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 whic
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))
```

```
/home/ai/Downloads/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 [3]:

```
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 [4]:

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

In [5]:

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

In [6]:

```
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 [7]:

```
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 t
he new Jelly Bean system, the Galaxy Tab 8.9 Wifi tablet, the Galaxy Tab 2 1
0.1, Galaxy Rugby Pro and Galaxy S III mini.
TOKENS ['The', 'six', 'phones', 'and', 'tablets', 'affected', 'are', 'the',
'Galaxy', 'S', 'III', ',', 'running', 'the', 'new', 'Jelly', 'Bean', 'syste
m', ',', 'the', 'Galaxy', 'Tab', '8.9', 'Wifi', 'tablet', ',', 'the', 'Galax
y', 'Tab', '2', '10.1', ',', 'Galaxy', '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 [8]:

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

```
[('https', 'NN'), (':', ':'), ('//www.telegraph.co.uk/technology/apple/97027
16/Apple-Samsung-lawsuit-six-more-products-under-scrutiny.html', 'JJ'), ('Do
cuments', 'NNS'), ('filed', 'VBN'), ('to', 'TO'), ('the', 'DT'), ('San', 'NN
P'), ('Jose', 'NNP'), ('federal', 'JJ'), ('court', 'NN'), ('in', 'IN'), ('Ca
lifornia', 'NNP'), ('on', 'IN'), ('November', 'NNP'), ('23', 'CD'), ('list',
'NN'), ('six', 'CD'), ('Samsung', 'NNP'), ('products', 'NNS'), ('running',
'VBG'), ('the', 'DT'), ('``', '``'), ('Jelly', 'RB'), ('Bean', 'NNP'),
("''", "''"), ('and', 'CC'), ('``', '``'), ('Ice', 'NNP'), ('Cream', 'NNP'),
('Sandwich', 'NNP'), ("''", "''"), ('operating', 'VBG'), ('systems', 'NNS'),
(',', ','), ('which', 'WDT'), ('Apple', 'NNP'), ('claims', 'VBZ'), ('infring
e', 'VB'), ('its', 'PRP$'), ('patents', 'NNS'), ('.', '.')]
------------------------
[('The', 'DT'), ('six', 'CD'), ('phones', 'NNS'), ('and', 'CC'), ('tablets',
'NNS'), ('affected', 'VBN'), ('are', 'VBP'), ('the', 'DT'), ('Galaxy', 'NN
P'), ('S', 'NNP'), ('III', 'NNP'), (',', ','), ('running', 'VBG'), ('the',
'DT'), ('new', 'JJ'), ('Jelly', 'NNP'), ('Bean', 'NNP'), ('system', 'NN'),
(',', ','), ('the', 'DT'), ('Galaxy', 'NNP'), ('Tab', 'NNP'), ('8.9', 'CD'),
('Wifi', 'NNP'), ('tablet', 'NN'), (',', ','), ('the', 'DT'), ('Galaxy', 'NN
P'), ('Tab', 'NNP'), ('2', 'CD'), ('10.1', 'CD'), (',', ','), ('Galaxy', 'NN
P'), ('Rugby', 'NNP'), ('Pro', 'NNP'), ('and', 'CC'), ('Galaxy', 'NNP'),
('S', 'NNP'), ('III', 'NNP'), ('mini', 'NN'), ('.', '.')]
------------------------
[('Apple', 'NNP'), ('stated', 'VBD'), ('it', 'PRP'), ('had', 'VBD'), ('“',
'NNP'), ('acted', 'VBD'), ('quickly', 'RB'), ('and', 'CC'), ('diligently',
'RB'), ("''", "''"), ('in', 'IN'), ('order', 'NN'), ('to', 'TO'), ('``', '`
`'), ('determine', 'VB'), ('that', 'IN'), ('these', 'DT'), ('newly', 'RB'),
('released', 'VBN'), ('products', 'NNS'), ('do', 'VBP'), ('infringe', 'VB'),
('many', 'JJ'), ('of', 'IN'), ('the', 'DT'), ('same', 'JJ'), ('claims', 'NN
S'), ('already', 'RB'), ('asserted', 'VBN'), ('by', 'IN'), ('Apple', 'NNP'),
('.', '.'), ("''", "''")]
------------------------
[('In', 'IN'), ('August', 'NNP'), (',', ','), ('Samsung', 'NNP'), ('lost',
'VBD'), ('a', 'DT'), ('US', 'NNP'), ('patent', 'NN'), ('case', 'NN'), ('to',
'TO'), ('Apple', 'NNP'), ('and', 'CC'), ('was', 'VBD'), ('ordered', 'VBN'),
('to', 'TO'), ('pay', 'VB'), ('its', 'PRP$'), ('rival', 'JJ'), ('$', '$'),
('1.05bn', 'CD'), ('(', '('), ('£0.66bn', 'NN'), (')', ')'), ('in', 'IN'),
('damages', 'NNS'), ('for', 'IN'), ('copying', 'VBG'), ('features', 'NNS'),
('of', 'IN'), ('the', 'DT'), ('iPad', 'NN'), ('and', 'CC'), ('iPhone', 'N
N'), ('in', 'IN'), ('its', 'PRP$'), ('Galaxy', 'NNP'), ('range', 'NN'), ('o
f', 'IN'), ('devices', 'NNS'), ('.', '.')]
------------------------
[('Samsung', 'NNP'), (',', ','), ('which', 'WDT'), ('is', 'VBZ'), ('the', 'D
T'), ('world', 'NN'), ("'s", 'POS'), ('top', 'JJ'), ('mobile', 'NN'), ('phon
e', 'NN'), ('maker', 'NN'), (',', ','), ('is', 'VBZ'), ('appealing', 'VBG'),
('the', 'DT'), ('ruling', 'NN'), ('.', '.')]
------------------------
[('A', 'DT'), ('similar', 'JJ'), ('case', 'NN'), ('in', 'IN'), ('the', 'D
T'), ('UK', 'NNP'), ('found', 'VBD'), ('in', 'IN'), ('Samsung', 'NNP'),
("'s", 'POS'), ('favour', 'NN'), ('and', 'CC'), ('ordered', 'VBD'), ('Appl
e', 'NNP'), ('to', 'TO'), ('publish', 'VB'), ('an', 'DT'), ('apology', 'N
N'), ('making', 'VBG'), ('clear', 'JJ'), ('that', 'IN'), ('the', 'DT'), ('So
uth', 'JJ'), ('Korean', 'JJ'), ('firm', 'NN'), ('had', 'VBD'), ('not', 'R
B'), ('copied', 'VBN'), ('its', 'PRP$'), ('iPad', 'NN'), ('when', 'WRB'),
('designing', 'VBG'), ('its', 'PRP$'), ('own', 'JJ'), ('devices', 'NNS'),
('.', '.')]
------------------------
```

In [9]:

```
print(pos_tags_per_sentence)
```

```
[[('https', 'NN'), (':', ':'), ('//www.telegraph.co.uk/technology/apple/9702
716/Apple-Samsung-lawsuit-six-more-products-under-scrutiny.html', 'JJ'), ('D
ocuments', 'NNS'), ('filed', 'VBN'), ('to', 'TO'), ('the', 'DT'), ('San', 'N
NP'), ('Jose', 'NNP'), ('federal', 'JJ'), ('court', 'NN'), ('in', 'IN'), ('C
alifornia', 'NNP'), ('on', 'IN'), ('November', 'NNP'), ('23', 'CD'), ('lis
t', 'NN'), ('six', 'CD'), ('Samsung', 'NNP'), ('products', 'NNS'), ('runnin
g', 'VBG'), ('the', 'DT'), ('``', '``'), ('Jelly', 'RB'), ('Bean', 'NNP'),
("''", "''"), ('and', 'CC'), ('``', '``'), ('Ice', 'NNP'), ('Cream', 'NNP'),
('Sandwich', 'NNP'), ("''", "''"), ('operating', 'VBG'), ('systems', 'NNS'),
(',', ','), ('which', 'WDT'), ('Apple', 'NNP'), ('claims', 'VBZ'), ('infring
e', 'VB'), ('its', 'PRP$'), ('patents', 'NNS'), ('.', '.')], [('The', 'DT'),
('six', 'CD'), ('phones', 'NNS'), ('and', 'CC'), ('tablets', 'NNS'), ('affec
ted', 'VBN'), ('are', 'VBP'), ('the', 'DT'), ('Galaxy', 'NNP'), ('S', 'NN
P'), ('III', 'NNP'), (',', ','), ('running', 'VBG'), ('the', 'DT'), ('new',
'JJ'), ('Jelly', 'NNP'), ('Bean', 'NNP'), ('system', 'NN'), (',', ','), ('th
e', 'DT'), ('Galaxy', 'NNP'), ('Tab', 'NNP'), ('8.9', 'CD'), ('Wifi', 'NN
P'), ('tablet', 'NN'), (',', ','), ('the', 'DT'), ('Galaxy', 'NNP'), ('Tab',
'NNP'), ('2', 'CD'), ('10.1', 'CD'), (',', ','), ('Galaxy', 'NNP'), ('Rugb
y', 'NNP'), ('Pro', 'NNP'), ('and', 'CC'), ('Galaxy', 'NNP'), ('S', 'NNP'),
('III', 'NNP'), ('mini', 'NN'), ('.', '.')], [('Apple', 'NNP'), ('stated',
'VBD'), ('it', 'PRP'), ('had', 'VBD'), ('“', 'NNP'), ('acted', 'VBD'), ('qui
ckly', 'RB'), ('and', 'CC'), ('diligently', 'RB'), ("''", "''"), ('in', 'I
N'), ('order', 'NN'), ('to', 'TO'), ('``', '``'), ('determine', 'VB'), ('tha
t', 'IN'), ('these', 'DT'), ('newly', 'RB'), ('released', 'VBN'), ('product
s', 'NNS'), ('do', 'VBP'), ('infringe', 'VB'), ('many', 'JJ'), ('of', 'IN'),
('the', 'DT'), ('same', 'JJ'), ('claims', 'NNS'), ('already', 'RB'), ('asser
ted', 'VBN'), ('by', 'IN'), ('Apple', 'NNP'), ('.', '.'), ("''", "''")],
[('In', 'IN'), ('August', 'NNP'), (',', ','), ('Samsung', 'NNP'), ('lost',
'VBD'), ('a', 'DT'), ('US', 'NNP'), ('patent', 'NN'), ('case', 'NN'), ('to',
'TO'), ('Apple', 'NNP'), ('and', 'CC'), ('was', 'VBD'), ('ordered', 'VBN'),
('to', 'TO'), ('pay', 'VB'), ('its', 'PRP$'), ('rival', 'JJ'), ('$', '$'),
('1.05bn', 'CD'), ('(', '('), ('£0.66bn', 'NN'), (')', ')'), ('in', 'IN'),
('damages', 'NNS'), ('for', 'IN'), ('copying', 'VBG'), ('features', 'NNS'),
('of', 'IN'), ('the', 'DT'), ('iPad', 'NN'), ('and', 'CC'), ('iPhone', 'N
N'), ('in', 'IN'), ('its', 'PRP$'), ('Galaxy', 'NNP'), ('range', 'NN'), ('o
f', 'IN'), ('devices', 'NNS'), ('.', '.')], [('Samsung', 'NNP'), (',', ','),
('which', 'WDT'), ('is', 'VBZ'), ('the', 'DT'), ('world', 'NN'), ("'s", 'PO
S'), ('top', 'JJ'), ('mobile', 'NN'), ('phone', 'NN'), ('maker', 'NN'),
(',', ','), ('is', 'VBZ'), ('appealing', 'VBG'), ('the', 'DT'), ('ruling',
'NN'), ('.', '.')], [('A', 'DT'), ('similar', 'JJ'), ('case', 'NN'), ('in',
'IN'), ('the', 'DT'), ('UK', 'NNP'), ('found', 'VBD'), ('in', 'IN'), ('Samsu
ng', 'NNP'), ("'s", 'POS'), ('favour', 'NN'), ('and', 'CC'), ('ordered', 'VB
D'), ('Apple', 'NNP'), ('to', 'TO'), ('publish', 'VB'), ('an', 'DT'), ('apol
ogy', 'NN'), ('making', 'VBG'), ('clear', 'JJ'), ('that', 'IN'), ('the', 'D
T'), ('South', 'JJ'), ('Korean', 'JJ'), ('firm', 'NN'), ('had', 'VBD'), ('no
t', 'RB'), ('copied', 'VBN'), ('its', 'PRP$'), ('iPad', 'NN'), ('when', 'WR
B'), ('designing', 'VBG'), ('its', 'PRP$'), ('own', 'JJ'), ('devices', 'NN
S'), ('.', '.')]]
```

[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 [10]:
```
ner_tags_per_sentence = []
```

In [11]:
```
from nltk.chunk import ne_chunk
```

```
In [12]: for pos_tag_sentence in pos_tags_per_sentence:
             ner_tags_per_sentence.append(ne_chunk(pos_tag_sentence))
             print(ne_chunk(pos_tag_sentence))
             print("------------------------")

         nltk_ner = ner_tags_per_sentence
```

```
(S
  https/NN
  :/:
  //www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-m
ore-products-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 [13]:

```
print(ner_tags_per_sentence)
```

```
[Tree('S', [('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'),
Tree('ORGANIZATION', [('San', 'NNP'), ('Jose', 'NNP')]), ('federal', 'JJ'),
('court', 'NN'), ('in', 'IN'), Tree('GPE', [('California', 'NNP')]), ('on',
'IN'), ('November', 'NNP'), ('23', 'CD'), ('list', 'NN'), ('six', 'CD'), Tre
e('ORGANIZATION', [('Samsung', 'NNP')]), ('products', 'NNS'), ('running', 'V
BG'), ('the', 'DT'), ('``', '``'), ('Jelly', 'RB'), Tree('GPE', [('Bean', 'N
NP')]), ("''", "''"), ('and', 'CC'), ('``', '``'), ('Ice', 'NNP'), ('Cream',
'NNP'), ('Sandwich', 'NNP'), ("''", "''"), ('operating', 'VBG'), ('systems',
'NNS'), (',', ','), ('which', 'WDT'), Tree('PERSON', [('Apple', 'NNP')]),
('claims', 'VBZ'), ('infringe', 'VB'), ('its', 'PRP$'), ('patents', 'NNS'),
('.', '.')]), Tree('S', [('The', 'DT'), ('six', 'CD'), ('phones', 'NNS'),
('and', 'CC'), ('tablets', 'NNS'), ('affected', 'VBN'), ('are', 'VBP'), ('th
e', 'DT'), Tree('ORGANIZATION', [('Galaxy', 'NNP')]), ('S', 'NNP'), ('III',
'NNP'), (',', ','), ('running', 'VBG'), ('the', 'DT'), ('new', 'JJ'), Tree
('PERSON', [('Jelly', 'NNP'), ('Bean', 'NNP')]), ('system', 'NN'), (',',
','), ('the', 'DT'), Tree('ORGANIZATION', [('Galaxy', 'NNP')]), ('Tab', 'NN
P'), ('8.9', 'CD'), ('Wifi', 'NNP'), ('tablet', 'NN'), (',', ','), ('the',
'DT'), Tree('ORGANIZATION', [('Galaxy', 'NNP')]), ('Tab', 'NNP'), ('2', 'C
D'), ('10.1', 'CD'), (',', ','), Tree('PERSON', [('Galaxy', 'NNP'), ('Rugb
y', 'NNP'), ('Pro', 'NNP')]), ('and', 'CC'), Tree('PERSON', [('Galaxy', 'NN
P'), ('S', 'NNP')]), ('III', 'NNP'), ('mini', 'NN'), ('.', '.')]), Tree('S',
[Tree('PERSON', [('Apple', 'NNP')]), ('stated', 'VBD'), ('it', 'PRP'), ('ha
d', 'VBD'), ('“', 'NNP'), ('acted', 'VBD'), ('quickly', 'RB'), ('and', 'C
C'), ('diligently', 'RB'), ("''", "''"), ('in', 'IN'), ('order', 'NN'), ('t
o', 'TO'), ('``', '``'), ('determine', 'VB'), ('that', 'IN'), ('these', 'D
T'), ('newly', 'RB'), ('released', 'VBN'), ('products', 'NNS'), ('do', 'VB
P'), ('infringe', 'VB'), ('many', 'JJ'), ('of', 'IN'), ('the', 'DT'), ('sam
e', 'JJ'), ('claims', 'NNS'), ('already', 'RB'), ('asserted', 'VBN'), ('by',
'IN'), Tree('PERSON', [('Apple', 'NNP')]), ('.', '.'), ("''", "''")]), Tree
('S', [('In', 'IN'), Tree('GPE', [('August', 'NNP')]), (',', ','), Tree('PER
SON', [('Samsung', 'NNP')]), ('lost', 'VBD'), ('a', 'DT'), Tree('GSP', [('U
S', 'NNP')]), ('patent', 'NN'), ('case', 'NN'), ('to', 'TO'), Tree('GPE',
[('Apple', 'NNP')]), ('and', 'CC'), ('was', 'VBD'), ('ordered', 'VBN'), ('t
o', 'TO'), ('pay', 'VB'), ('its', 'PRP$'), ('rival', 'JJ'), ('$', '$'), ('1.
05bn', 'CD'), ('(', '('), ('£0.66bn', 'NN'), (')', ')'), ('in', 'IN'), ('dam
ages', 'NNS'), ('for', 'IN'), ('copying', 'VBG'), ('features', 'NNS'), ('o
f', 'IN'), ('the', 'DT'), Tree('ORGANIZATION', [('iPad', 'NN')]), ('and', 'C
C'), Tree('ORGANIZATION', [('iPhone', 'NN')]), ('in', 'IN'), ('its', 'PRP
$'), Tree('GPE', [('Galaxy', 'NNP')]), ('range', 'NN'), ('of', 'IN'), ('devi
ces', 'NNS'), ('.', '.')]), Tree('S', [Tree('GPE', [('Samsung', 'NNP')]),
(',', ','), ('which', 'WDT'), ('is', 'VBZ'), ('the', 'DT'), ('world', 'NN'),
("'s", 'POS'), ('top', 'JJ'), ('mobile', 'NN'), ('phone', 'NN'), ('maker',
'NN'), (',', ','), ('is', 'VBZ'), ('appealing', 'VBG'), ('the', 'DT'), ('rul
ing', 'NN'), ('.', '.')]), Tree('S', [('A', 'DT'), ('similar', 'JJ'), ('cas
e', 'NN'), ('in', 'IN'), ('the', 'DT'), Tree('ORGANIZATION', [('UK', 'NN
P')]), ('found', 'VBD'), ('in', 'IN'), Tree('GPE', [('Samsung', 'NNP')]),
("'s", 'POS'), ('favour', 'NN'), ('and', 'CC'), ('ordered', 'VBD'), Tree('PE
RSON', [('Apple', 'NNP')]), ('to', 'TO'), ('publish', 'VB'), ('an', 'DT'),
('apology', 'NN'), ('making', 'VBG'), ('clear', 'JJ'), ('that', 'IN'), ('th
e', 'DT'), Tree('LOCATION', [('South', 'JJ'), ('Korean', 'JJ')]), ('firm',
'NN'), ('had', 'VBD'), ('not', 'RB'), ('copied', 'VBN'), ('its', 'PRP$'),
('iPad', 'NN'), ('when', 'WRB'), ('designing', 'VBG'), ('its', 'PRP$'), ('ow
n', '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 [14]:

```
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 [15]:

```
constituency_output_per_sentence = []
```

In [16]:

```
for sentence in pos_tags_per_sentence:
    constituent_structure = constituent_parser.parse(sentence)
    constituency_output_per_sentence.append(constituent_structure)
    print(constituent_structure)
    #constituent_structure.draw()
    print("-------------------------")
```

```
(S
  (NP https/NN)
  :/:
  (NP
    //www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six
-more-products-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 [17]:

```
print(constituency_output_per_sentence)
```

```
[Tree('S', [Tree('NP', [('https', 'NN')]), (':', ':'), Tree('NP', [('//www.t
elegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more-produ
cts-under-scrutiny.html', 'JJ')]), ('Documents', 'NNS'), Tree('VP', [Tree
('V', [('filed', 'VBN')])]), ('to', 'TO'), Tree('NP', [('the', 'DT')]), ('Sa
n', 'NNP'), ('Jose', 'NNP'), Tree('NP', [('federal', 'JJ'), ('court', 'N
N')]), Tree('P', [('in', 'IN')]), ('California', 'NNP'), Tree('P', [('on',
'IN')]), ('November', 'NNP'), ('23', 'CD'), Tree('NP', [('list', 'NN')]),
('six', 'CD'), ('Samsung', 'NNP'), ('products', 'NNS'), Tree('VP', [Tree
('V', [('running', 'VBG')]), Tree('NP', [('the', 'DT')])]), ('``', '``'),
('Jelly', 'RB'), ('Bean', 'NNP'), ("''", "''"), ('and', 'CC'), ('``', '``'),
('Ice', 'NNP'), ('Cream', 'NNP'), ('Sandwich', 'NNP'), ("''", "''"), Tree('V
P', [Tree('V', [('operating', 'VBG')])]), ('systems', 'NNS'), (',', ','),
('which', 'WDT'), ('Apple', 'NNP'), Tree('VP', [Tree('V', [('claims', 'VB
Z')])]), Tree('VP', [Tree('V', [('infringe', 'VB')])]), ('its', 'PRP$'), ('p
atents', 'NNS'), ('.', '.')]), Tree('S', [Tree('NP', [('The', 'DT')]), ('si
x', 'CD'), ('phones', 'NNS'), ('and', 'CC'), ('tablets', 'NNS'), Tree('VP',
[Tree('V', [('affected', 'VBN')])]), Tree('VP', [Tree('V', [('are', 'VB
P')]), Tree('NP', [('the', 'DT')])]), ('Galaxy', 'NNP'), ('S', 'NNP'), ('II
I', 'NNP'), (',', ','), Tree('VP', [Tree('V', [('running', 'VBG')]), Tree('N
P', [('the', 'DT'), ('new', 'JJ')])]), ('Jelly', 'NNP'), ('Bean', 'NNP'), Tr
ee('NP', [('system', 'NN')]), (',', ','), Tree('NP', [('the', 'DT')]), ('Gal
axy', 'NNP'), ('Tab', 'NNP'), ('8.9', 'CD'), ('Wifi', 'NNP'), Tree('NP',
[('tablet', 'NN')]), (',', ','), Tree('NP', [('the', 'DT')]), ('Galaxy', 'NN
P'), ('Tab', 'NNP'), ('2', 'CD'), ('10.1', 'CD'), (',', ','), ('Galaxy', 'NN
P'), ('Rugby', 'NNP'), ('Pro', 'NNP'), ('and', 'CC'), ('Galaxy', 'NNP'),
('S', 'NNP'), ('III', 'NNP'), Tree('NP', [('mini', 'NN')]), ('.', '.')]), Tr
ee('S', [('Apple', 'NNP'), Tree('VP', [Tree('V', [('stated', 'VBD')])]), ('i
t', 'PRP'), Tree('VP', [Tree('V', [('had', 'VBD')])]), ('“', 'NNP'), Tree('V
P', [Tree('V', [('acted', 'VBD')])]), ('quickly', 'RB'), ('and', 'CC'), ('di
ligently', 'RB'), ("''", "''"), Tree('PP', [Tree('P', [('in', 'IN')]), Tree
('NP', [('order', 'NN')])]), ('to', 'TO'), ('``', '``'), Tree('VP', [Tree
('V', [('determine', 'VB')]), Tree('PP', [Tree('P', [('that', 'IN')]), Tree
('NP', [('these', 'DT')])])]), ('newly', 'RB'), Tree('VP', [Tree('V', [('rel
eased', 'VBN')])]), ('products', 'NNS'), Tree('VP', [Tree('V', [('do', 'VB
P')])]), Tree('VP', [Tree('V', [('infringe', 'VB')]), Tree('NP', [('many',
'JJ')]), Tree('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', [Tree('P', [('In', 'IN')]), ('August',
'NNP'), (',', ','), ('Samsung', 'NNP'), Tree('VP', [Tree('V', [('lost', 'VB
D')]), Tree('NP', [('a', 'DT')])]), ('US', 'NNP'), Tree('NP', [('patent', 'N
N'), ('case', 'NN')]), ('to', 'TO'), ('Apple', 'NNP'), ('and', 'CC'), Tree
('VP', [Tree('V', [('was', 'VBD')])]), Tree('VP', [Tree('V', [('ordered', 'V
BN')])]), ('to', 'TO'), Tree('VP', [Tree('V', [('pay', 'VB')])]), ('its', 'P
RP$'), Tree('NP', [('rival', 'JJ')]), ('$', '$'), ('1.05bn', 'CD'), ('(',
'('), Tree('NP', [('£0.66bn', 'NN')]), (')', ')'), Tree('P', [('in', 'I
N')]), ('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$'), ('Ga
laxy', 'NNP'), Tree('NP', [('range', 'NN')]), Tree('P', [('of', 'IN')]), ('d
evices', 'NNS'), ('.', '.')]), Tree('S', [('Samsung', 'NNP'), (',', ','),
('which', 'WDT'), Tree('VP', [Tree('V', [('is', 'VBZ')]), Tree('NP', [('th
e', 'DT'), ('world', 'NN')])]), ("'s", 'POS'), Tree('NP', [('top', 'JJ'),
('mobile', 'NN'), ('phone', 'NN'), ('maker', 'NN')]), (',', ','), Tree('VP',
[Tree('V', [('is', 'VBZ')])]), Tree('VP', [Tree('V', [('appealing', 'VB
G')]), Tree('NP', [('the', 'DT'), ('ruling', 'NN')])]), ('.', '.')]), Tree
('S', [Tree('NP', [('A', 'DT'), ('similar', 'JJ'), ('case', 'NN')]), Tree('P
P', [Tree('P', [('in', 'IN')]), Tree('NP', [('the', 'DT')])]), ('UK', 'NN
P'), Tree('VP', [Tree('V', [('found', 'VBD')])]), Tree('P', [('in', 'IN')]),
('Samsung', 'NNP'), ("'s", 'POS'), Tree('NP', [('favour', 'NN')]), ('and',
'CC'), Tree('VP', [Tree('V', [('ordered', 'VBD')])]), ('Apple', 'NNP'), ('t
o', 'TO'), Tree('VP', [Tree('V', [('publish', 'VB')]), Tree('NP', [('an', 'D
T'), ('apology', 'NN')])]), Tree('VP', [Tree('V', [('making', 'VBG')]), Tree
('NP', [('clear', 'JJ')]), Tree('PP', [Tree('P', [('that', 'IN')]), Tree('N
P', [('the', 'DT'), ('South', 'JJ'), ('Korean', 'JJ'), ('firm', 'NN')])])]),
Tree('VP', [Tree('V', [('had', 'VBD')])]), ('not', 'RB'), Tree('VP', [Tree
('V', [('copied', 'VBN')])]), ('its', 'PRP$'), Tree('NP', [('iPad', 'NN')]),
('when', 'WRB'), Tree('VP', [Tree('V', [('designing', 'VBG')])]), ('its', 'P
RP$'), 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 [18]:

```
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 [19]:

```
constituency_v2_output_per_sentence = []
```

In [20]:

```
for sentence in pos_tags_per_sentence:
    constituent_structure = constituent_parser_v2.parse(sentence)
    constituency_v2_output_per_sentence.append(constituent_structure)
    print(constituent_structure)
    #constituent_structure.draw()
    print("-------------------------")
```

```
(S
  (NP https/NN)
  :/:
  (NP
    //www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six
-more-products-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 [21]:

```
print(constituency_v2_output_per_sentence)
```

```
[Tree('S', [Tree('NP', [('https', 'NN')]), (':', ':'), Tree('NP', [('//www.t
elegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more-produ
cts-under-scrutiny.html', 'JJ')]), ('Documents', 'NNS'), Tree('VP', [Tree
('V', [('filed', 'VBN')])]), ('to', 'TO'), Tree('NP', [('the', 'DT')]), ('Sa
n', 'NNP'), ('Jose', 'NNP'), Tree('NP', [('federal', 'JJ'), ('court', 'N
N')]), Tree('P', [('in', 'IN')]), ('California', 'NNP'), Tree('P', [('on',
'IN')]), ('November', 'NNP'), ('23', 'CD'), Tree('NP', [('list', 'NN')]),
('six', 'CD'), ('Samsung', 'NNP'), ('products', 'NNS'), Tree('VP', [Tree
('V', [('running', 'VBG')]), Tree('NP', [('the', 'DT')])]), ('``', '``'),
('Jelly', 'RB'), ('Bean', 'NNP'), ("''", "''"), ('and', 'CC'), ('``', '``'),
('Ice', 'NNP'), ('Cream', 'NNP'), ('Sandwich', 'NNP'), ("''", "''"), Tree('V
P', [Tree('V', [('operating', 'VBG')])]), ('systems', 'NNS'), (',', ','),
('which', 'WDT'), ('Apple', 'NNP'), Tree('VP', [Tree('V', [('claims', 'VB
Z')])]), Tree('VP', [Tree('V', [('infringe', 'VB')])]), ('its', 'PRP$'), ('p
atents', 'NNS'), ('.', '.')]), Tree('S', [Tree('NP', [('The', 'DT')]), ('si
x', 'CD'), ('phones', 'NNS'), ('and', 'CC'), ('tablets', 'NNS'), Tree('VP',
[Tree('V', [('affected', 'VBN')])]), Tree('VP', [Tree('V', [('are', 'VB
P')]), Tree('NP', [('the', 'DT')])]), ('Galaxy', 'NNP'), ('S', 'NNP'), ('II
I', 'NNP'), (',', ','), Tree('VP', [Tree('V', [('running', 'VBG')]), Tree('N
P', [('the', 'DT'), ('new', 'JJ')])]), ('Jelly', 'NNP'), ('Bean', 'NNP'), Tr
ee('NP', [('system', 'NN')]), (',', ','), Tree('NP', [('the', 'DT')]), ('Gal
axy', 'NNP'), ('Tab', 'NNP'), ('8.9', 'CD'), ('Wifi', 'NNP'), Tree('NP',
[('tablet', 'NN')]), (',', ','), Tree('NP', [('the', 'DT')]), ('Galaxy', 'NN
P'), ('Tab', 'NNP'), ('2', 'CD'), ('10.1', 'CD'), (',', ','), ('Galaxy', 'NN
P'), ('Rugby', 'NNP'), ('Pro', 'NNP'), ('and', 'CC'), ('Galaxy', 'NNP'),
('S', 'NNP'), ('III', 'NNP'), Tree('NP', [('mini', 'NN')]), ('.', '.')]), Tr
ee('S', [('Apple', 'NNP'), Tree('VP', [Tree('V', [('stated', 'VBD')])]), ('i
t', 'PRP'), Tree('VP', [Tree('V', [('had', 'VBD')])]), ('“', 'NNP'), Tree('V
P', [Tree('V', [('acted', 'VBD')])]), ('quickly', 'RB'), ('and', 'CC'), ('di
ligently', 'RB'), ("''", "''"), Tree('PP', [Tree('P', [('in', 'IN')]), Tree
('NP', [('order', 'NN')])]), ('to', 'TO'), ('``', '``'), Tree('VP', [Tree
('V', [('determine', 'VB')]), Tree('PP', [Tree('P', [('that', 'IN')]), Tree
('NP', [('these', 'DT')])])]), ('newly', 'RB'), Tree('VP', [Tree('V', [('rel
eased', 'VBN')])]), ('products', 'NNS'), Tree('VP', [Tree('V', [('do', 'VB
P')])]), Tree('VP', [Tree('V', [('infringe', 'VB')]), Tree('NP', [('many',
'JJ')]), Tree('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', [Tree('P', [('In', 'IN')]), ('August',
'NNP'), (',', ','), ('Samsung', 'NNP'), Tree('VP', [Tree('V', [('lost', 'VB
D')]), Tree('NP', [('a', 'DT')])]), ('US', 'NNP'), Tree('NP', [('patent', 'N
N'), ('case', 'NN')]), ('to', 'TO'), ('Apple', 'NNP'), ('and', 'CC'), Tree
('VP', [Tree('V', [('was', 'VBD')])]), Tree('VP', [Tree('V', [('ordered', 'V
BN')])]), ('to', 'TO'), Tree('VP', [Tree('V', [('pay', 'VB')])]), ('its', 'P
RP$'), Tree('NP', [('rival', 'JJ')]), ('$', '$'), ('1.05bn', 'CD'), ('(',
'('), Tree('NP', [('£0.66bn', 'NN')]), (')', ')'), Tree('P', [('in', 'I
N')]), ('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$'), ('Ga
laxy', 'NNP'), Tree('NP', [('range', 'NN')]), Tree('P', [('of', 'IN')]), ('d
evices', 'NNS'), ('.', '.')]), Tree('S', [('Samsung', 'NNP'), (',', ','),
('which', 'WDT'), Tree('VP', [Tree('V', [('is', 'VBZ')]), Tree('NP', [('th
e', 'DT'), ('world', 'NN')])]), ("'s", 'POS'), Tree('NP', [('top', 'JJ'),
('mobile', 'NN'), ('phone', 'NN'), ('maker', 'NN')]), (',', ','), Tree('VP',
[Tree('V', [('is', 'VBZ')])]), Tree('VP', [Tree('V', [('appealing', 'VB
G')]), Tree('NP', [('the', 'DT'), ('ruling', 'NN')])]), ('.', '.')]), Tree
('S', [Tree('NP', [('A', 'DT'), ('similar', 'JJ'), ('case', 'NN')]), Tree('P
P', [Tree('P', [('in', 'IN')]), Tree('NP', [('the', 'DT')])]), ('UK', 'NN
P'), Tree('VP', [Tree('V', [('found', 'VBD')])]), Tree('P', [('in', 'IN')]),
('Samsung', 'NNP'), ("'s", 'POS'), Tree('NP', [('favour', 'NN')]), ('and',
'CC'), Tree('VP', [Tree('V', [('ordered', 'VBD')])]), ('Apple', 'NNP'), ('t
o', 'TO'), Tree('VP', [Tree('V', [('publish', 'VB')]), Tree('NP', [('an', 'D
T'), ('apology', 'NN')])]), Tree('VP', [Tree('V', [('making', 'VBG')]), Tree
('NP', [('clear', 'JJ')]), Tree('PP', [Tree('P', [('that', 'IN')]), Tree('N
P', [('the', 'DT'), ('South', 'JJ'), ('Korean', 'JJ'), ('firm', 'NN')])])]),
Tree('VP', [Tree('V', [('had', 'VBD')])]), ('not', 'RB'), Tree('VP', [Tree
('V', [('copied', 'VBN')])]), ('its', 'PRP$'), Tree('NP', [('iPad', 'NN')]),
('when', 'WRB'), Tree('VP', [Tree('V', [('designing', 'VBG')])]), ('its', 'P
RP$'), 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 [22]:

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

In [23]:

```python
doc = nlp(text)
```

In [24]:

```python
sents = list(doc.sents)
#tokenization

tokens_per_sentence = []
for sent in sents:
    for token in sent:
        tokens_per_sentence.append(token.text)
print(tokens_per_sentence)
```

```
['https://www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit
-six-more-products-under-scrutiny.html', '\n\n', 'Documents', 'filed', 'to',
'the', 'San', 'Jose', 'federal', 'court', 'in', 'California', 'on', 'Novembe
r', '23', 'list', 'six', 'Samsung', 'products', 'running', 'the', '"', 'Jell
y', 'Bean', '"', 'and', '"', 'Ice', 'Cream', 'Sandwich', '"', 'operating',
'systems', ',', 'which', 'Apple', 'claims', 'infringe', 'its', 'patents',
'.', '\n', 'The', 'six', 'phones', 'and', 'tablets', 'affected', 'are', 'th
e', 'Galaxy', 'S', 'III', ',', 'running', 'the', 'new', 'Jelly', 'Bean', 'sy
stem', ',', 'the', 'Galaxy', 'Tab', '8.9', 'Wifi', 'tablet', ',', 'the', 'Ga
laxy', 'Tab', '2', '10.1', ',', 'Galaxy', 'Rugby', 'Pro', 'and', 'Galaxy',
'S', 'III', 'mini', '.', '\n', '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', 'asserted', 'by', 'Apple', '.',
'"', '\n', 'In', 'August', ',', 'Samsung', 'lost', 'a', 'US', 'patent', 'cas
e', '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', '.', 'Samsung', ',', 'which', 'is', 'the', 'world',
"'s", 'top', 'mobile', 'phone', 'maker', ',', 'is', 'appealing', 'the', 'rul
ing', '.', '\n', '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', 'ha
d', 'not', 'copied', 'its', 'iPad', 'when', 'designing', 'its', 'own', 'devi
ces', '.']
```

In [25]:

```python
#pos tagging
pos_tags_per_sentence = []
for sent in sents:
    for token in sent:
        pos_tags_per_sentence.append((token.text, token.tag_))
        #print(token.text, token.tag_, token.pos_)
        #print("------------------------")
print(pos_tags_per_sentence)
```

```
[('https://www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsui
t-six-more-products-under-scrutiny.html', 'NNP'), ('\n\n', '_SP'), ('Documen
ts', 'NNS'), ('filed', 'VBD'), ('to', 'IN'), ('the', 'DT'), ('San', 'NNP'),
('Jose', 'NNP'), ('federal', 'JJ'), ('court', 'NN'), ('in', 'IN'), ('Califor
nia', 'NNP'), ('on', 'IN'), ('November', 'NNP'), ('23', 'CD'), ('list', 'N
N'), ('six', 'CD'), ('Samsung', 'JJ'), ('products', 'NNS'), ('running', 'VB
G'), ('the', 'DT'), ('"', '``'), ('Jelly', 'NNP'), ('Bean', 'NNP'), ('"',
"''"), ('and', 'CC'), ('"', '``'), ('Ice', 'NNP'), ('Cream', 'NNP'), ('Sandw
ich', 'NNP'), ('"', "''"), ('operating', 'VBG'), ('systems', 'NNS'), (',',
','), ('which', 'WDT'), ('Apple', 'NNP'), ('claims', 'VBZ'), ('infringe', 'V
BP'), ('its', 'PRP$'), ('patents', 'NNS'), ('.', '.'), ('\n', '_SP'), ('Th
e', 'DT'), ('six', 'CD'), ('phones', 'NNS'), ('and', 'CC'), ('tablets', 'NN
S'), ('affected', 'VBN'), ('are', 'VBP'), ('the', 'DT'), ('Galaxy', 'NNP'),
('S', 'NNP'), ('III', 'NNP'), (',', ','), ('running', 'VBG'), ('the', 'DT'),
('new', 'JJ'), ('Jelly', 'NNP'), ('Bean', 'NNP'), ('system', 'NN'), (',',
','), ('the', 'DT'), ('Galaxy', 'NNP'), ('Tab', 'NNP'), ('8.9', 'CD'), ('Wif
i', 'NNP'), ('tablet', 'NN'), (',', ','), ('the', 'DT'), ('Galaxy', 'NNP'),
('Tab', 'NNP'), ('2', 'CD'), ('10.1', 'CD'), (',', ','), ('Galaxy', 'NNP'),
('Rugby', 'NNP'), ('Pro', 'NNP'), ('and', 'CC'), ('Galaxy', 'NNP'), ('S', 'N
NP'), ('III', 'CD'), ('mini', 'NN'), ('.', '.'), ('\n', '_SP'), ('Apple', 'N
NP'), ('stated', 'VBD'), ('it', 'PRP'), ('had', 'VBD'), ('“', '``'), ('acte
d', 'VBN'), ('quickly', 'RB'), ('and', 'CC'), ('diligently', 'RB'), ('"',
"''"), ('in', 'IN'), ('order', 'NN'), ('to', 'TO'), ('"', '``'), ('determin
e', 'VB'), ('that', 'IN'), ('these', 'DT'), ('newly', 'RB'), ('released', 'V
BN'), ('products', 'NNS'), ('do', 'VBP'), ('infringe', 'VB'), ('many', 'J
J'), ('of', 'IN'), ('the', 'DT'), ('same', 'JJ'), ('claims', 'NNS'), ('alrea
dy', 'RB'), ('asserted', 'VBN'), ('by', 'IN'), ('Apple', 'NNP'), ('.', '.'),
('"', "''"), ('\n', '_SP'), ('In', 'IN'), ('August', 'NNP'), (',', ','), ('S
amsung', 'NNP'), ('lost', 'VBD'), ('a', 'DT'), ('US', 'NNP'), ('patent', 'N
N'), ('case', 'NN'), ('to', 'IN'), ('Apple', 'NNP'), ('and', 'CC'), ('was',
'VBD'), ('ordered', 'VBN'), ('to', 'TO'), ('pay', 'VB'), ('its', 'PRP$'),
('rival', 'JJ'), ('$', '$'), ('1.05bn', 'CD'), ('(', '-LRB-'), ('£', '$'),
('0.66bn', 'CD'), (')', '-RRB-'), ('in', 'IN'), ('damages', 'NNS'), ('for',
'IN'), ('copying', 'NN'), ('features', 'NNS'), ('of', 'IN'), ('the', 'DT'),
('iPad', 'NNP'), ('and', 'CC'), ('iPhone', 'NNP'), ('in', 'IN'), ('its', 'PR
P$'), ('Galaxy', 'NNP'), ('range', 'NN'), ('of', 'IN'), ('devices', 'NNS'),
('.', '.'), ('Samsung', 'NNP'), (',', ','), ('which', 'WDT'), ('is', 'VBZ'),
('the', 'DT'), ('world', 'NN'), ("'s", 'POS'), ('top', 'JJ'), ('mobile', 'J
J'), ('phone', 'NN'), ('maker', 'NN'), (',', ','), ('is', 'VBZ'), ('appealin
g', 'VBG'), ('the', 'DT'), ('ruling', 'NN'), ('.', '.'), ('\n', '_SP'),
('A', 'DT'), ('similar', 'JJ'), ('case', 'NN'), ('in', 'IN'), ('the', 'DT'),
('UK', 'NNP'), ('found', 'VBD'), ('in', 'IN'), ('Samsung', 'NNP'), ("'s", 'P
OS'), ('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'), ('K
orean', 'JJ'), ('firm', 'NN'), ('had', 'VBD'), ('not', 'RB'), ('copied', 'VB
N'), ('its', 'PRP$'), ('iPad', 'NNP'), ('when', 'WRB'), ('designing', 'VB
G'), ('its', 'PRP$'), ('own', 'JJ'), ('devices', 'NNS'), ('.', '.')]
```

In [26]:

```
#ner
from spacy import displacy
displacy.render(doc, jupyter=True, style='ent')
```

https://www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more-products-under-scrutiny.html

Documents filed to the San Jose GPE federal court in California GPE on November 23 DATE list six CARDINAL Samsung ORG products running the "Jelly Bean" LAW and " Ice Cream Sandwich WORK_OF_ART " operating systems, which Apple ORG claims infringe its patents.

The six CARDINAL phones and tablets affected are the Galaxy S III GPE , running the new Jelly Bean ORG system, the Galaxy Tab 8.9 PRODUCT Wifi PERSON tablet, the Galaxy Tab 2 10.1 DATE , Galaxy Rugby Pro PERSON and Galaxy S PERSON III mini.

Apple ORG stated it had "acted quickly and diligently" in order to "determine that these newly released products do infringe many of the same claims already asserted by Apple ORG ."

In August DATE , Samsung ORG lost a US GPE patent case to Apple ORG and was ordered to pay its rival $ 1.05bn MONEY (£ 0.66bn MONEY) in damages for copying features of the iPad LOC and iPhone ORG in its Galaxy ORG range of devices. Samsung ORG , which is the world's top mobile phone maker, is appealing the ruling.

A similar case in the UK GPE found in Samsung ORG 's favour and ordered Apple ORG to publish an apology making clear that the South Korean NORP firm had not copied its iPad when designing its own devices.

In [27]:
```
ner_text_and_labels = []
for ent in doc.ents:
    ner_text_and_labels.append([(ent.text, ent.label_)])
    #print(ent.text, ent.label_)
print(ner_text_and_labels)
spacy_ner = ner_text_and_labels
```

```
[[('San Jose', 'GPE')], [('California', 'GPE')], [('November 23', 'DATE')],
[('six', 'CARDINAL')], [('Samsung', 'ORG')], [('the "Jelly Bean"', 'LAW')],
[('Ice Cream Sandwich', 'WORK_OF_ART')], [('Apple', 'ORG')], [('six', 'CARDI
NAL')], [('the Galaxy S III', 'GPE')], [('Jelly Bean', 'ORG')], [('Tab 8.9',
'PRODUCT')], [('Wifi', 'PERSON')], [('2 10.1', 'DATE')], [('Rugby Pro', 'PER
SON')], [('Galaxy S', 'PERSON')], [('Apple', 'ORG')], [('Apple', 'ORG')],
[('August', 'DATE')], [('Samsung', 'ORG')], [('US', 'GPE')], [('Apple', 'OR
G')], [('1.05bn', 'MONEY')], [('0.66bn', 'MONEY')], [('iPad', 'LOC')], [('iP
hone', 'ORG')], [('Galaxy', 'ORG')], [('Samsung', 'ORG')], [('UK', 'GPE')],
[('Samsung', 'ORG')], [('Apple', 'ORG')], [('South Korean', 'NORP')]]
```

In [28]:

```
#Constituency/dependency parsing
```

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.

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

Sentence splitting in NLTK and spaCy is different. NLTK did not put the link and the first sentence in the same sentence whereas spaCy views them as seperate sentences. spaCy made an error by splitting "Galaxy S" and "III mini." whereas NLTK did it properly. Also spacy puts endline after its sentences.

In [29]:

```
#Print all sentences
print("NLTK")
i = 0
for sentence in sentences_nltk:
    print("\nNo: " + str(i))
    print(sentence)
    i+=1

i = 0
print("\nSPACY")
for sentence in list(doc.sents):
    print("\nNo: " + str(i))
    print(sentence)
    i+=1
```

```
NLTK

No: 0
https://www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-s
ix-more-products-under-scrutiny.html

Documents filed to the San Jose federal court in California on November 23 l
ist six Samsung products running the "Jelly Bean" and "Ice Cream Sandwich" o
perating systems, which Apple claims infringe its patents.

No: 1
The six phones and tablets affected are the Galaxy S III, running the new Je
lly Bean system, the Galaxy Tab 8.9 Wifi tablet, the Galaxy Tab 2 10.1, Gala
xy Rugby Pro and Galaxy S III mini.

No: 2
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."

No: 3
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 iPho
ne in its Galaxy range of devices.

No: 4
Samsung, which is the world's top mobile phone maker, is appealing the rulin
g.

No: 5
A similar case in the UK found in Samsung's favour and ordered Apple to publ
ish an apology making clear that the South Korean firm had not copied its iP
ad when designing its own devices.

SPACY

No: 0
https://www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-s
ix-more-products-under-scrutiny.html




No: 1
Documents filed to the San Jose federal court in California on November 23 l
ist six Samsung products running the "Jelly Bean" and "Ice Cream Sandwich" o
perating systems, which Apple claims infringe its patents.



No: 2
The six phones and tablets affected are the Galaxy S III, running the new Je
lly Bean system, the Galaxy Tab 8.9 Wifi tablet, the Galaxy Tab 2 10.1, Gala
xy Rugby Pro and Galaxy S

No: 3
III mini.



No: 4
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."



No: 5
```

```
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 iPho
ne in its Galaxy range of devices.

No: 6
Samsung, which is the world's top mobile phone maker, is appealing the rulin
g.


No: 7
A similar case in the UK found in Samsung's favour and ordered Apple to publ
ish an apology making clear that the South Korean firm had not copied its iP
ad when designing its own devices.
```

We select the sentence 2 because it is the one that spaCy made an error and that might cause the parts of speech to be mixed.

In [36]:

```
nltk_tokens = word_tokenize(sentences_nltk[1])
nltk_pos = nltk.pos_tag(nltk_tokens)

sent = list(doc.sents)[2]

for i in range(len(sent)):
    print("NLTK/spaCy: " + str(nltk_pos[i]) + "/" + str( (sent[i].text, sent


sent2 = list(doc.sents)[3]

for i in range(len(sent), len(sent) + len(sent2) - 1):
    j = i - len(sent)
    print("NLTK/spaCy: " + str(nltk_pos[i]) + "/" + str( (sent2[j].text, sen
NLTK/spaCy: ('The', 'DT')/('The', 'DET')
NLTK/spaCy: ('six', 'CD')/('six', 'NUM')
NLTK/spaCy: ('phones', 'NNS')/('phones', 'NOUN')
NLTK/spaCy: ('and', 'CC')/('and', 'CCONJ')
NLTK/spaCy: ('tablets', 'NNS')/('tablets', 'NOUN')
NLTK/spaCy: ('affected', 'VBN')/('affected', 'VERB')
NLTK/spaCy: ('are', 'VBP')/('are', 'AUX')
NLTK/spaCy: ('the', 'DT')/('the', 'DET')
NLTK/spaCy: ('Galaxy', 'NNP')/('Galaxy', 'PROPN')
NLTK/spaCy: ('S', 'NNP')/('S', 'PROPN')
NLTK/spaCy: ('III', 'NNP')/('III', 'PROPN')
NLTK/spaCy: (',', ',')/(',', 'PUNCT')
NLTK/spaCy: ('running', 'VBG')/('running', 'VERB')
NLTK/spaCy: ('the', 'DT')/('the', 'DET')
NLTK/spaCy: ('new', 'JJ')/('new', 'ADJ')
NLTK/spaCy: ('Jelly', 'NNP')/('Jelly', 'PROPN')
NLTK/spaCy: ('Bean', 'NNP')/('Bean', 'PROPN')
NLTK/spaCy: ('system', 'NN')/('system', 'NOUN')
NLTK/spaCy: (',', ',')/(',', 'PUNCT')
NLTK/spaCy: ('the', 'DT')/('the', 'DET')
NLTK/spaCy: ('Galaxy', 'NNP')/('Galaxy', 'PROPN')
NLTK/spaCy: ('Tab', 'NNP')/('Tab', 'PROPN')
NLTK/spaCy: ('8.9', 'CD')/('8.9', 'NUM')
NLTK/spaCy: ('Wifi', 'NNP')/('Wifi', 'PROPN')
NLTK/spaCy: ('tablet', 'NN')/('tablet', 'NOUN')
NLTK/spaCy: (',', ',')/(',', 'PUNCT')
NLTK/spaCy: ('the', 'DT')/('the', 'DET')
NLTK/spaCy: ('Galaxy', 'NNP')/('Galaxy', 'PROPN')
NLTK/spaCy: ('Tab', 'NNP')/('Tab', 'PROPN')
NLTK/spaCy: ('2', 'CD')/('2', 'NUM')
NLTK/spaCy: ('10.1', 'CD')/('10.1', 'NUM')
NLTK/spaCy: (',', ',')/(',', 'PUNCT')
NLTK/spaCy: ('Galaxy', 'NNP')/('Galaxy', 'PROPN')
NLTK/spaCy: ('Rugby', 'NNP')/('Rugby', 'PROPN')
NLTK/spaCy: ('Pro', 'NNP')/('Pro', 'PROPN')
NLTK/spaCy: ('and', 'CC')/('and', 'CCONJ')
NLTK/spaCy: ('Galaxy', 'NNP')/('Galaxy', 'PROPN')
NLTK/spaCy: ('S', 'NNP')/('S', 'PROPN')
NLTK/spaCy: ('III', 'NNP')/('III', 'NUM')
NLTK/spaCy: ('mini', 'NN')/('mini', 'NOUN')
NLTK/spaCy: ('.', '.')/('.', 'PUNCT')
```

There are no difference between the outputs.

[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 [38]:

```python
print("NLTK:\n")
print(nltk_ner)

print("\n\nspaCy:\n")
print(spacy_ner)
```

```
NLTK:

[Tree('S', [('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'),
Tree('ORGANIZATION', [('San', 'NNP'), ('Jose', 'NNP')]), ('federal', 'JJ'),
('court', 'NN'), ('in', 'IN'), Tree('GPE', [('California', 'NNP')]), ('on',
'IN'), ('November', 'NNP'), ('23', 'CD'), ('list', 'NN'), ('six', 'CD'), Tre
e('ORGANIZATION', [('Samsung', 'NNP')]), ('products', 'NNS'), ('running', 'V
BG'), ('the', 'DT'), ('``', '``'), ('Jelly', 'RB'), Tree('GPE', [('Bean', 'N
NP')]), ("''", "''"), ('and', 'CC'), ('``', '``'), ('Ice', 'NNP'), ('Cream',
'NNP'), ('Sandwich', 'NNP'), ("''", "''"), ('operating', 'VBG'), ('systems',
'NNS'), (',', ','), ('which', 'WDT'), Tree('PERSON', [('Apple', 'NNP')]),
('claims', 'VBZ'), ('infringe', 'VB'), ('its', 'PRP$'), ('patents', 'NNS'),
('.', '.')]), Tree('S', [('The', 'DT'), ('six', 'CD'), ('phones', 'NNS'),
('and', 'CC'), ('tablets', 'NNS'), ('affected', 'VBN'), ('are', 'VBP'), ('th
e', 'DT'), Tree('ORGANIZATION', [('Galaxy', 'NNP')]), ('S', 'NNP'), ('III',
'NNP'), (',', ','), ('running', 'VBG'), ('the', 'DT'), ('new', 'JJ'), Tree
('PERSON', [('Jelly', 'NNP'), ('Bean', 'NNP')]), ('system', 'NN'), (',',
','), ('the', 'DT'), Tree('ORGANIZATION', [('Galaxy', 'NNP')]), ('Tab', 'NN
P'), ('8.9', 'CD'), ('Wifi', 'NNP'), ('tablet', 'NN'), (',', ','), ('the',
'DT'), Tree('ORGANIZATION', [('Galaxy', 'NNP')]), ('Tab', 'NNP'), ('2', 'C
D'), ('10.1', 'CD'), (',', ','), Tree('PERSON', [('Galaxy', 'NNP'), ('Rugb
y', 'NNP'), ('Pro', 'NNP')]), ('and', 'CC'), Tree('PERSON', [('Galaxy', 'NN
P'), ('S', 'NNP')]), ('III', 'NNP'), ('mini', 'NN'), ('.', '.')]), Tree('S',
[Tree('PERSON', [('Apple', 'NNP')]), ('stated', 'VBD'), ('it', 'PRP'), ('ha
d', 'VBD'), ('“', 'NNP'), ('acted', 'VBD'), ('quickly', 'RB'), ('and', 'C
C'), ('diligently', 'RB'), ("''", "''"), ('in', 'IN'), ('order', 'NN'), ('t
o', 'TO'), ('``', '``'), ('determine', 'VB'), ('that', 'IN'), ('these', 'D
T'), ('newly', 'RB'), ('released', 'VBN'), ('products', 'NNS'), ('do', 'VB
P'), ('infringe', 'VB'), ('many', 'JJ'), ('of', 'IN'), ('the', 'DT'), ('sam
e', 'JJ'), ('claims', 'NNS'), ('already', 'RB'), ('asserted', 'VBN'), ('by',
'IN'), Tree('PERSON', [('Apple', 'NNP')]), ('.', '.'), ("''", "''")]), Tree
('S', [('In', 'IN'), Tree('GPE', [('August', 'NNP')]), (',', ','), Tree('PER
SON', [('Samsung', 'NNP')]), ('lost', 'VBD'), ('a', 'DT'), Tree('GSP', [('U
S', 'NNP')]), ('patent', 'NN'), ('case', 'NN'), ('to', 'TO'), Tree('GPE',
[('Apple', 'NNP')]), ('and', 'CC'), ('was', 'VBD'), ('ordered', 'VBN'), ('t
o', 'TO'), ('pay', 'VB'), ('its', 'PRP$'), ('rival', 'JJ'), ('$', '$'), ('1.
05bn', 'CD'), ('(', '('), ('£0.66bn', 'NN'), (')', ')'), ('in', 'IN'), ('dam
ages', 'NNS'), ('for', 'IN'), ('copying', 'VBG'), ('features', 'NNS'), ('o
f', 'IN'), ('the', 'DT'), Tree('ORGANIZATION', [('iPad', 'NN')]), ('and', 'C
C'), Tree('ORGANIZATION', [('iPhone', 'NN')]), ('in', 'IN'), ('its', 'PRP
$'), Tree('GPE', [('Galaxy', 'NNP')]), ('range', 'NN'), ('of', 'IN'), ('devi
ces', 'NNS'), ('.', '.')]), Tree('S', [Tree('GPE', [('Samsung', 'NNP')]),
(',', ','), ('which', 'WDT'), ('is', 'VBZ'), ('the', 'DT'), ('world', 'NN'),
("'s", 'POS'), ('top', 'JJ'), ('mobile', 'NN'), ('phone', 'NN'), ('maker',
'NN'), (',', ','), ('is', 'VBZ'), ('appealing', 'VBG'), ('the', 'DT'), ('rul
ing', 'NN'), ('.', '.')]), Tree('S', [('A', 'DT'), ('similar', 'JJ'), ('cas
e', 'NN'), ('in', 'IN'), ('the', 'DT'), Tree('ORGANIZATION', [('UK', 'NN
P')]), ('found', 'VBD'), ('in', 'IN'), Tree('GPE', [('Samsung', 'NNP')]),
("'s", 'POS'), ('favour', 'NN'), ('and', 'CC'), ('ordered', 'VBD'), Tree('PE
RSON', [('Apple', 'NNP')]), ('to', 'TO'), ('publish', 'VB'), ('an', 'DT'),
('apology', 'NN'), ('making', 'VBG'), ('clear', 'JJ'), ('that', 'IN'), ('th
e', 'DT'), Tree('LOCATION', [('South', 'JJ'), ('Korean', 'JJ')]), ('firm',
'NN'), ('had', 'VBD'), ('not', 'RB'), ('copied', 'VBN'), ('its', 'PRP$'),
('iPad', 'NN'), ('when', 'WRB'), ('designing', 'VBG'), ('its', 'PRP$'), ('ow
n', 'JJ'), ('devices', 'NNS'), ('.', '.')])]


spaCy:

[[('San Jose', 'GPE')], [('California', 'GPE')], [('November 23', 'DATE')],
[('six', 'CARDINAL')], [('Samsung', 'ORG')], [('the "Jelly Bean"', 'LAW')],
[('Ice Cream Sandwich', 'WORK_OF_ART')], [('Apple', 'ORG')], [('six', 'CARDI
NAL')], [('the Galaxy S III', 'GPE')], [('Jelly Bean', 'ORG')], [('Tab 8.9',
'PRODUCT')], [('Wifi', 'PERSON')], [('2 10.1', 'DATE')], [('Rugby Pro', 'PER
SON')], [('Galaxy S', 'PERSON')], [('Apple', 'ORG')], [('Apple', 'ORG')],
[('August', 'DATE')], [('Samsung', 'ORG')], [('US', 'GPE')], [('Apple', 'OR
G')], [('1.05bn', 'MONEY')], [('0.66bn', 'MONEY')], [('iPad', 'LOC')], [('iP
hone', 'ORG')], [('Galaxy', 'ORG')], [('Samsung', 'ORG')], [('UK', 'GPE')],
[('Samsung', 'ORG')], [('Apple', 'ORG')], [('South Korean', 'NORP')]]
```

Even though we used the same text for both methods, spaCy produced a shorter answer. NLTK processed every single word seperately. However, spaCy only included some of the text. We think that NLTK performs better since it gives the more detailed analyze of the processed text.

[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.

ORIGINAL 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.

Constituency parsing using NLTK

(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) ./

Dependency parsing using spaCy

Please see the attachment below for dependency parsing schema.

https://drive.google.com/drive/folders/1VtlAWwEKSyfrklDA2rr3RsOnLrhusM79?usp=sharing

The DET DT six NUM CD phones NOUN NNS and CCONJ CC tablets NOUN NNS affected VERB VBN are AUX VBP the DET DT Galaxy PROPN NNP S PROPN NNP III PROPN NNP , PUNCT , running VERB VBG the DET DT new ADJ JJ Jelly PROPN NNP Bean PROPN NNP system NOUN NN , PUNCT , the DET DT Galaxy PROPN NNP Tab PROPN NNP 8.9 NUM CD Wifi PROPN NNP tablet NOUN NN , PUNCT , the DET DT Galaxy PROPN NNP Tab PROPN NNP 2 NUM CD 10.1 NUM CD , PUNCT , Galaxy PROPN NNP Rugby PROPN NNP Pro PROPN NNP and CCONJ CC Galaxy PROPN NNP S PROPN NNP III PROPN NNP mini NOUN NN . PUNCT .

Both methods produced quite similar outputs. The difference is spaCy gave extra explanation about some words such as “Jelly PROPN NNP” and “Jelly/NNP” in NLTK.

By identifying each word as a node and showing linkages to its dependents, dependency parsing defines the grammatical structure of a phrase. A constituency parsed tree uses context-free grammar to show the syntactic structure of a sentence. As opposed to dependency parsing, which uses dependency grammar.

End of this notebook