NNP
  Pro/NNP
  and/CC
  Galaxy/NNP
  S/NNP
  III/NNP
  (NP mini/NN)
  ./.)
(S
  Apple/NNP
  (VP (V stated/VBD))
  it/PRP
  (VP (V had/VBD))
  “/NNP
  (VP (V acted/VBD))
  quickly/RB
  and/CC
  diligently/RB
  ''/''
  (PP (P in/IN) (NP order/NN))
  to/TO
  ``/``
  (VP (V determine/VB) (PP (P that/IN) (NP these/DT)))
  newly/RB
  (VP (V released/VBN))
  products/NNS
  (VP (V do/VBP))
  (VP
    (V infringe/VB)
    (NP many/JJ)
    (PP (P of/IN) (NP the/DT same/JJ)))
  claims/NNS
  already/RB
  (VP (V asserted/VBN))
  (P by/IN)
  Apple/NNP
  ./.
  ''/'')
(S
  (P In/IN)
  August/NNP
  ,/,
  Samsung/NNP
  (VP (V lost/VBD) (NP a/DT))
  US/NNP
  (NP patent/NN case/NN)
  to/TO
  Apple/NNP
  and/CC
  (VP (V was/VBD))
  (VP (V ordered/VBN))
  to/TO
```

```
    (VP (V pay/VB))
    its/PRP$
    (NP rival/JJ)
    $/$
    1.05bn/CD
    (/(
    (NP £0.66bn/NN)
    )/)
    (P in/IN)
    damages/NNS
    (P for/IN)
    (VP (V copying/VBG))
    features/NNS
    (PP (P of/IN) (NP the/DT iPad/NN))
    and/CC
    (NP iPhone/NN)
    (P in/IN)
    its/PRP$
    Galaxy/NNP
    (NP range/NN)
    (P of/IN)
    devices/NNS
    ./.)
  (S
    Samsung/NNP
    ,/,
    which/WDT
    (VP (V is/VBZ) (NP the/DT world/NN))
    's/POS
    (NP top/JJ mobile/NN phone/NN maker/NN)
    ,/,
    (VP (V is/VBZ))
    (VP (V appealing/VBG) (NP the/DT ruling/NN))
    ./.)
  (S
    (NP A/DT similar/JJ case/NN)
    (PP (P in/IN) (NP the/DT))
    UK/NNP
    (VP (V found/VBD))
    (P in/IN)
    Samsung/NNP
    's/POS
    (NP favour/NN)
    and/CC
    (VP (V ordered/VBD))
    Apple/NNP
    to/TO
    (VP (V publish/VB) (NP an/DT apology/NN))
    (VP
      (V making/VBG)
      (NP clear/JJ)
      (PP (P that/IN) (NP the/DT South/JJ Korean/JJ firm/NN)))
    (VP (V had/VBD))
    not/RB
    (VP (V copied/VBN))
    its/PRP$
    (NP iPad/NN)
    when/WRB
    (VP (V designing/VBG))
    its/PRP$
    (NP own/JJ)
    devices/NNS
    ./.)
```

In [15]:

```
print(constituency_output_per_sentence)
```

```
[Tree('S', [Tree('NP', [('https', 'NN')]), (':', ':'), Tree('NP', [('//www.telegraph.c
o.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more-products-under-scrutiny.h
tml', 'JJ')]), ('Documents', 'NNS'), Tree('VP', [Tree('V', [('filed', 'VBN')])]), ('t
o', 'TO'), Tree('NP', [('the', 'DT')]), ('San', 'NNP'), ('Jose', 'NNP'), Tree('NP',
[('federal', 'JJ'), ('court', 'NN')]), Tree('P', [('in', 'IN')]), ('California', 'NN
P'), Tree('P', [('on', 'IN')]), ('November', 'NNP'), ('23', 'CD'), Tree('NP', [('lis
t', 'NN')]), ('six', 'CD'), ('Samsung', 'NNP'), ('products', 'NNS'), Tree('VP', [Tree
('V', [('running', 'VBG')]), Tree('NP', [('the', 'DT')])]), ('``', '``'), ('Jelly', 'R
B'), ('Bean', 'NNP'), ("''", "''"), ('and', 'CC'), ('``', '``'), ('Ice', 'NNP'), ('Cre
am', 'NNP'), ('Sandwich', 'NNP'), ("''", "''"), Tree('VP', [Tree('V', [('operating',
'VBG')])]), ('systems', 'NNS'), (',', ','), ('which', 'WDT'), ('Apple', 'NNP'), Tree
('VP', [Tree('V', [('claims', 'VBZ')])]), Tree('VP', [Tree('V', [('infringe', 'V
B')])]), ('its', 'PRP$'), ('patents', 'NNS'), ('.', '.')]), Tree('S', [Tree('NP', [('T
he', 'DT')]), ('six', 'CD'), ('phones', 'NNS'), ('and', 'CC'), ('tablets', 'NNS'), Tre
e('VP', [Tree('V', [('affected', 'VBN')])]), Tree('VP', [Tree('V', [('are', 'VBP')]),
Tree('NP', [('the', 'DT')])]), ('Galaxy', 'NNP'), ('S', 'NNP'), ('III', 'NNP'), (',',
','), Tree('VP', [Tree('V', [('running', 'VBG')]), Tree('NP', [('the', 'DT'), ('new',
'JJ')])]), ('Jelly', 'NNP'), ('Bean', 'NNP'), Tree('NP', [('system', 'NN')]), (',',
','), Tree('NP', [('the', 'DT')]), ('Galaxy', 'NNP'), ('Tab', 'NNP'), ('8.9', 'CD'),
('Wifi', 'NNP'), Tree('NP', [('tablet', 'NN')]), (',', ','), Tree('NP', [('the', 'D
T')]), ('Galaxy', 'NNP'), ('Tab', 'NNP'), ('2', 'CD'), ('10.1', 'CD'), (',', ','), ('G
alaxy', 'NNP'), ('Rugby', 'NNP'), ('Pro', 'NNP'), ('and', 'CC'), ('Galaxy', 'NNP'),
('S', 'NNP'), ('III', 'NNP'), Tree('NP', [('mini', 'NN')]), ('.', '.')]), Tree('S',
[('Apple', 'NNP'), Tree('VP', [Tree('V', [('stated', 'VBD')])]), ('it', 'PRP'), Tree
('VP', [Tree('V', [('had', 'VBD')])]), ('“', 'NNP'), Tree('VP', [Tree('V', [('acted',
'VBD')])]), ('quickly', 'RB'), ('and', 'CC'), ('diligently', 'RB'), ("''", "''"), Tree
('PP', [Tree('P', [('in', 'IN')]), Tree('NP', [('order', 'NN')])]), ('to', 'TO'), ('`
`', '``'), Tree('VP', [Tree('V', [('determine', 'VB')]), Tree('PP', [Tree('P', [('tha
t', 'IN')]), Tree('NP', [('these', 'DT')])])]), ('newly', 'RB'), Tree('VP', [Tree('V',
[('released', 'VBN')])]), ('products', 'NNS'), Tree('VP', [Tree('V', [('do', 'VB
P')])]), Tree('VP', [Tree('V', [('infringe', 'VB')]), Tree('NP', [('many', 'JJ')]), Tr
ee('PP', [Tree('P', [('of', 'IN')]), Tree('NP', [('the', 'DT'), ('same', 'JJ')])])]),
('claims', 'NNS'), ('already', 'RB'), Tree('VP', [Tree('V', [('asserted', 'VBN')])]),
Tree('P', [('by', 'IN')]), ('Apple', 'NNP'), ('.', '.'), ("''", "''")]), Tree('S', [Tr
ee('P', [('In', 'IN')]), ('August', 'NNP'), (',', ','), ('Samsung', 'NNP'), Tree('VP',
[Tree('V', [('lost', 'VBD')]), Tree('NP', [('a', 'DT')])]), ('US', 'NNP'), Tree('NP',
[('patent', 'NN'), ('case', 'NN')]), ('to', 'TO'), ('Apple', 'NNP'), ('and', 'CC'), Tr
ee('VP', [Tree('V', [('was', 'VBD')])]), Tree('VP', [Tree('V', [('ordered', 'VB
N')])]), ('to', 'TO'), Tree('VP', [Tree('V', [('pay', 'VB')])]), ('its', 'PRP$'), Tree
('NP', [('rival', 'JJ')]), ('$', '$'), ('1.05bn', 'CD'), ('(', '('), Tree('NP', [('£0.
66bn', 'NN')]), (')', ')'), Tree('P', [('in', 'IN')]), ('damages', 'NNS'), Tree('P',
[('for', 'IN')]), Tree('VP', [Tree('V', [('copying', 'VBG')])]), ('features', 'NNS'),
Tree('PP', [Tree('P', [('of', 'IN')]), Tree('NP', [('the', 'DT'), ('iPad', 'NN')])]),
('and', 'CC'), Tree('NP', [('iPhone', 'NN')]), Tree('P', [('in', 'IN')]), ('its', 'PRP
$'), ('Galaxy', 'NNP'), Tree('NP', [('range', 'NN')]), Tree('P', [('of', 'IN')]), ('de
vices', 'NNS'), ('.', '.')]), Tree('S', [('Samsung', 'NNP'), (',', ','), ('which', 'WD
T'), Tree('VP', [Tree('V', [('is', 'VBZ')]), Tree('NP', [('the', 'DT'), ('world', 'N
N')])]), ("'s", 'POS'), Tree('NP', [('top', 'JJ'), ('mobile', 'NN'), ('phone', 'NN'),
('maker', 'NN')]), (',', ','), Tree('VP', [Tree('V', [('is', 'VBZ')])]), Tree('VP', [T
ree('V', [('appealing', 'VBG')]), Tree('NP', [('the', 'DT'), ('ruling', 'NN')])]),
('.', '.')]), Tree('S', [Tree('NP', [('A', 'DT'), ('similar', 'JJ'), ('case', 'NN')]),
Tree('PP', [Tree('P', [('in', 'IN')]), Tree('NP', [('the', 'DT')])]), ('UK', 'NNP'), T
ree('VP', [Tree('V', [('found', 'VBD')])]), Tree('P', [('in', 'IN')]), ('Samsung', 'NN
P'), ("'s", 'POS'), Tree('NP', [('favour', 'NN')]), ('and', 'CC'), Tree('VP', [Tree
('V', [('ordered', 'VBD')])]), ('Apple', 'NNP'), ('to', 'TO'), Tree('VP', [Tree('V',
[('publish', 'VB')]), Tree('NP', [('an', 'DT'), ('apology', 'NN')])]), Tree('VP', [Tre
e('V', [('making', 'VBG')]), Tree('NP', [('clear', 'JJ')]), Tree('PP', [Tree('P', [('t
hat', 'IN')]), Tree('NP', [('the', 'DT'), ('South', 'JJ'), ('Korean', 'JJ'), ('firm',
'NN')])])]), Tree('VP', [Tree('V', [('had', 'VBD')])]), ('not', 'RB'), Tree('VP', [Tre
e('V', [('copied', 'VBN')])]), ('its', 'PRP$'), Tree('NP', [('iPad', 'NN')]), ('when',
'WRB'), Tree('VP', [Tree('V', [('designing', 'VBG')])]), ('its', 'PRP$'), Tree('NP',
[('own', 'JJ')]), ('devices', 'NNS'), ('.', '.')])]
```

Augment the RegexpParser so that it also detects Named Entity Phrases (NEP), e.g., that it detects *Galaxy S III* and *Ice Cream Sandwich*

In [16]:

```
constituent_parser_v2 = nltk.RegexpParser('''
NP: {<DT>? <JJ>* <NN>*} # NP
P: {<IN>}           # Preposition
V: {<V.*>}          # Verb
PP: {<P> <NP>}      # PP -> P NP
VP: {<V> <NP|PP>*}  # VP -> V (NP|PP)*
NEP: {}             # ???''')
```

In [17]:

```
constituency_v2_output_per_sentence = []
for tokens in tokens_per_sentence:
    constituency_output = constituent_parser_v2.parse(nltk.pos_tag(tokens))
    constituency_v2_output_per_sentence.append(constituency_output)
    print(constituency_output)
```

```
(S
  (NP https/NN)
  :/:
  (NP
    //www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more-prod
ucts-under-scrutiny.html/JJ)
  Documents/NNS
  (VP (V filed/VBN))
  to/TO
  (NP the/DT)
  San/NNP
  Jose/NNP
  (NP federal/JJ court/NN)
  (P in/IN)
  California/NNP
  (P on/IN)
  November/NNP
  23/CD
  (NP list/NN)
  six/CD
  Samsung/NNP
  products/NNS
  (VP (V running/VBG) (NP the/DT))
  ``/``
  Jelly/RB
  Bean/NNP
  ''/''
  and/CC
  ``/``
  Ice/NNP
  Cream/NNP
  Sandwich/NNP
  ''/''
  (VP (V operating/VBG))
  systems/NNS
  ,/,
  which/WDT
  Apple/NNP
  (VP (V claims/VBZ))
  (VP (V infringe/VB))
  its/PRP$
  patents/NNS
  ./.)
(S
  (NP The/DT)
  six/CD
  phones/NNS
  and/CC
  tablets/NNS
  (VP (V affected/VBN))
  (VP (V are/VBP) (NP the/DT))
  Galaxy/NNP
  S/NNP
  III/NNP
  ,/,
  (VP (V running/VBG) (NP the/DT new/JJ))
  Jelly/NNP
  Bean/NNP
  (NP system/NN)
  ,/,
  (NP the/DT)
  Galaxy/NNP
  Tab/NNP
  8.9/CD
  Wifi/NNP
  (NP tablet/NN)
  ,/,
  (NP the/DT)
  Galaxy/NNP
  Tab/NNP
  2/CD
  10.1/CD
  ,/,
  Galaxy/NNP
  Rugby/NNP
  Pro/NNP
  and/CC
  Galaxy/NNP
  S/NNP
  III/NNP
  (NP mini/NN)
  ./.)
(S
  Apple/NNP
  (VP (V stated/VBD))
  it/PRP
  (VP (V had/VBD))
  “/NNP
  (VP (V acted/VBD))
  quickly/RB
  and/CC
  diligently/RB
  ''/''
  (PP (P in/IN) (NP order/NN))
  to/TO
  ``/``
  (VP (V determine/VB) (PP (P that/IN) (NP these/DT)))
  newly/RB
  (VP (V released/VBN))
  products/NNS
  (VP (V do/VBP))
  (VP
    (V infringe/VB)
    (NP many/JJ)
    (PP (P of/IN) (NP the/DT same/JJ)))
  claims/NNS
  already/RB
  (VP (V asserted/VBN))
  (P by/IN)
  Apple/NNP
  ./.
  ''/'')
(S
  (P In/IN)
  August/NNP
  ,/,
  Samsung/NNP
  (VP (V lost/VBD) (NP a/DT))
  US/NNP
  (NP patent/NN case/NN)
  to/TO
  Apple/NNP
  and/CC
  (VP (V was/VBD))
  (VP (V ordered/VBN))
  to/TO
  (VP (V pay/VB))
  its/PRP$
  (NP rival/JJ)
  $/$
  1.05bn/CD
  (/(
  (NP £0.66bn/NN)
  )/)
  (P in/IN)
  damages/NNS
  (P for/IN)
  (VP (V copying/VBG))
  features/NNS
  (PP (P of/IN) (NP the/DT iPad/NN))
  and/CC
  (NP iPhone/NN)
  (P in/IN)
  its/PRP$
  Galaxy/NNP
  (NP range/NN)
  (P of/IN)
  devices/NNS
  ./.)
(S
  Samsung/NNP
  ,/,
  which/WDT
  (VP (V is/VBZ) (NP the/DT world/NN))
  's/POS
  (NP top/JJ mobile/NN phone/NN maker/NN)
  ,/,
  (VP (V is/VBZ))
  (VP (V appealing/VBG) (NP the/DT ruling/NN))
  ./.)
(S
  (NP A/DT similar/JJ case/NN)
  (PP (P in/IN) (NP the/DT))
  UK/NNP
  (VP (V found/VBD))
  (P in/IN)
  Samsung/NNP
  's/POS
  (NP favour/NN)
  and/CC
  (VP (V ordered/VBD))
  Apple/NNP
  to/TO
  (VP (V publish/VB) (NP an/DT apology/NN))
  (VP
    (V making/VBG)
    (NP clear/JJ)
    (PP (P that/IN) (NP the/DT South/JJ Korean/JJ firm/NN)))
  (VP (V had/VBD))
  not/RB
  (VP (V copied/VBN))
  its/PRP$
  (NP iPad/NN)
  when/WRB
  (VP (V designing/VBG))
  its/PRP$
  (NP own/JJ)
  devices/NNS
  ./.)
```

```
In [18]: print(constituency_v2_output_per_sentence)
```

```
[Tree('S', [Tree('NP', [('https', 'NN')]), (':', ':'), Tree('NP', [('//www.telegraph.c
o.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more-products-under-scrutiny.h
tml', 'JJ')]), ('Documents', 'NNS'), Tree('VP', [Tree('V', [('filed', 'VBN')])]), ('t
o', 'TO'), Tree('NP', [('the', 'DT')]), ('San', 'NNP'), ('Jose', 'NNP'), Tree('NP',
[('federal', 'JJ'), ('court', 'NN')]), Tree('P', [('in', 'IN')]), ('California', 'NN
P'), Tree('P', [('on', 'IN')]), ('November', 'NNP'), ('23', 'CD'), Tree('NP', [('lis
t', 'NN')]), ('six', 'CD'), ('Samsung', 'NNP'), ('products', 'NNS'), Tree('VP', [Tree
('V', [('running', 'VBG')]), Tree('NP', [('the', 'DT')])]), ('``', '``'), ('Jelly', 'R
B'), ('Bean', 'NNP'), ("''", "''"), ('and', 'CC'), ('``', '``'), ('Ice', 'NNP'), ('Cre
am', 'NNP'), ('Sandwich', 'NNP'), ("''", "''"), Tree('VP', [Tree('V', [('operating',
'VBG')])]), ('systems', 'NNS'), (',', ','), ('which', 'WDT'), ('Apple', 'NNP'), Tree
('VP', [Tree('V', [('claims', 'VBZ')])]), Tree('VP', [Tree('V', [('infringe', 'V
B')])]), ('its', 'PRP$'), ('patents', 'NNS'), ('.', '.')]), Tree('S', [Tree('NP', [('T
he', 'DT')]), ('six', 'CD'), ('phones', 'NNS'), ('and', 'CC'), ('tablets', 'NNS'), Tre
e('VP', [Tree('V', [('affected', 'VBN')])]), Tree('VP', [Tree('V', [('are', 'VBP')]),
Tree('NP', [('the', 'DT')])]), ('Galaxy', 'NNP'), ('S', 'NNP'), ('III', 'NNP'), (',',
','), Tree('VP', [Tree('V', [('running', 'VBG')]), Tree('NP', [('the', 'DT'), ('new',
'JJ')])]), ('Jelly', 'NNP'), ('Bean', 'NNP'), Tree('NP', [('system', 'NN')]), (',',
','), Tree('NP', [('the', 'DT')]), ('Galaxy', 'NNP'), ('Tab', 'NNP'), ('8.9', 'CD'),
('Wifi', 'NNP'), Tree('NP', [('tablet', 'NN')]), (',', ','), Tree('NP', [('the', 'D
T')]), ('Galaxy', 'NNP'), ('Tab', 'NNP'), ('2', 'CD'), ('10.1', 'CD'), (',', ','), ('G
alaxy', 'NNP'), ('Rugby', 'NNP'), ('Pro', 'NNP'), ('and', 'CC'), ('Galaxy', 'NNP'),
('S', 'NNP'), ('III', 'NNP'), Tree('NP', [('mini', 'NN')]), ('.', '.')]), Tree('S',
[('Apple', 'NNP'), Tree('VP', [Tree('V', [('stated', 'VBD')])]), ('it', 'PRP'), Tree
('VP', [Tree('V', [('had', 'VBD')])]), ('“', 'NNP'), Tree('VP', [Tree('V', [('acted',
'VBD')])]), ('quickly', 'RB'), ('and', 'CC'), ('diligently', 'RB'), ("''", "''"), Tree
('PP', [Tree('P', [('in', 'IN')]), Tree('NP', [('order', 'NN')])]), ('to', 'TO'), ('`
`', '``'), Tree('VP', [Tree('V', [('determine', 'VB')]), Tree('PP', [Tree('P', [('tha
t', 'IN')]), Tree('NP', [('these', 'DT')])])]), ('newly', 'RB'), Tree('VP', [Tree('V',
[('released', 'VBN')])]), ('products', 'NNS'), Tree('VP', [Tree('V', [('do', 'VB
P')])]), Tree('VP', [Tree('V', [('infringe', 'VB')]), Tree('NP', [('many', 'JJ')]), Tr
ee('PP', [Tree('P', [('of', 'IN')]), Tree('NP', [('the', 'DT'), ('same', 'JJ')])])]),
('claims', 'NNS'), ('already', 'RB'), Tree('VP', [Tree('V', [('asserted', 'VBN')])]),
Tree('P', [('by', 'IN')]), ('Apple', 'NNP'), ('.', '.'), ("''", "''")]), Tree('S', [Tr
ee('P', [('In', 'IN')]), ('August', 'NNP'), (',', ','), ('Samsung', 'NNP'), Tree('VP',
[Tree('V', [('lost', 'VBD')]), Tree('NP', [('a', 'DT')])]), ('US', 'NNP'), Tree('NP',
[('patent', 'NN'), ('case', 'NN')]), ('to', 'TO'), ('Apple', 'NNP'), ('and', 'CC'), Tr
ee('VP', [Tree('V', [('was', 'VBD')])]), Tree('VP', [Tree('V', [('ordered', 'VB
N')])]), ('to', 'TO'), Tree('VP', [Tree('V', [('pay', 'VB')])]), ('its', 'PRP$'), Tree
('NP', [('rival', 'JJ')]), ('$', '$'), ('1.05bn', 'CD'), ('(', '('), Tree('NP', [('£0.
66bn', 'NN')]), (')', ')'), Tree('P', [('in', 'IN')]), ('damages', 'NNS'), Tree('P',
[('for', 'IN')]), Tree('VP', [Tree('V', [('copying', 'VBG')])]), ('features', 'NNS'),
Tree('PP', [Tree('P', [('of', 'IN')]), Tree('NP', [('the', 'DT'), ('iPad', 'NN')])]),
('and', 'CC'), Tree('NP', [('iPhone', 'NN')]), Tree('P', [('in', 'IN')]), ('its', 'PRP
$'), ('Galaxy', 'NNP'), Tree('NP', [('range', 'NN')]), Tree('P', [('of', 'IN')]), ('de
vices', 'NNS'), ('.', '.')]), Tree('S', [('Samsung', 'NNP'), (',', ','), ('which', 'WD
T'), Tree('VP', [Tree('V', [('is', 'VBZ')]), Tree('NP', [('the', 'DT'), ('world', 'N
N')])]), ("'s", 'POS'), Tree('NP', [('top', 'JJ'), ('mobile', 'NN'), ('phone', 'NN'),
('maker', 'NN')]), (',', ','), Tree('VP', [Tree('V', [('is', 'VBZ')])]), Tree('VP', [T
ree('V', [('appealing', 'VBG')]), Tree('NP', [('the', 'DT'), ('ruling', 'NN')])]),
('.', '.')]), Tree('S', [Tree('NP', [('A', 'DT'), ('similar', 'JJ'), ('case', 'NN')]),
Tree('PP', [Tree('P', [('in', 'IN')]), Tree('NP', [('the', 'DT')])]), ('UK', 'NNP'), T
ree('VP', [Tree('V', [('found', 'VBD')])]), Tree('P', [('in', 'IN')]), ('Samsung', 'NN
P'), ("'s", 'POS'), Tree('NP', [('favour', 'NN')]), ('and', 'CC'), Tree('VP', [Tree
('V', [('ordered', 'VBD')])]), ('Apple', 'NNP'), ('to', 'TO'), Tree('VP', [Tree('V',
[('publish', 'VB')]), Tree('NP', [('an', 'DT'), ('apology', 'NN')])]), Tree('VP', [Tre
e('V', [('making', 'VBG')]), Tree('NP', [('clear', 'JJ')]), Tree('PP', [Tree('P', [('t
hat', 'IN')]), Tree('NP', [('the', 'DT'), ('South', 'JJ'), ('Korean', 'JJ'), ('firm',
'NN')])])]), Tree('VP', [Tree('V', [('had', 'VBD')])]), ('not', 'RB'), Tree('VP', [Tre
e('V', [('copied', 'VBN')])]), ('its', 'PRP$'), Tree('NP', [('iPad', 'NN')]), ('when',
'WRB'), Tree('VP', [Tree('V', [('designing', 'VBG')])]), ('its', 'PRP$'), Tree('NP',
[('own', 'JJ')]), ('devices', 'NNS'), ('.', '.')])]
```

[total points: 1] Exercise 2: spaCy

Use Spacy to process the same text as you analyzed with NLTK.

In [19]:

```python
import spacy
nlp = spacy.load('en_core_web_sm')
```

In [20]:

```python
doc = nlp(text)

# insert code here
sents = list(doc.sents)
for i, sentence in enumerate(sents):
    print('Sentence', i+1, ':', sentence.text)
```

```
Sentence 1 : https://www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsui
t-six-more-products-under-scrutiny.html

Documents filed to the San Jose federal court in California on November 23 list six Sa
msung products running the "Jelly Bean" and "Ice Cream Sandwich" operating systems, wh
ich Apple claims infringe its patents.

Sentence 2 : The six phones and tablets affected are the Galaxy S III, running the new
Jelly Bean system, the Galaxy Tab 8.9 Wifi tablet, the Galaxy Tab 2 10.1, Galaxy Rugby
Pro and Galaxy S III mini.

Sentence 3 : Apple stated it had “acted quickly and diligently" in order to "determine
that these newly released products do infringe many of the same claims already asserte
d by Apple.
Sentence 4 : "
In August, Samsung lost a US patent case to Apple and was ordered to pay its rival $1.
05bn (£0.66bn) in damages for copying features of the iPad and iPhone in its Galaxy ra
nge of devices.
Sentence 5 : Samsung, which is the world's top mobile phone maker, is appealing the ru
ling.

Sentence 6 : A similar case in the UK found in Samsung's favour and ordered Apple to p
ublish an apology making clear that the South Korean firm had not copied its iPad when
designing its own devices.
```

In [21]:

```python
# POS tagging
pos_tags = [(token.text, token.pos_) for token in doc]
print("POS tagging:")
print(pos_tags)
print()

# NER
ner_results = [(ent.text, ent.label_) for ent in doc.ents]
print("NER:")
print(ner_results)
print()

# constituency parsing (syntactic dependency parsing using the `dep_`)
constituency_parsing = [(token.text, token.dep_) for token in doc]
print("Constituency Parsing:")
print(constituency_parsing)
```

```
POS tagging:
[('https://www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more
-products-under-scrutiny.html', 'NOUN'), ('\n\n', 'SPACE'), ('Documents', 'NOUN'), ('f
iled', 'VERB'), ('to', 'ADP'), ('the', 'DET'), ('San', 'PROPN'), ('Jose', 'PROPN'),
('federal', 'ADJ'), ('court', 'NOUN'), ('in', 'ADP'), ('California', 'PROPN'), ('on',
'ADP'), ('November', 'PROPN'), ('23', 'NUM'), ('list', 'NOUN'), ('six', 'NUM'), ('Sams
ung', 'PROPN'), ('products', 'NOUN'), ('running', 'VERB'), ('the', 'DET'), ('"', 'PUNC
T'), ('Jelly', 'PROPN'), ('Bean', 'PROPN'), ('"', 'PUNCT'), ('and', 'CCONJ'), ('"', 'P
UNCT'), ('Ice', 'PROPN'), ('Cream', 'PROPN'), ('Sandwich', 'NOUN'), ('"', 'PUNCT'),
('operating', 'NOUN'), ('systems', 'NOUN'), (',', 'PUNCT'), ('which', 'PRON'), ('Appl
e', 'PROPN'), ('claims', 'VERB'), ('infringe', 'VERB'), ('its', 'PRON'), ('patents',
'NOUN'), ('.', 'PUNCT'), ('\n', 'SPACE'), ('The', 'DET'), ('six', 'NUM'), ('phones',
'NOUN'), ('and', 'CCONJ'), ('tablets', 'NOUN'), ('affected', 'VERB'), ('are', 'AUX'),
('the', 'DET'), ('Galaxy', 'PROPN'), ('S', 'PROPN'), ('III', 'PROPN'), (',', 'PUNCT'),
('running', 'VERB'), ('the', 'DET'), ('new', 'ADJ'), ('Jelly', 'PROPN'), ('Bean', 'PRO
PN'), ('system', 'NOUN'), (',', 'PUNCT'), ('the', 'DET'), ('Galaxy', 'PROPN'), ('Tab',
'PROPN'), ('8.9', 'NUM'), ('Wifi', 'PROPN'), ('tablet', 'NOUN'), (',', 'PUNCT'), ('th
e', 'DET'), ('Galaxy', 'PROPN'), ('Tab', 'PROPN'), ('2', 'NUM'), ('10.1', 'NUM'),
(',', 'PUNCT'), ('Galaxy', 'PROPN'), ('Rugby', 'PROPN'), ('Pro', 'PROPN'), ('and', 'CC
ONJ'), ('Galaxy', 'PROPN'), ('S', 'PROPN'), ('III', 'PROPN'), ('mini', 'NOUN'), ('.',
'PUNCT'), ('\n', 'SPACE'), ('Apple', 'PROPN'), ('stated', 'VERB'), ('it', 'PRON'), ('h
ad', 'AUX'), ('“', 'PUNCT'), ('acted', 'VERB'), ('quickly', 'ADV'), ('and', 'CCONJ'),
('diligently', 'ADV'), ('"', 'PUNCT'), ('in', 'ADP'), ('order', 'NOUN'), ('to', 'PAR
T'), ('"', 'PUNCT'), ('determine', 'VERB'), ('that', 'SCONJ'), ('these', 'DET'), ('new
ly', 'ADV'), ('released', 'VERB'), ('products', 'NOUN'), ('do', 'AUX'), ('infringe',
'VERB'), ('many', 'ADJ'), ('of', 'ADP'), ('the', 'DET'), ('same', 'ADJ'), ('claims',
'NOUN'), ('already', 'ADV'), ('asserted', 'VERB'), ('by', 'ADP'), ('Apple', 'PROPN'),
('.', 'PUNCT'), ('"', 'PUNCT'), ('\n', 'SPACE'), ('In', 'ADP'), ('August', 'PROPN'),
(',', 'PUNCT'), ('Samsung', 'PROPN'), ('lost', 'VERB'), ('a', 'DET'), ('US', 'PROPN'),
('patent', 'NOUN'), ('case', 'NOUN'), ('to', 'ADP'), ('Apple', 'PROPN'), ('and', 'CCON
J'), ('was', 'AUX'), ('ordered', 'VERB'), ('to', 'PART'), ('pay', 'VERB'), ('its', 'PR
ON'), ('rival', 'NOUN'), ('$', 'SYM'), ('1.05bn', 'NUM'), ('(', 'PUNCT'), ('£', 'SY
M'), ('0.66bn', 'NOUN'), (')', 'PUNCT'), ('in', 'ADP'), ('damages', 'NOUN'), ('for',
'ADP'), ('copying', 'VERB'), ('features', 'NOUN'), ('of', 'ADP'), ('the', 'DET'), ('iP
ad', 'PROPN'), ('and', 'CCONJ'), ('iPhone', 'PROPN'), ('in', 'ADP'), ('its', 'PRON'),
('Galaxy', 'PROPN'), ('range', 'NOUN'), ('of', 'ADP'), ('devices', 'NOUN'), ('.', 'PUN
CT'), ('Samsung', 'PROPN'), (',', 'PUNCT'), ('which', 'PRON'), ('is', 'AUX'), ('the',
'DET'), ('world', 'NOUN'), ("'s", 'PART'), ('top', 'ADJ'), ('mobile', 'ADJ'), ('phon
e', 'NOUN'), ('maker', 'NOUN'), (',', 'PUNCT'), ('is', 'AUX'), ('appealing', 'VERB'),
('the', 'DET'), ('ruling', 'NOUN'), ('.', 'PUNCT'), ('\n', 'SPACE'), ('A', 'DET'), ('s
imilar', 'ADJ'), ('case', 'NOUN'), ('in', 'ADP'), ('the', 'DET'), ('UK', 'PROPN'), ('f
ound', 'VERB'), ('in', 'ADP'), ('Samsung', 'PROPN'), ("'s", 'PART'), ('favour', 'NOU
N'), ('and', 'CCONJ'), ('ordered', 'VERB'), ('Apple', 'PROPN'), ('to', 'PART'), ('publ
ish', 'VERB'), ('an', 'DET'), ('apology', 'NOUN'), ('making', 'VERB'), ('clear', 'AD
J'), ('that', 'SCONJ'), ('the', 'DET'), ('South', 'ADJ'), ('Korean', 'ADJ'), ('firm',
'NOUN'), ('had', 'AUX'), ('not', 'PART'), ('copied', 'VERB'), ('its', 'PRON'), ('iPa
d', 'PROPN'), ('when', 'SCONJ'), ('designing', 'VERB'), ('its', 'PRON'), ('own', 'AD
J'), ('devices', 'NOUN'), ('.', 'PUNCT')]

NER:
[('https://www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more
-products-under-scrutiny.html', 'TIME'), ('San Jose', 'GPE'), ('California', 'GPE'),
('November 23', 'DATE'), ('six', 'CARDINAL'), ('Samsung', 'ORG'), ('the "Jelly Bean',
'LAW'), ('Apple', 'ORG'), ('six', 'CARDINAL'), ('the Galaxy S III', 'ORG'), ('Jelly Be
an', 'ORG'), ('8.9', 'CARDINAL'), ('2 10.1', 'DATE'), ('Galaxy Rugby Pro', 'ORG'), ('G
alaxy S III', 'PERSON'), ('Apple', 'ORG'), ('Apple', 'ORG'), ('August', 'DATE'), ('Sam
sung', 'ORG'), ('US', 'GPE'), ('Apple', 'ORG'), ('1.05bn', 'MONEY'), ('0.66bn', 'MONE
Y'), ('iPad', 'ORG'), ('Galaxy', 'FAC'), ('Samsung', 'ORG'), ('UK', 'GPE'), ('Samsun
g', 'ORG'), ('Apple', 'ORG'), ('South Korean', 'NORP'), ('iPad', 'ORG')]

Constituency Parsing:
[('https://www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more
-products-under-scrutiny.html', 'amod'), ('\n\n', 'dep'), ('Documents', 'nsubj'), ('fi
led', 'ROOT'), ('to', 'prep'), ('the', 'det'), ('San', 'nmod'), ('Jose', 'nmod'), ('fe
deral', 'amod'), ('court', 'pobj'), ('in', 'prep'), ('California', 'pobj'), ('on', 'pr
ep'), ('November', 'pobj'), ('23', 'nummod'), ('list', 'compound'), ('six', 'nummod'),
('Samsung', 'compound'), ('products', 'dobj'), ('running', 'acl'), ('the', 'det'),
('"', 'punct'), ('Jelly', 'compound'), ('Bean', 'dobj'), ('"', 'punct'), ('and', 'c
c'), ('"', 'punct'), ('Ice', 'compound'), ('Cream', 'compound'), ('Sandwich', 'nmod'),
('"', 'punct'), ('operating', 'compound'), ('systems', 'conj'), (',', 'punct'), ('whic
h', 'nsubj'), ('Apple', 'compound'), ('claims', 'nsubj'), ('infringe', 'relcl'), ('it
s', 'poss'), ('patents', 'dobj'), ('.', 'punct'), ('\n', 'dep'), ('The', 'det'), ('si
x', 'nummod'), ('phones', 'nsubj'), ('and', 'cc'), ('tablets', 'conj'), ('affected',
'acl'), ('are', 'ROOT'), ('the', 'det'), ('Galaxy', 'compound'), ('S', 'compound'),
('III', 'attr'), (',', 'punct'), ('running', 'advcl'), ('the', 'det'), ('new', 'amo
d'), ('Jelly', 'compound'), ('Bean', 'compound'), ('system', 'dobj'), (',', 'punct'),
('the', 'det'), ('Galaxy', 'compound'), ('Tab', 'nmod'), ('8.9', 'nummod'), ('Wifi',
'compound'), ('tablet', 'appos'), (',', 'punct'), ('the', 'det'), ('Galaxy', 'compoun
d'), ('Tab', 'conj'), ('2', 'compound'), ('10.1', 'nummod'), (',', 'punct'), ('Galax
y', 'compound'), ('Rugby', 'compound'), ('Pro', 'conj'), ('and', 'cc'), ('Galaxy', 'co
mpound'), ('S', 'compound'), ('III', 'conj'), ('mini', 'appos'), ('.', 'punct'),
('\n', 'dep'), ('Apple', 'nsubj'), ('stated', 'ROOT'), ('it', 'nsubj'), ('had', 'au
x'), ('“', 'punct'), ('acted', 'ccomp'), ('quickly', 'advmod'), ('and', 'cc'), ('dilig
ently', 'conj'), ('"', 'punct'), ('in', 'prep'), ('order', 'pobj'), ('to', 'aux'),
('"', 'punct'), ('determine', 'acl'), ('that', 'mark'), ('these', 'det'), ('newly', 'a
dvmod'), ('released', 'amod'), ('products', 'nsubj'), ('do', 'aux'), ('infringe', 'cco
mp'), ('many', 'dobj'), ('of', 'prep'), ('the', 'det'), ('same', 'amod'), ('claims',
'pobj'), ('already', 'advmod'), ('asserted', 'acl'), ('by', 'agent'), ('Apple', 'pob
j'), ('.', 'punct'), ('"', 'punct'), ('\n', 'dep'), ('In', 'prep'), ('August', 'pob
j'), (',', 'punct'), ('Samsung', 'nsubj'), ('lost', 'ROOT'), ('a', 'det'), ('US', 'com
pound'), ('patent', 'compound'), ('case', 'dobj'), ('to', 'prep'), ('Apple', 'pobj'),
('and', 'cc'), ('was', 'auxpass'), ('ordered', 'conj'), ('to', 'aux'), ('pay', 'xcom
p'), ('its', 'poss'), ('rival', 'dative'), ('$', 'nmod'), ('1.05bn', 'dobj'), ('(', 'p
unct'), ('£', 'nmod'), ('0.66bn', 'appos'), (')', 'punct'), ('in', 'prep'), ('damage
s', 'pobj'), ('for', 'prep'), ('copying', 'pcomp'), ('features', 'dobj'), ('of', 'pre
p'), ('the', 'det'), ('iPad', 'pobj'), ('and', 'cc'), ('iPhone', 'conj'), ('in', 'pre
p'), ('its', 'poss'), ('Galaxy', 'compound'), ('range', 'pobj'), ('of', 'prep'), ('dev
ices', 'pobj'), ('.', 'punct'), ('Samsung', 'nsubj'), (',', 'punct'), ('which', 'nsub
j'), ('is', 'relcl'), ('the', 'det'), ('world', 'poss'), ("'s", 'case'), ('top', 'amo
d'), ('mobile', 'amod'), ('phone', 'compound'), ('maker', 'attr'), (',', 'punct'), ('i
s', 'aux'), ('appealing', 'ROOT'), ('the', 'det'), ('ruling', 'dobj'), ('.', 'punct'),
('\n', 'dep'), ('A', 'det'), ('similar', 'amod'), ('case', 'nsubj'), ('in', 'prep'),
('the', 'det'), ('UK', 'pobj'), ('found', 'ROOT'), ('in', 'prep'), ('Samsung', 'pos
s'), ("'s", 'case'), ('favour', 'pobj'), ('and', 'cc'), ('ordered', 'conj'), ('Apple',
'dobj'), ('to', 'aux'), ('publish', 'xcomp'), ('an', 'det'), ('apology', 'dobj'), ('ma
king', 'acl'), ('clear', 'acomp'), ('that', 'mark'), ('the', 'det'), ('South', 'amo
d'), ('Korean', 'amod'), ('firm', 'nsubj'), ('had', 'aux'), ('not', 'neg'), ('copied',
'ccomp'), ('its', 'poss'), ('iPad', 'dobj'), ('when', 'advmod'), ('designing', 'advc
l'), ('its', 'poss'), ('own', 'amod'), ('devices', 'dobj'), ('.', 'punct')]
```

small tip: You can use **sents = list(doc.sents)** to be able to use the index to access a sentence like **sents[2]** for the third sentence.

[total points: 7] Exercise 3: Comparison NLTK and spaCy

We will now compare the output of NLTK and spaCy, i.e., in what do they differ?

[points: 3] Exercise 3a: Part of speech tagging

Compare the output from NLTK and spaCy regarding part of speech tagging.

In [22]:
```
nltk_sentences = sent_tokenize(text)
print("NLTK Sentences:")
print(nltk_sentences)

# spaCy sentence splitting
nlp = spacy.load("en_core_web_sm")
doc = nlp(text)
spacy_sentences = [sent.text for sent in doc.sents]
print("spaCy Sentences:")
print(spacy_sentences)
```

```
NLTK Sentences:
['https://www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more-
products-under-scrutiny.html\n\nDocuments filed to the San Jose federal court in Calif
ornia on November 23 list six Samsung products running the "Jelly Bean" and "Ice Cream
Sandwich" operating systems, which Apple claims infringe its patents.', 'The six phone
s and tablets affected are the Galaxy S III, running the new Jelly Bean system, the Ga
laxy Tab 8.9 Wifi tablet, the Galaxy Tab 2 10.1, Galaxy Rugby Pro and Galaxy S III min
i.', 'Apple stated it had “acted quickly and diligently" in order to "determine that t
hese newly released products do infringe many of the same claims already asserted by A
pple."', 'In August, Samsung lost a US patent case to Apple and was ordered to pay its
rival $1.05bn (£0.66bn) in damages for copying features of the iPad and iPhone in its
Galaxy range of devices.', "Samsung, which is the world's top mobile phone maker, is a
ppealing the ruling.", "A similar case in the UK found in Samsung's favour and ordered
Apple to publish an apology making clear that the South Korean firm had not copied its
iPad when designing its own devices."]
spaCy Sentences:
['https://www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more-
products-under-scrutiny.html\n\nDocuments filed to the San Jose federal court in Calif
ornia on November 23 list six Samsung products running the "Jelly Bean" and "Ice Cream
Sandwich" operating systems, which Apple claims infringe its patents.\n', 'The six pho
nes and tablets affected are the Galaxy S III, running the new Jelly Bean system, the
Galaxy Tab 8.9 Wifi tablet, the Galaxy Tab 2 10.1, Galaxy Rugby Pro and Galaxy S III m
ini.\n', 'Apple stated it had “acted quickly and diligently" in order to "determine th
at these newly released products do infringe many of the same claims already asserted
by Apple.', '"\nIn August, Samsung lost a US patent case to Apple and was ordered to p
ay its rival $1.05bn (£0.66bn) in damages for copying features of the iPad and iPhone
in its Galaxy range of devices.', "Samsung, which is the world's top mobile phone make
r, is appealing the ruling.\n", "A similar case in the UK found in Samsung's favour an
d ordered Apple to publish an apology making clear that the South Korean firm had not
copied its iPad when designing its own devices."]
```

```
In [23]: print(pos_tags_per_sentence)
         print(pos_tags)
```

```
[[('https', 'NN'), (':', ':'), ('//www.telegraph.co.uk/technology/apple/9702716/Apple-
Samsung-lawsuit-six-more-products-under-scrutiny.html', 'JJ'), ('Documents', 'NNS'),
('filed', 'VBN'), ('to', 'TO'), ('the', 'DT'), ('San', 'NNP'), ('Jose', 'NNP'), ('fede
ral', 'JJ'), ('court', 'NN'), ('in', 'IN'), ('California', 'NNP'), ('on', 'IN'), ('Nov
ember', 'NNP'), ('23', 'CD'), ('list', 'NN'), ('six', 'CD'), ('Samsung', 'NNP'), ('pro
ducts', 'NNS'), ('running', 'VBG'), ('the', 'DT'), ('``', '``'), ('Jelly', 'RB'), ('Be
an', 'NNP'), ("''", "''"), ('and', 'CC'), ('``', '``'), ('Ice', 'NNP'), ('Cream', 'NN
P'), ('Sandwich', 'NNP'), ("''", "''"), ('operating', 'VBG'), ('systems', 'NNS'),
(',', ','), ('which', 'WDT'), ('Apple', 'NNP'), ('claims', 'VBZ'), ('infringe', 'VB'),
('its', 'PRP$'), ('patents', 'NNS'), ('.', '.')], [('The', 'DT'), ('six', 'CD'), ('pho
nes', 'NNS'), ('and', 'CC'), ('tablets', 'NNS'), ('affected', 'VBN'), ('are', 'VBP'),
('the', 'DT'), ('Galaxy', 'NNP'), ('S', 'NNP'), ('III', 'NNP'), (',', ','), ('runnin
g', 'VBG'), ('the', 'DT'), ('new', 'JJ'), ('Jelly', 'NNP'), ('Bean', 'NNP'), ('syste
m', 'NN'), (',', ','), ('the', 'DT'), ('Galaxy', 'NNP'), ('Tab', 'NNP'), ('8.9', 'C
D'), ('Wifi', 'NNP'), ('tablet', 'NN'), (',', ','), ('the', 'DT'), ('Galaxy', 'NNP'),
('Tab', 'NNP'), ('2', 'CD'), ('10.1', 'CD'), (',', ','), ('Galaxy', 'NNP'), ('Rugby',
'NNP'), ('Pro', 'NNP'), ('and', 'CC'), ('Galaxy', 'NNP'), ('S', 'NNP'), ('III', 'NN
P'), ('mini', 'NN'), ('.', '.')], [('Apple', 'NNP'), ('stated', 'VBD'), ('it', 'PRP'),
('had', 'VBD'), ('“', 'NNP'), ('acted', 'VBD'), ('quickly', 'RB'), ('and', 'CC'), ('di
ligently', 'RB'), ("''", "''"), ('in', 'IN'), ('order', 'NN'), ('to', 'TO'), ('``', '`
`'), ('determine', 'VB'), ('that', 'IN'), ('these', 'DT'), ('newly', 'RB'), ('release
d', 'VBN'), ('products', 'NNS'), ('do', 'VBP'), ('infringe', 'VB'), ('many', 'JJ'),
('of', 'IN'), ('the', 'DT'), ('same', 'JJ'), ('claims', 'NNS'), ('already', 'RB'), ('a
sserted', 'VBN'), ('by', 'IN'), ('Apple', 'NNP'), ('.', '.'), ("''", "''")], [('In',
'IN'), ('August', 'NNP'), (',', ','), ('Samsung', 'NNP'), ('lost', 'VBD'), ('a', 'D
T'), ('US', 'NNP'), ('patent', 'NN'), ('case', 'NN'), ('to', 'TO'), ('Apple', 'NNP'),
('and', 'CC'), ('was', 'VBD'), ('ordered', 'VBN'), ('to', 'TO'), ('pay', 'VB'), ('it
s', 'PRP$'), ('rival', 'JJ'), ('$', '$'), ('1.05bn', 'CD'), ('(', '('), ('£0.66bn', 'N
N'), (')', ')'), ('in', 'IN'), ('damages', 'NNS'), ('for', 'IN'), ('copying', 'VBG'),
('features', 'NNS'), ('of', 'IN'), ('the', 'DT'), ('iPad', 'NN'), ('and', 'CC'), ('iPh
one', 'NN'), ('in', 'IN'), ('its', 'PRP$'), ('Galaxy', 'NNP'), ('range', 'NN'), ('of',
'IN'), ('devices', 'NNS'), ('.', '.')], [('Samsung', 'NNP'), (',', ','), ('which', 'WD
T'), ('is', 'VBZ'), ('the', 'DT'), ('world', 'NN'), ("'s", 'POS'), ('top', 'JJ'), ('mo
bile', 'NN'), ('phone', 'NN'), ('maker', 'NN'), (',', ','), ('is', 'VBZ'), ('appealin
g', 'VBG'), ('the', 'DT'), ('ruling', 'NN'), ('.', '.')], [('A', 'DT'), ('similar', 'J
J'), ('case', 'NN'), ('in', 'IN'), ('the', 'DT'), ('UK', 'NNP'), ('found', 'VBD'), ('i
n', 'IN'), ('Samsung', 'NNP'), ("'s", 'POS'), ('favour', 'NN'), ('and', 'CC'), ('order
ed', 'VBD'), ('Apple', 'NNP'), ('to', 'TO'), ('publish', 'VB'), ('an', 'DT'), ('apolog
y', 'NN'), ('making', 'VBG'), ('clear', 'JJ'), ('that', 'IN'), ('the', 'DT'), ('Sout
h', 'JJ'), ('Korean', 'JJ'), ('firm', 'NN'), ('had', 'VBD'), ('not', 'RB'), ('copied',
'VBN'), ('its', 'PRP$'), ('iPad', 'NN'), ('when', 'WRB'), ('designing', 'VBG'), ('it
s', 'PRP$'), ('own', 'JJ'), ('devices', 'NNS'), ('.', '.')]]
[('https://www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more
-products-under-scrutiny.html', 'NOUN'), ('\n\n', 'SPACE'), ('Documents', 'NOUN'), ('f
iled', 'VERB'), ('to', 'ADP'), ('the', 'DET'), ('San', 'PROPN'), ('Jose', 'PROPN'),
('federal', 'ADJ'), ('court', 'NOUN'), ('in', 'ADP'), ('California', 'PROPN'), ('on',
'ADP'), ('November', 'PROPN'), ('23', 'NUM'), ('list', 'NOUN'), ('six', 'NUM'), ('Sams
ung', 'PROPN'), ('products', 'NOUN'), ('running', 'VERB'), ('the', 'DET'), ('"', 'PUNC
T'), ('Jelly', 'PROPN'), ('Bean', 'PROPN'), ('"', 'PUNCT'), ('and', 'CCONJ'), ('"', 'P
UNCT'), ('Ice', 'PROPN'), ('Cream', 'PROPN'), ('Sandwich', 'NOUN'), ('"', 'PUNCT'),
('operating', 'NOUN'), ('systems', 'NOUN'), (',', 'PUNCT'), ('which', 'PRON'), ('Appl
e', 'PROPN'), ('claims', 'VERB'), ('infringe', 'VERB'), ('its', 'PRON'), ('patents',
'NOUN'), ('.', 'PUNCT'), ('\n', 'SPACE'), ('The', 'DET'), ('six', 'NUM'), ('phones',
'NOUN'), ('and', 'CCONJ'), ('tablets', 'NOUN'), ('affected', 'VERB'), ('are', 'AUX'),
('the', 'DET'), ('Galaxy', 'PROPN'), ('S', 'PROPN'), ('III', 'PROPN'), (',', 'PUNCT'),
('running', 'VERB'), ('the', 'DET'), ('new', 'ADJ'), ('Jelly', 'PROPN'), ('Bean', 'PRO
PN'), ('system', 'NOUN'), (',', 'PUNCT'), ('the', 'DET'), ('Galaxy', 'PROPN'), ('Tab',
'PROPN'), ('8.9', 'NUM'), ('Wifi', 'PROPN'), ('tablet', 'NOUN'), (',', 'PUNCT'), ('th
e', 'DET'), ('Galaxy', 'PROPN'), ('Tab', 'PROPN'), ('2', 'NUM'), ('10.1', 'NUM'),
(',', 'PUNCT'), ('Galaxy', 'PROPN'), ('Rugby', 'PROPN'), ('Pro', 'PROPN'), ('and', 'CC
ONJ'), ('Galaxy', 'PROPN'), ('S', 'PROPN'), ('III', 'PROPN'), ('mini', 'NOUN'), ('.',
'PUNCT'), ('\n', 'SPACE'), ('Apple', 'PROPN'), ('stated', 'VERB'), ('it', 'PRON'), ('h
ad', 'AUX'), ('“', 'PUNCT'), ('acted', 'VERB'), ('quickly', 'ADV'), ('and', 'CCONJ'),
('diligently', 'ADV'), ('"', 'PUNCT'), ('in', 'ADP'), ('order', 'NOUN'), ('to', 'PAR
T'), ('"', 'PUNCT'), ('determine', 'VERB'), ('that', 'SCONJ'), ('these', 'DET'), ('new
ly', 'ADV'), ('released', 'VERB'), ('products', 'NOUN'), ('do', 'AUX'), ('infringe',
'VERB'), ('many', 'ADJ'), ('of', 'ADP'), ('the', 'DET'), ('same', 'ADJ'), ('claims',
'NOUN'), ('already', 'ADV'), ('asserted', 'VERB'), ('by', 'ADP'), ('Apple', 'PROPN'),
('.', 'PUNCT'), ('"', 'PUNCT'), ('\n', 'SPACE'), ('In', 'ADP'), ('August', 'PROPN'),
(',', 'PUNCT'), ('Samsung', 'PROPN'), ('lost', 'VERB'), ('a', 'DET'), ('US', 'PROPN'),
('patent', 'NOUN'), ('case', 'NOUN'), ('to', 'ADP'), ('Apple', 'PROPN'), ('and', 'CCON
J'), ('was', 'AUX'), ('ordered', 'VERB'), ('to', 'PART'), ('pay', 'VERB'), ('its', 'PR
ON'), ('rival', 'NOUN'), ('$', 'SYM'), ('1.05bn', 'NUM'), ('(', 'PUNCT'), ('£', 'SY
M'), ('0.66bn', 'NOUN'), (')', 'PUNCT'), ('in', 'ADP'), ('damages', 'NOUN'), ('for',
'ADP'), ('copying', 'VERB'), ('features', 'NOUN'), ('of', 'ADP'), ('the', 'DET'), ('iP
ad', 'PROPN'), ('and', 'CCONJ'), ('iPhone', 'PROPN'), ('in', 'ADP'), ('its', 'PRON'),
('Galaxy', 'PROPN'), ('range', 'NOUN'), ('of', 'ADP'), ('devices', 'NOUN'), ('.', 'PUN
CT'), ('Samsung', 'PROPN'), (',', 'PUNCT'), ('which', 'PRON'), ('is', 'AUX'), ('the',
'DET'), ('world', 'NOUN'), ("'s", 'PART'), ('top', 'ADJ'), ('mobile', 'ADJ'), ('phon
e', 'NOUN'), ('maker', 'NOUN'), (',', 'PUNCT'), ('is', 'AUX'), ('appealing', 'VERB'),
('the', 'DET'), ('ruling', 'NOUN'), ('.', 'PUNCT'), ('\n', 'SPACE'), ('A', 'DET'), ('s
imilar', 'ADJ'), ('case', 'NOUN'), ('in', 'ADP'), ('the', 'DET'), ('UK', 'PROPN'), ('f
ound', 'VERB'), ('in', 'ADP'), ('Samsung', 'PROPN'), ("'s", 'PART'), ('favour', 'NOU
N'), ('and', 'CCONJ'), ('ordered', 'VERB'), ('Apple', 'PROPN'), ('to', 'PART'), ('publ
ish', 'VERB'), ('an', 'DET'), ('apology', 'NOUN'), ('making', 'VERB'), ('clear', 'AD
J'), ('that', 'SCONJ'), ('the', 'DET'), ('South', 'ADJ'), ('Korean', 'ADJ'), ('firm',
'NOUN'), ('had', 'AUX'), ('not', 'PART'), ('copied', 'VERB'), ('its', 'PRON'), ('iPa
d', 'PROPN'), ('when', 'SCONJ'), ('designing', 'VERB'), ('its', 'PRON'), ('own', 'AD
J'), ('devices', 'NOUN'), ('.', 'PUNCT')]
```

- **To compare, you probably would like to compare sentence per sentence. Describe if the sentence splitting is different for NLTK than for spaCy. If not, where do they differ?**

Answer: Both NLTK and spaCy correctly split the text into sentences but the way it splits them is different. For SpaCy, the sentence splitting appears to be more straightforward, as it treats each newline character (\n) as a separate sentence boundary while the sentences are split based on more sophisticated rules in NLTK without the newline character. In both, the URL link is not considered as a separate token, effectively ignoring punctuations within the URL.

- **After checking the sentence splitting, select a sentence for which you expect interesting results and perhaps differences. Motivate your choice.**

Answer: I choose the sentence "The six phones and tablets affected are the Galaxy S III, running the new Jelly Bean system, the Galaxy Tab 8.9 Wifi tablet, the Galaxy Tab 2 10.1, Galaxy Rugby Pro and Galaxy S III mini." because it contains several product names ("Galaxy S III", "Jelly Bean system", "Galaxy Tab 8.9 Wifi tablet"...) and their descriptions, which could present interesting results, showing differences in POS tagging between NLTK and spaCy.

- **Compare the output in `token.tag` from spaCy to the part of speech tagging from NLTK for each token in your selected sentence. Are there any differences? This is not a trick question; it is possible that there are no differences.**

Answer: When comparing the POS tagging output from NLTK and spaCy for each token in the selected sentence, we observe differences in the tagging results. Outside of the sentence chosen, we can see that the url for NLTK is actually split into two tokens while SpaCy maintains the full link as one entity. Already we can see differences and when looking at the dedicated sentence, the word "phones" is identified as NNS (plural noun) in NLTK while in SpaCy it's tagged only as a NOUN which would suggest that NLTK is a more complex system. However, in the word "are" NLTK tagges it as a VERB while SpaCy goes further to identify it as a AUX (or auxiliary verb), a more fitting tag for the word in question. In the phrase "Galaxy S III", both NLTK and SpaCY tag it as a proper noun.

Similarly, in the phrase "Jelly Bean system", both spaCy and NLTK correctly identify "Jelly" and "Bean" as part of a proper noun (PROPN). The last difference is in notation where SpaCy tags punctuations aas (PUNCT) while NLTK maintains their original form. Both are effective and correct in most parts and significant differences were not observed as both strategies demonstrate complexity in different areas.

[points: 2] Exercise 3b: Named Entity Recognition (NER)

- Describe differences between the output from NLTK and spaCy for Named Entity Recognition. Which one do you think performs better?

In general, spaCY shows to perform more accurately and faster than NLTK. For instance, NLTK labels Apple, Galaxy and Samsung as PERSONS instead of ORGANISATION. In addition it also sometimes labels Apple as GPE when it is not a location. spaCy however, is consistent with correct answers, labelling San Jose and California as GPE and Samsung, Apple and Galaxy S III as ORG.

[points: 2] Exercise 3c: Constituency/dependency parsing

Choose one sentence from the text and run constituency parsing using NLTK and dependency parsing using spaCy.

- describe briefly the difference between constituency parsing and dependency parsing
- describe differences between the output from NLTK and spaCy.

Constituency and dependency parsing are common approaches used to analyse the grammatical structure of sentences during natural language processing. Firstly, constituency parsing breaks down a sentence into smaller fragments of phrases and creates a hierarchical tree structure where each node (phrase) is labelled with a grammatical category. Meanwhile dependency parsing aims to identify the relationships between words in a sentence in terms of their directed dependencies. It is represented using a directed graph such that each word is a node connected by direct edges, thus indicating the grammatical relationships between the words.

In regards to constituency parsing, NLTK will output nested tree structures where each node is phrase labelled by its grammatical category. However, spaCy will undergo dependency parsing in which it consists of a ‘doc’ object that contains the parsed sentence alongside its dependency relation.

End of this notebook