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 noteboo
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))
```

```
C:\Users\User\Desktop\Text mining\Text_Mining_Group45\lab_sessions\lab1\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 1142
```

[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) # sentence splitting
```

In [6]:
```
tokens_per_sentence = []
for sentence_nltk in sentences_nltk:
    sent_tokens = word_tokenize(sentence_nltk) # tokenization
    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 the new Jelly B
ean system, the Galaxy Tab 8.9 Wifi tablet, the Galaxy Tab 2 10.1, Galaxy Rugby Pro and Ga
laxy S III mini.
TOKENS ['The', 'six', 'phones', 'and', 'tablets', 'affected', 'are', 'the', 'Galaxy', 'S',
'III', ',', 'running', 'the', 'new', 'Jelly', 'Bean', 'system', ',', 'the', 'Galaxy', 'Ta
b', '8.9', 'Wifi', 'tablet', ',', 'the', 'Galaxy', 'Tab', '2', '10.1', ',', 'Galaxy', 'Rug
by', '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]:
```
from nltk import pos_tag
```

In [9]:
```
pos_tags_per_sentence = []
for tokens in tokens_per_sentence:
    tagged_token = pos_tag(tokens) #tag the tokens
    pos_tags_per_sentence.append(tagged_token) # add the tagged tokens to pos_tags_per_se
    print(tagged_token)
```

```
[('https', 'NN'), (':', ':'), ('//www.telegraph.co.uk/technology/apple/9702716/Apple-Samsu
ng-lawsuit-six-more-products-under-scrutiny.html', 'JJ'), ('Documents', 'NNS'), ('filed',
'VBN'), ('to', 'TO'), ('the', 'DT'), ('San', 'NNP'), ('Jose', 'NNP'), ('federal', 'JJ'),
('court', 'NN'), ('in', 'IN'), ('California', 'NNP'), ('on', 'IN'), ('November', 'NNP'),
('23', 'CD'), ('list', 'NN'), ('six', 'CD'), ('Samsung', 'NNP'), ('products', 'NNS'), ('ru
nning', 'VBG'), ('the', 'DT'), ('``', '``'), ('Jelly', 'RB'), ('Bean', 'NNP'), ("''",
"''"), ('and', 'CC'), ('``', '``'), ('Ice', 'NNP'), ('Cream', 'NNP'), ('Sandwich', 'NNP'),
("''", "''"), ('operating', 'VBG'), ('systems', 'NNS'), (',', ','), ('which', 'WDT'), ('Ap
ple', 'NNP'), ('claims', 'VBZ'), ('infringe', 'VB'), ('its', 'PRP$'), ('patents', 'NNS'),
('.', '.')]
[('The', 'DT'), ('six', 'CD'), ('phones', 'NNS'), ('and', 'CC'), ('tablets', 'NNS'), ('aff
ected', 'VBN'), ('are', 'VBP'), ('the', 'DT'), ('Galaxy', 'NNP'), ('S', 'NNP'), ('III', 'N
NP'), (',', ','), ('running', 'VBG'), ('the', 'DT'), ('new', 'JJ'), ('Jelly', 'NNP'), ('Be
an', 'NNP'), ('system', 'NN'), (',', ','), ('the', 'DT'), ('Galaxy', 'NNP'), ('Tab', 'NN
P'), ('8.9', 'CD'), ('Wifi', 'NNP'), ('tablet', 'NN'), (',', ','), ('the', 'DT'), ('Galax
y', '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'), ('â€œacted', 'VBN'),
('quickly', 'RB'), ('and', 'CC'), ('diligently', 'RB'), ("''", "''"), ('in', 'IN'), ('orde
r', 'NN'), ('to', 'TO'), ('``', '``'), ('determine', 'VB'), ('that', 'IN'), ('these', 'D
T'), ('newly', 'RB'), ('released', 'VBN'), ('products', 'NNS'), ('do', 'VBP'), ('infring
e', 'VB'), ('many', 'JJ'), ('of', 'IN'), ('the', 'DT'), ('same', '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', 'VB'), ('its', 'P
RP$'), ('rival', 'JJ'), ('$', '$'), ('1.05bn', 'CD'), ('(', '('), ('Â£0.66bn', 'NN'),
(')', ')'), ('in', 'IN'), ('damages', 'NNS'), ('for', 'IN'), ('copying', 'VBG'), ('feature
s', 'NNS'), ('of', 'IN'), ('the', 'DT'), ('iPad', 'NN'), ('and', 'CC'), ('iPhone', 'NN'),
('in', 'IN'), ('its', 'PRP$'), ('Galaxy', 'NNP'), ('range', 'NN'), ('of', 'IN'), ('device
s', 'NNS'), ('.', '.')]
[('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'), ('.',
'.')]
[('A', 'DT'), ('similar', 'JJ'), ('case', 'NN'), ('in', 'IN'), ('the', 'DT'), ('UK', 'NN
P'), ('found', 'VBD'), ('in', 'IN'), ('Samsung', 'NNP'), ("'s", 'POS'), ('favour', 'NN'),
('and', 'CC'), ('ordered', 'VBD'), ('Apple', 'NNP'), ('to', 'TO'), ('publish', 'VB'), ('a
n', 'DT'), ('apology', 'NN'), ('making', 'VBG'), ('clear', 'JJ'), ('that', 'IN'), ('the',
'DT'), ('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 [10]:

```
print(pos_tags_per_sentence)
```

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

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

In [12]:
```
ner_tags_per_sentence = []
for tagged_tokens in pos_tags_per_sentence:
    # ne_chunk requires POS-tagged tokens
    tokens_pos_tagged_and_named_entities = ne_chunk(tagged_tokens) # apply NER to to tagge
    ner_tags_per_sentence.append(tokens_pos_tagged_and_named_entities)
    print()
    print('NAMED ENTITY RECOGNITION OUTPUT', tokens_pos_tagged_and_named_entities)
```

```
NAMED ENTITY RECOGNITION OUTPUT (S
  https/NN
  :/:
  //www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more-products-u
nder-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
  ./.)

NAMED ENTITY RECOGNITION OUTPUT (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
  ./.)

NAMED ENTITY RECOGNITION OUTPUT (S
  (PERSON Apple/NNP)
  stated/VBD
  it/PRP
  had/VBD
  â€œacted/VBN
  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)
  ./.
  ''/'')

NAMED ENTITY RECOGNITION OUTPUT (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
  (/(
  Â£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
  ./.)

NAMED ENTITY RECOGNITION OUTPUT (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
  ./.)

NAMED ENTITY RECOGNITION OUTPUT (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'), ('Jos
e', 'NNP')]), ('federal', 'JJ'), ('court', 'NN'), ('in', 'IN'), Tree('GPE', [('Californi
a', 'NNP')]), ('on', 'IN'), ('November', 'NNP'), ('23', 'CD'), ('list', 'NN'), ('six', 'C
D'), Tree('ORGANIZATION', [('Samsung', 'NNP')]), ('products', 'NNS'), ('running', 'VBG'),
('the', 'DT'), ('``', '``'), ('Jelly', 'RB'), Tree('GPE', [('Bean', 'NNP')]), ("''",
"''"), ('and', 'CC'), ('``', '``'), ('Ice', 'NNP'), ('Cream', 'NNP'), ('Sandwich', 'NNP'),
("''", "''"), ('operating', 'VBG'), ('systems', 'NNS'), (',', ','), ('which', 'WDT'), Tree
('PERSON', [('Apple', 'NNP')]), ('claims', 'VBZ'), ('infringe', 'VB'), ('its', 'PRP$'),
('patents', 'NNS'), ('.', '.')]), Tree('S', [('The', 'DT'), ('six', 'CD'), ('phones', 'NN
S'), ('and', 'CC'), ('tablets', 'NNS'), ('affected', 'VBN'), ('are', 'VBP'), ('the', 'D
T'), Tree('ORGANIZATION', [('Galaxy', 'NNP')]), ('S', 'NNP'), ('III', 'NNP'), (',', ','),
('running', 'VBG'), ('the', 'DT'), ('new', 'JJ'), Tree('PERSON', [('Jelly', 'NNP'), ('Bea
n', 'NNP')]), ('system', 'NN'), (',', ','), ('the', 'DT'), Tree('ORGANIZATION', [('Galax
y', 'NNP')]), ('Tab', 'NNP'), ('8.9', 'CD'), ('Wifi', 'NNP'), ('tablet', 'NN'), (',',
','), ('the', 'DT'), Tree('ORGANIZATION', [('Galaxy', 'NNP')]), ('Tab', 'NNP'), ('2', 'C
D'), ('10.1', 'CD'), (',', ','), Tree('PERSON', [('Galaxy', 'NNP'), ('Rugby', 'NNP'), ('Pr
o', 'NNP')]), ('and', 'CC'), Tree('PERSON', [('Galaxy', 'NNP'), ('S', 'NNP')]), ('III', 'N
NP'), ('mini', 'NN'), ('.', '.')]), Tree('S', [Tree('PERSON', [('Apple', 'NNP')]), ('state
d', 'VBD'), ('it', 'PRP'), ('had', 'VBD'), ('â€œacted', 'VBN'), ('quickly', 'RB'), ('and',
'CC'), ('diligently', 'RB'), ("''", "''"), ('in', 'IN'), ('order', 'NN'), ('to', 'TO'),
('``', '``'), ('determine', 'VB'), ('that', 'IN'), ('these', 'DT'), ('newly', 'RB'), ('rel
eased', 'VBN'), ('products', 'NNS'), ('do', 'VBP'), ('infringe', 'VB'), ('many', 'JJ'),
('of', 'IN'), ('the', 'DT'), ('same', 'JJ'), ('claims', 'NNS'), ('already', 'RB'), ('asser
ted', 'VBN'), ('by', 'IN'), Tree('PERSON', [('Apple', 'NNP')]), ('.', '.'), ("''",
"''")]), Tree('S', [('In', 'IN'), Tree('GPE', [('August', 'NNP')]), (',', ','), Tree('PERS
ON', [('Samsung', 'NNP')]), ('lost', 'VBD'), ('a', 'DT'), Tree('GSP', [('US', 'NNP')]),
('patent', 'NN'), ('case', 'NN'), ('to', 'TO'), Tree('GPE', [('Apple', 'NNP')]), ('and',
'CC'), ('was', 'VBD'), ('ordered', 'VBN'), ('to', 'TO'), ('pay', 'VB'), ('its', 'PRP$'),
('rival', 'JJ'), ('$', '$'), ('1.05bn', 'CD'), ('(', '('), ('Â£0.66bn', 'NN'), (')', ')'),
('in', 'IN'), ('damages', 'NNS'), ('for', 'IN'), ('copying', 'VBG'), ('features', 'NNS'),
('of', 'IN'), ('the', 'DT'), Tree('ORGANIZATION', [('iPad', 'NN')]), ('and', 'CC'), Tree
('ORGANIZATION', [('iPhone', 'NN')]), ('in', 'IN'), ('its', 'PRP$'), Tree('GPE', [('Galax
y', 'NNP')]), ('range', 'NN'), ('of', 'IN'), ('devices', 'NNS'), ('.', '.')]), Tree('S',
[Tree('GPE', [('Samsung', 'NNP')]), (',', ','), ('which', 'WDT'), ('is', 'VBZ'), ('the',
'DT'), ('world', 'NN'), ("'s", 'POS'), ('top', 'JJ'), ('mobile', 'NN'), ('phone', 'NN'),
('maker', 'NN'), (',', ','), ('is', 'VBZ'), ('appealing', 'VBG'), ('the', 'DT'), ('rulin
g', 'NN'), ('.', '.')]), Tree('S', [('A', 'DT'), ('similar', 'JJ'), ('case', 'NN'), ('in',
'IN'), ('the', 'DT'), Tree('ORGANIZATION', [('UK', 'NNP')]), ('found', 'VBD'), ('in', 'I
N'), Tree('GPE', [('Samsung', 'NNP')]), ("'s", 'POS'), ('favour', 'NN'), ('and', 'CC'),
('ordered', 'VBD'), Tree('PERSON', [('Apple', 'NNP')]), ('to', 'TO'), ('publish', 'VB'),
('an', 'DT'), ('apology', 'NN'), ('making', 'VBG'), ('clear', 'JJ'), ('that', 'IN'), ('th
e', 'DT'), Tree('LOCATION', [('South', 'JJ'), ('Korean', 'JJ')]), ('firm', 'NN'), ('had',
'VBD'), ('not', 'RB'), ('copied', 'VBN'), ('its', 'PRP$'), ('iPad', 'NN'), ('when', '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 [14]:
```
# Define the grammar
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 = []
for tagged_tokens in pos_tags_per_sentence:
    constituent_structure = constituent_parser.parse(tagged_tokens) # constituency parse 
    constituency_output_per_sentence.append(constituent_structure)
    print()
    print(constituent_structure)
    constituent_structure.draw()
```

```
(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))
    (VP (V â€œacted/VBN))
    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 Â£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 [16]:
```
print(constituency_output_per_sentence)
```

```
[Tree('S', [Tree('NP', [('https', 'NN')]), (':', ':'), Tree('NP', [('//www.telegraph.co.u
k/technology/apple/9702716/Apple-Samsung-lawsuit-six-more-products-under-scrutiny.html',
'JJ')]), ('Documents', 'NNS'), Tree('VP', [Tree('V', [('filed', 'VBN')])]), ('to', 'TO'),
Tree('NP', [('the', 'DT')]), ('San', 'NNP'), ('Jose', 'NNP'), Tree('NP', [('federal', 'J
J'), ('court', 'NN')]), Tree('P', [('in', 'IN')]), ('California', 'NNP'), Tree('P', [('o
n', 'IN')]), ('November', 'NNP'), ('23', 'CD'), Tree('NP', [('list', 'NN')]), ('six', 'C
D'), ('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('VP', [Tree('V', [('operating', 'VBG')])]), ('systems', 'NNS'), (',',
','), ('which', 'WDT'), ('Apple', 'NNP'), Tree('VP', [Tree('V', [('claims', 'VBZ')])]), Tr
ee('VP', [Tree('V', [('infringe', 'VB')])]), ('its', 'PRP$'), ('patents', 'NNS'), ('.',
'.')]), Tree('S', [Tree('NP', [('The', 'DT')]), ('six', 'CD'), ('phones', 'NNS'), ('and',
'CC'), ('tablets', 'NNS'), Tree('VP', [Tree('V', [('affected', 'VBN')])]), Tree('VP', [Tre
e('V', [('are', 'VBP')]), Tree('NP', [('the', 'DT')])]), ('Galaxy', 'NNP'), ('S', 'NNP'),
('III', 'NNP'), (',', ','), Tree('VP', [Tree('V', [('running', 'VBG')]), Tree('NP', [('th
e', 'DT'), ('new', 'JJ')])]), ('Jelly', 'NNP'), ('Bean', 'NNP'), Tree('NP', [('system', 'N
N')]), (',', ','), 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'), (',', ','), ('Galax
y', 'NNP'), ('Rugby', 'NNP'), ('Pro', 'NNP'), ('and', 'CC'), ('Galaxy', 'NNP'), ('S', 'NN
P'), ('III', 'NNP'), Tree('NP', [('mini', 'NN')]), ('.', '.')]), Tree('S', [('Apple', 'NN
P'), Tree('VP', [Tree('V', [('stated', 'VBD')])]), ('it', 'PRP'), Tree('VP', [Tree('V',
[('had', 'VBD')])]), Tree('VP', [Tree('V', [('â€œacted', 'VBN')])]), ('quickly', 'RB'),
('and', 'CC'), ('diligently', 'RB'), ("''", "''"), Tree('PP', [Tree('P', [('in', 'IN')]),
Tree('NP', [('order', 'NN')])]), ('to', 'TO'), ('``', '``'), Tree('VP', [Tree('V', [('dete
rmine', 'VB')]), Tree('PP', [Tree('P', [('that', 'IN')]), Tree('NP', [('these', 'D
T')])])]), ('newly', 'RB'), Tree('VP', [Tree('V', [('released', 'VBN')])]), ('products',
'NNS'), Tree('VP', [Tree('V', [('do', 'VBP')])]), Tree('VP', [Tree('V', [('infringe', 'V
B')]), Tree('NP', [('many', 'JJ')]), Tree('PP', [Tree('P', [('of', 'IN')]), Tree('NP',
[('the', 'DT'), ('same', 'JJ')])])]), ('claims', 'NNS'), ('already', 'RB'), Tree('VP', [Tr
ee('V', [('asserted', 'VBN')])]), Tree('P', [('by', 'IN')]), ('Apple', 'NNP'), ('.', '.'),
("''", "''")]), Tree('S', [Tree('P', [('In', 'IN')]), ('August', 'NNP'), (',', ','), ('Sam
sung', 'NNP'), Tree('VP', [Tree('V', [('lost', 'VBD')]), Tree('NP', [('a', 'DT')])]), ('U
S', 'NNP'), Tree('NP', [('patent', 'NN'), ('case', 'NN')]), ('to', 'TO'), ('Apple', 'NN
P'), ('and', 'CC'), Tree('VP', [Tree('V', [('was', 'VBD')])]), Tree('VP', [Tree('V', [('or
dered', 'VBN')])]), ('to', 'TO'), Tree('VP', [Tree('V', [('pay', 'VB')])]), ('its', 'PRP
$'), Tree('NP', [('rival', 'JJ')]), ('$', '$'), ('1.05bn', 'CD'), ('(', '('), Tree('NP',
[('Â£0.66bn', 'NN')]), (')', ')'), Tree('P', [('in', 'IN')]), ('damages', 'NNS'), Tree
('P', [('for', 'IN')]), Tree('VP', [Tree('V', [('copying', 'VBG')])]), ('features', 'NN
S'), Tree('PP', [Tree('P', [('of', 'IN')]), Tree('NP', [('the', 'DT'), ('iPad', 'NN')])]),
('and', 'CC'), Tree('NP', [('iPhone', 'NN')]), Tree('P', [('in', 'IN')]), ('its', 'PRP$'),
('Galaxy', 'NNP'), Tree('NP', [('range', 'NN')]), Tree('P', [('of', 'IN')]), ('devices',
'NNS'), ('.', '.')]), Tree('S', [('Samsung', 'NNP'), (',', ','), ('which', 'WDT'), Tree('V
P', [Tree('V', [('is', 'VBZ')]), Tree('NP', [('the', 'DT'), ('world', 'NN')])]), ("'s", 'P
OS'), Tree('NP', [('top', 'JJ'), ('mobile', 'NN'), ('phone', 'NN'), ('maker', 'NN')]),
(',', ','), Tree('VP', [Tree('V', [('is', 'VBZ')])]), Tree('VP', [Tree('V', [('appealing',
'VBG')]), Tree('NP', [('the', 'DT'), ('ruling', 'NN')])]), ('.', '.')]), Tree('S', [Tree
('NP', [('A', 'DT'), ('similar', 'JJ'), ('case', 'NN')]), Tree('PP', [Tree('P', [('in', 'I
N')]), Tree('NP', [('the', 'DT')])]), ('UK', 'NNP'), Tree('VP', [Tree('V', [('found', 'VB
D')])]), Tree('P', [('in', 'IN')]), ('Samsung', 'NNP'), ("'s", 'POS'), Tree('NP', [('favou
r', 'NN')]), ('and', 'CC'), Tree('VP', [Tree('V', [('ordered', 'VBD')])]), ('Apple', 'NN
P'), ('to', 'TO'), Tree('VP', [Tree('V', [('publish', 'VB')]), Tree('NP', [('an', 'DT'),
('apology', 'NN')])]), Tree('VP', [Tree('V', [('making', 'VBG')]), Tree('NP', [('clear',
'JJ')]), Tree('PP', [Tree('P', [('that', 'IN')]), Tree('NP', [('the', 'DT'), ('South', 'J
J'), ('Korean', 'JJ'), ('firm', 'NN')])])]), Tree('VP', [Tree('V', [('had', 'VBD')])]),
('not', 'RB'), Tree('VP', [Tree('V', [('copied', 'VBN')])]), ('its', 'PRP$'), Tree('NP',
[('iPad', 'NN')]), ('when', 'WRB'), Tree('VP', [Tree('V', [('designing', 'VBG')])]), ('it
s', 'PRP$'), 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 [17]:
```
# * -> 0 or more
# + -> 1 or more
# ? -> Optional

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: {<NNP>+<CD>?}  # NEP -> One or more NNP(Proper noun, singular) optionally followed by
```

In [18]:
```
constituency_v2_output_per_sentence = []
for tagged_tokens in pos_tags_per_sentence:
    constituent_structure = constituent_parser_v2.parse(tagged_tokens)
    constituency_v2_output_per_sentence.append(constituent_structure)
    print()
    print(constituent_structure)
    constituent_structure.draw()
```

```
(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)
  (NEP San/NNP Jose/NNP)
  (NP federal/JJ court/NN)
  (P in/IN)
  (NEP California/NNP)
  (P on/IN)
  (NEP November/NNP 23/CD)
  (NP list/NN)
  six/CD
  (NEP Samsung/NNP)
  products/NNS
  (VP (V running/VBG) (NP the/DT))
  ``/``
  Jelly/RB
  (NEP Bean/NNP)
  ''/''
  and/CC
  ``/``
  (NEP Ice/NNP Cream/NNP Sandwich/NNP)
  ''/''
  (VP (V operating/VBG))
  systems/NNS
  ,/,
  which/WDT
  (NEP 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))
  (NEP Galaxy/NNP S/NNP III/NNP)
  ,/,
  (VP (V running/VBG) (NP the/DT new/JJ))
  (NEP Jelly/NNP Bean/NNP)
  (NP system/NN)
  ,/,
  (NP the/DT)
  (NEP Galaxy/NNP Tab/NNP 8.9/CD)
  (NEP Wifi/NNP)
  (NP tablet/NN)
  ,/,
  (NP the/DT)
  (NEP Galaxy/NNP Tab/NNP 2/CD)
  10.1/CD
  ,/,
  (NEP Galaxy/NNP Rugby/NNP Pro/NNP)
  and/CC
  (NEP Galaxy/NNP S/NNP III/NNP)
  (NP mini/NN)
  ./.)

(S
  (NEP Apple/NNP)
  (VP (V stated/VBD))
  it/PRP
  (VP (V had/VBD))
  (VP (V â€œacted/VBN))
  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)
  (NEP Apple/NNP)
  ./.
  ''/'')

(S
  (P In/IN)
  (NEP August/NNP)
  ,/,
  (NEP Samsung/NNP)
  (VP (V lost/VBD) (NP a/DT))
  (NEP US/NNP)
  (NP patent/NN case/NN)
  to/TO
  (NEP 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 Â£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$
  (NEP Galaxy/NNP)
  (NP range/NN)
  (P of/IN)
  devices/NNS
  ./.)

(S
  (NEP 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))
  (NEP UK/NNP)
  (VP (V found/VBD))
  (P in/IN)
  (NEP Samsung/NNP)
  's/POS
  (NP favour/NN)
  and/CC
  (VP (V ordered/VBD))
  (NEP 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 [19]:

```python
print(constituency_v2_output_per_sentence)
## these were the output for  the exaples: Galaxy S III and Ice Cream Sandwich
# Tree('NEP', [('Galaxy', 'NNP'), ('S', 'NNP'), ('III', 'NNP')])
# Tree('NEP', [('Ice', 'NNP'), ('Cream', 'NNP'), ('Sandwich', 'NNP')]), ("''", "''")
```

```
[Tree('S', [Tree('NP', [('https', 'NN')]), (':', ':'), Tree('NP', [('//www.telegraph.co.u
k/technology/apple/9702716/Apple-Samsung-lawsuit-six-more-products-under-scrutiny.html',
'JJ')]), ('Documents', 'NNS'), Tree('VP', [Tree('V', [('filed', 'VBN')])]), ('to', 'TO'),
Tree('NP', [('the', 'DT')]), Tree('NEP', [('San', 'NNP'), ('Jose', 'NNP')]), Tree('NP',
[('federal', 'JJ'), ('court', 'NN')]), Tree('P', [('in', 'IN')]), Tree('NEP', [('Californi
a', 'NNP')]), Tree('P', [('on', 'IN')]), Tree('NEP', [('November', 'NNP'), ('23', 'CD')]),
Tree('NP', [('list', 'NN')]), ('six', 'CD'), Tree('NEP', [('Samsung', 'NNP')]), ('product
s', 'NNS'), Tree('VP', [Tree('V', [('running', 'VBG')]), Tree('NP', [('the', 'DT')])]),
('``', '``'), ('Jelly', 'RB'), Tree('NEP', [('Bean', 'NNP')]), ("''", "''"), ('and', 'C
C'), ('``', '``'), Tree('NEP', [('Ice', 'NNP'), ('Cream', 'NNP'), ('Sandwich', 'NNP')]),
("''", "''"), Tree('VP', [Tree('V', [('operating', 'VBG')])]), ('systems', 'NNS'), (',',
','), ('which', 'WDT'), Tree('NEP', [('Apple', 'NNP')]), Tree('VP', [Tree('V', [('claims',
'VBZ')])]), Tree('VP', [Tree('V', [('infringe', 'VB')])]), ('its', 'PRP$'), ('patents', 'N
NS'), ('.', '.')]), Tree('S', [Tree('NP', [('The', 'DT')]), ('six', 'CD'), ('phones', 'NN
S'), ('and', 'CC'), ('tablets', 'NNS'), Tree('VP', [Tree('V', [('affected', 'VBN')])]), Tr
ee('VP', [Tree('V', [('are', 'VBP')]), Tree('NP', [('the', 'DT')])]), Tree('NEP', [('Galax
y', 'NNP'), ('S', 'NNP'), ('III', 'NNP')]), (',', ','), Tree('VP', [Tree('V', [('running',
'VBG')]), Tree('NP', [('the', 'DT'), ('new', 'JJ')])]), Tree('NEP', [('Jelly', 'NNP'), ('B
ean', 'NNP')]), Tree('NP', [('system', 'NN')]), (',', ','), Tree('NP', [('the', 'DT')]), T
ree('NEP', [('Galaxy', 'NNP'), ('Tab', 'NNP'), ('8.9', 'CD')]), Tree('NEP', [('Wifi', 'NN
P')]), Tree('NP', [('tablet', 'NN')]), (',', ','), Tree('NP', [('the', 'DT')]), Tree('NE
P', [('Galaxy', 'NNP'), ('Tab', 'NNP'), ('2', 'CD')]), ('10.1', 'CD'), (',', ','), Tree('N
EP', [('Galaxy', 'NNP'), ('Rugby', 'NNP'), ('Pro', 'NNP')]), ('and', 'CC'), Tree('NEP',
[('Galaxy', 'NNP'), ('S', 'NNP'), ('III', 'NNP')]), Tree('NP', [('mini', 'NN')]), ('.',
'.')]), Tree('S', [Tree('NEP', [('Apple', 'NNP')]), Tree('VP', [Tree('V', [('stated', 'VB
D')])]), ('it', 'PRP'), Tree('VP', [Tree('V', [('had', 'VBD')])]), Tree('VP', [Tree('V',
[('â€œacted', 'VBN')])]), ('quickly', 'RB'), ('and', 'CC'), ('diligently', 'RB'), ("''",
"''"), Tree('PP', [Tree('P', [('in', 'IN')]), Tree('NP', [('order', 'NN')])]), ('to', 'T
O'), ('``', '``'), Tree('VP', [Tree('V', [('determine', 'VB')]), Tree('PP', [Tree('P',
[('that', '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')]), Tree
('PP', [Tree('P', [('of', 'IN')]), Tree('NP', [('the', 'DT'), ('same', 'JJ')])])]), ('clai
ms', 'NNS'), ('already', 'RB'), Tree('VP', [Tree('V', [('asserted', 'VBN')])]), Tree('P',
[('by', 'IN')]), Tree('NEP', [('Apple', 'NNP')]), ('.', '.'), ("''", "''")]), Tree('S', [T
ree('P', [('In', 'IN')]), Tree('NEP', [('August', 'NNP')]), (',', ','), Tree('NEP', [('Sam
sung', 'NNP')]), Tree('VP', [Tree('V', [('lost', 'VBD')]), Tree('NP', [('a', 'DT')])]), Tr
ee('NEP', [('US', 'NNP')]), Tree('NP', [('patent', 'NN'), ('case', 'NN')]), ('to', 'TO'),
Tree('NEP', [('Apple', 'NNP')]), ('and', 'CC'), Tree('VP', [Tree('V', [('was', 'VBD')])]),
Tree('VP', [Tree('V', [('ordered', 'VBN')])]), ('to', 'TO'), Tree('VP', [Tree('V', [('pa
y', 'VB')])]), ('its', 'PRP$'), Tree('NP', [('rival', 'JJ')]), ('$', '$'), ('1.05bn', 'C
D'), ('(', '('), Tree('NP', [('Â£0.66bn', 'NN')]), (')', ')'), Tree('P', [('in', 'IN')]),
('damages', 'NNS'), Tree('P', [('for', 'IN')]), Tree('VP', [Tree('V', [('copying', 'VB
G')])]), ('features', 'NNS'), Tree('PP', [Tree('P', [('of', 'IN')]), Tree('NP', [('the',
'DT'), ('iPad', 'NN')])]), ('and', 'CC'), Tree('NP', [('iPhone', 'NN')]), Tree('P', [('i
n', 'IN')]), ('its', 'PRP$'), Tree('NEP', [('Galaxy', 'NNP')]), Tree('NP', [('range', 'N
N')]), Tree('P', [('of', 'IN')]), ('devices', 'NNS'), ('.', '.')]), Tree('S', [Tree('NEP',
[('Samsung', 'NNP')]), (',', ','), ('which', 'WDT'), Tree('VP', [Tree('V', [('is', 'VB
Z')]), Tree('NP', [('the', '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', 'VBG')]), Tree('NP', [('th
e', 'DT'), ('ruling', 'NN')])]), ('.', '.')]), Tree('S', [Tree('NP', [('A', 'DT'), ('simil
ar', 'JJ'), ('case', 'NN')]), Tree('PP', [Tree('P', [('in', 'IN')]), Tree('NP', [('the',
'DT')])]), Tree('NEP', [('UK', 'NNP')]), Tree('VP', [Tree('V', [('found', 'VBD')])]), Tree
('P', [('in', 'IN')]), Tree('NEP', [('Samsung', 'NNP')]), ("'s", 'POS'), Tree('NP', [('fav
our', 'NN')]), ('and', 'CC'), Tree('VP', [Tree('V', [('ordered', 'VBD')])]), Tree('NEP',
[('Apple', 'NNP')]), ('to', 'TO'), Tree('VP', [Tree('V', [('publish', 'VB')]), Tree('NP',
[('an', 'DT'), ('apology', 'NN')])]), Tree('VP', [Tree('V', [('making', 'VBG')]), Tree('N
P', [('clear', 'JJ')]), Tree('PP', [Tree('P', [('that', 'IN')]), Tree('NP', [('the', 'D
T'), ('South', 'JJ'), ('Korean', 'JJ'), ('firm', 'NN')])])]), Tree('VP', [Tree('V', [('ha
d', 'VBD')])]), ('not', 'RB'), Tree('VP', [Tree('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 [20]:
```
import spacy
```

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

In [21]:
```
doc = nlp(text) # insert code here
```

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.

In [22]:
```
sents = list(doc.sents)
```

(Sentence splitting &) Tokenization, POS, NER and Constituency/dependency parsing using spaCY

Tokenization

In [23]:
```
for sentence in doc.sents:
    print()
    print(sentence)
    for token in sentence:
        print(token.text)
```

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

Documents filed to the San Jose federal court in California on November 23 list six Samsun
g products running the "Jelly Bean" and "Ice Cream Sandwich" operating systems, which Appl
e claims infringe its patents.

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




Documents
filed
to
the
San
Jose
federal
court
in
California
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 phones and tablets affected are the Galaxy S III, running the new Jelly Bean syste
m, the Galaxy Tab 8.9 Wifi tablet, the Galaxy Tab 2 10.1, Galaxy Rugby Pro and Galaxy S II
I mini.

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
.

Apple stated it had â€œacted quickly and diligently" in order to "determine that these new
ly released products do infringe many of the same claims already asserted by Apple.
Apple

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

"
In August, Samsung lost a US patent case to Apple and was ordered to pay its rival $1.05bn (Â£0.66bn) in damages for copying features of the iPad and iPhone in its Galaxy range of d
evices.
"

In
August
,
Samsung
lost
a
US
patent
case
to
Apple
and
was
ordered
to
pay
its
rival
$
1.05bn
(
Â£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 appealing the ruling.

Samsung
,
which
is
the
world
's
top
mobile
phone
maker
,
is
appealing
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 dev
ices.

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
.
```

Part of speech tagging

In [24]:
```python
# in the attribute pos_ of each Token object: The simple part-of-speech tag
#in the attribute tag_ of each Token object: The detailed part-of-speech tag

for sentence in sents:
    print()
    print(sentence)
    for token in sentence:
        print(token.text, token.pos_, token.tag_)
```

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

Documents filed to the San Jose federal court in California on November 23 list six Samsun
g products running the "Jelly Bean" and "Ice Cream Sandwich" operating systems, which Appl
e claims infringe its patents.

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


 SPACE _SP
Documents NOUN NNS
filed VERB VBD
to ADP IN
the DET DT
San PROPN NNP
Jose PROPN NNP
federal ADJ JJ
court NOUN NN
in ADP IN
California PROPN NNP
on ADP IN
November PROPN NNP
23 NUM CD
list NOUN NN
six NUM CD
Samsung PROPN NNP
products NOUN NNS
running VERB VBG
the DET DT
" PUNCT ``
Jelly PROPN NNP
Bean PROPN NNP
" PUNCT ''
and CCONJ CC
" PUNCT ``
Ice PROPN NNP
Cream PROPN NNP
Sandwich NOUN NN
" PUNCT ''
operating NOUN NN
systems NOUN NNS
, PUNCT ,
which PRON WDT
Apple PROPN NNP
claims VERB VBZ
infringe VERB VBP
its PRON PRP$
patents NOUN NNS
. PUNCT .

 SPACE _SP

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 Galaxy Tab 2 10.1, Galaxy Rugby Pro and Galaxy S II
I mini.

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 .

 SPACE _SP

Apple stated it had â€œacted quickly and diligently" in order to "determine that these new
ly released products do infringe many of the same claims already asserted by Apple.
Apple PROPN NNP
stated VERB VBD
it PRON PRP
had AUX VBD
â€œacted VERB VBN
quickly ADV RB
```

```
and CCONJ CC
diligently ADV RB
" PUNCT ''
in ADP IN
order NOUN NN
to PART TO
" PUNCT ``
determine VERB VB
that SCONJ IN
these DET DT
newly ADV RB
released VERB VBN
products NOUN NNS
do AUX VBP
infringe VERB VB
many ADJ JJ
of ADP IN
the DET DT
same ADJ JJ
claims NOUN NNS
already ADV RB
asserted VERB VBN
by ADP IN
Apple PROPN NNP
. PUNCT .

"
In August, Samsung lost a US patent case to Apple and was ordered to pay its rival $1.05bn
(Â£0.66bn) in damages for copying features of the iPad and iPhone in its Galaxy range of d
evices.
" PUNCT ''

 SPACE _SP
In ADP IN
August PROPN NNP
, PUNCT ,
Samsung PROPN NNP
lost VERB VBD
a DET DT
US PROPN NNP
patent NOUN NN
case NOUN NN
to ADP IN
Apple PROPN NNP
and CCONJ CC
was AUX VBD
ordered VERB VBN
to PART TO
pay VERB VB
its PRON PRP$
rival NOUN NN
$ SYM $
1.05bn NUM CD
( PUNCT -LRB-
Â£0.66bn PROPN NNP
) PUNCT -RRB-
in ADP IN
damages NOUN NNS
for ADP IN
copying VERB VBG
features NOUN NNS
of ADP IN
the DET DT
iPad PROPN NNP
and CCONJ CC
iPhone PROPN NNP
```

```
in ADP IN
its PRON PRP$
Galaxy PROPN NNP
range NOUN NN
of ADP IN
devices NOUN NNS
. PUNCT .

Samsung, which is the world's top mobile phone maker, is appealing the ruling.

Samsung PROPN NNP
, PUNCT ,
which PRON WDT
is AUX VBZ
the DET DT
world NOUN NN
's PART POS
top ADJ JJ
mobile ADJ JJ
phone NOUN NN
maker NOUN NN
, PUNCT ,
is AUX VBZ
appealing VERB VBG
the DET DT
ruling NOUN NN
. PUNCT .

 SPACE _SP

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 dev
ices.
A DET DT
similar ADJ JJ
case NOUN NN
in ADP IN
the DET DT
UK PROPN NNP
found VERB VBN
in ADP IN
Samsung PROPN NNP
's PART POS
favour NOUN NN
and CCONJ CC
ordered VERB VBD
Apple PROPN NNP
to PART TO
publish VERB VB
an DET DT
apology NOUN NN
making VERB VBG
clear ADJ JJ
that SCONJ IN
the DET DT
South ADJ JJ
Korean ADJ JJ
firm NOUN NN
had AUX VBD
not PART RB
copied VERB VBN
its PRON PRP$
iPad PROPN NNP
when SCONJ WRB
designing VERB VBG
its PRON PRP$
```

```
own ADJ JJ
devices NOUN NNS
. PUNCT .
```

Named Entity Recognition

In [25]:
```
spacy.displacy.render(doc, jupyter=True, style='ent')
```

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

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" operating systems, which Apple **ORG** claims infringe its patents.

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

Apple **ORG** stated it had â€œacted 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** (Â£0.66bn) in damages for copying features of the iPad **ORG** and iPhone in its Galaxy **FAC** 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 **ORG** when designing its own devices.

In [26]:
```
# The attribute label_ and an ent (of type spacy.tokens.span.Span) contains the named enti

for ent in doc.ents:
    print(ent.text, ent.label_)
```

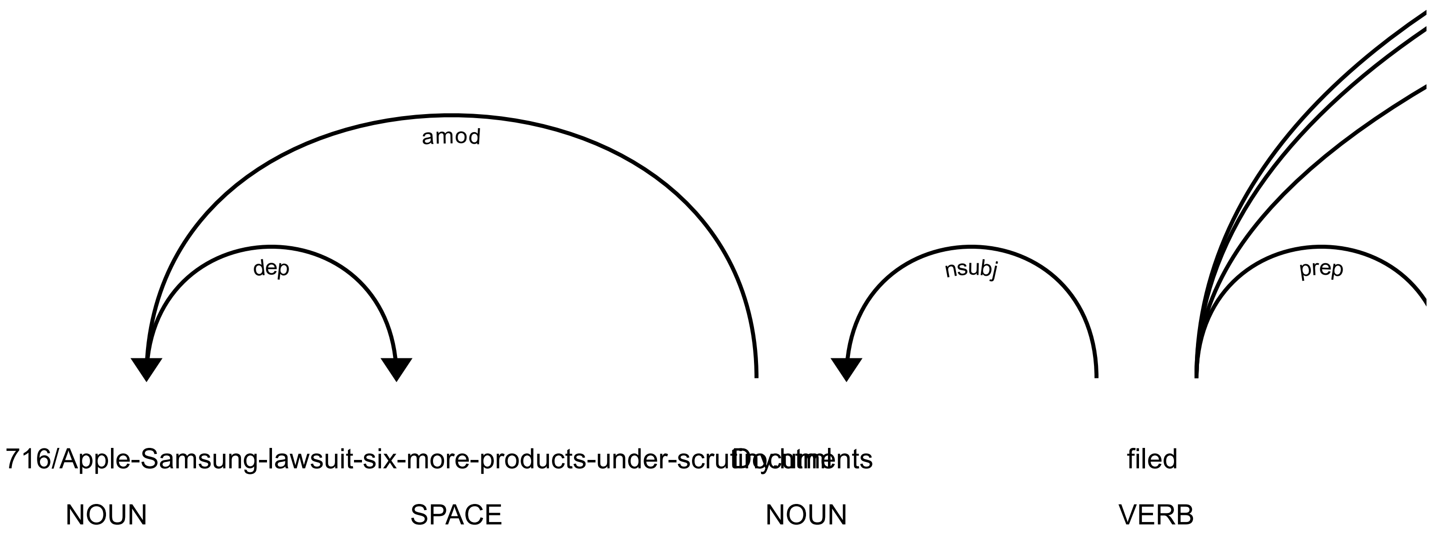
```
https://www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more-produc
ts-under-scrutiny.html TIME
San Jose GPE
California GPE
November 23 DATE
six CARDINAL
Samsung ORG
the "Jelly Bean LAW
Apple ORG
six CARDINAL
```

```
the Galaxy S III ORG
Jelly Bean ORG
8.9 CARDINAL
2 10.1 DATE
Galaxy Rugby Pro ORG
Galaxy S III PERSON
Apple ORG
Apple ORG
August DATE
Samsung ORG
US GPE
Apple ORG
1.05bn MONEY
iPad ORG
Galaxy FAC
Samsung ORG
UK GPE
Samsung ORG
Apple ORG
South Korean NORP
iPad ORG
```

Constituency/dependency parsing

In [27]:

```
spacy.displacy.render(doc, jupyter=True, style='dep')
```


amod
dep
nsubj
prep
716/Apple-Samsung-lawsuit-six-more-products-under-scrutiny.html
Documents
filed
NOUN
SPACE
NOUN
VERB

```
In [28]:
# dep_ provides the syntactic relation, e.g., nsubj
# head provides the head of a Token

for sentence in sents:
    print()
    print(sentence)
    for token in sentence:
        print(token.text, token.dep_, token.head)
```

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

Documents filed to the San Jose federal court in California on November 23 list six Samsun
g products running the "Jelly Bean" and "Ice Cream Sandwich" operating systems, which Appl
e claims infringe its patents.

https://www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more-produc
```

```
ts-under-scrutiny.html amod Documents


 dep https://www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more-p
roducts-under-scrutiny.html
Documents nsubj filed
filed ROOT filed
to prep filed
the det court
San nmod Jose
Jose nmod court
federal amod court
court pobj to
in prep court
California pobj in
on prep filed
November pobj on
23 nummod November
list compound products
six nummod products
Samsung compound products
products dobj filed
running acl products
the det Bean
" punct Bean
Jelly compound Bean
Bean dobj running
" punct Bean
and cc Bean
" punct Sandwich
Ice compound Cream
Cream compound Sandwich
Sandwich nmod systems
" punct Sandwich
operating compound systems
systems conj Bean
, punct systems
which nsubj infringe
Apple compound claims
claims nsubj infringe
infringe relcl systems
its poss patents
patents dobj infringe
. punct filed

 dep .

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 Galaxy Tab 2 10.1, Galaxy Rugby Pro and Galaxy S II
I mini.

The det phones
six nummod phones
phones nsubj are
and cc phones
tablets conj phones
affected acl tablets
are ROOT are
the det III
Galaxy compound III
S compound III
III attr are
, punct are
running advcl are
the det system
new amod system
```

```
Jelly compound Bean
Bean compound system
system dobj running
, punct system
the det tablet
Galaxy compound tablet
Tab nmod tablet
8.9 nummod tablet
Wifi compound tablet
tablet appos system
, punct tablet
the det Tab
Galaxy compound Tab
Tab conj tablet
2 compound 10.1
10.1 nummod Tab
, punct Tab
Galaxy compound Pro
Rugby compound Pro
Pro conj Tab
and cc Pro
Galaxy compound III
S compound III
III conj Pro
mini appos Pro
. punct are

 dep .

Apple stated it had â€œacted quickly and diligently" in order to "determine that these new
ly released products do infringe many of the same claims already asserted by Apple.
Apple nsubj stated
stated ROOT stated
it nsubj â€œacted
had aux â€œacted
â€œacted ccomp stated
quickly advmod â€œacted
and cc quickly
diligently conj quickly
" punct â€œacted
in prep â€œacted
order pobj in
to aux determine
" punct determine
determine acl order
that mark infringe
these det products
newly advmod released
released amod products
products nsubj infringe
do aux infringe
infringe ccomp determine
many dobj infringe
of prep many
the det claims
same amod claims
claims pobj of
already advmod asserted
asserted acl claims
by agent asserted
Apple pobj by
. punct stated

"
In August, Samsung lost a US patent case to Apple and was ordered to pay its rival $1.05bn
(Â£0.66bn) in damages for copying features of the iPad and iPhone in its Galaxy range of d
```

```
evices.
" punct lost

 dep "
In prep lost
August pobj In
, punct lost
Samsung nsubj lost
lost ROOT lost
a det case
US compound case
patent compound case
case dobj lost
to prep lost
Apple pobj to
and cc lost
was auxpass ordered
ordered conj lost
to aux pay
pay xcomp ordered
its poss rival
rival dative pay
$ nmod 1.05bn
1.05bn dobj pay
( punct 1.05bn
Â£0.66bn appos 1.05bn
) punct 1.05bn
in prep pay
damages pobj in
for prep damages
copying pcomp for
features dobj copying
of prep features
the det iPad
iPad pobj of
and cc iPad
iPhone conj iPad
in prep copying
its poss range
Galaxy compound range
range pobj in
of prep range
devices pobj of
. punct lost

Samsung, which is the world's top mobile phone maker, is appealing the ruling.

Samsung nsubj appealing
, punct Samsung
which nsubj is
is relcl Samsung
the det world
world poss maker
's case world
top amod maker
mobile amod phone
phone compound maker
maker attr is
, punct Samsung
is aux appealing
appealing ROOT appealing
the det ruling
ruling dobj appealing
. punct appealing

 dep .
```

```
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 dev
ices.
A det case
similar amod case
case nsubj found
in prep case
the det UK
UK pobj in
found ROOT found
in prep found
Samsung poss favour
's case Samsung
favour pobj in
and cc found
ordered conj found
Apple dobj ordered
to aux publish
publish xcomp ordered
an det apology
apology dobj publish
making acl apology
clear acomp making
that mark copied
the det firm
South amod Korean
Korean amod firm
firm nsubj copied
had aux copied
not neg copied
copied ccomp making
its poss iPad
iPad dobj copied
when advmod designing
designing advcl copied
its poss devices
own amod devices
devices dobj designing
. punct found
```

[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.

Exercise 3a answers:

- For sentence splitting, NLTK demonstrates better performance, effectively handling the text without any noticeable errors. In contrast, spaCy encounters difficulties distinguishing between the end of the third sentence [3] and the beginning of the fourth [4], mistakenly interpreting the closing quotation mark followed by a period as the start of a new sentence. This misinterpretation wrongly positions the newline character ('\n') that should denote the end of the third sentence at the start of the fourth instead. Consequently, in this specific instance, NLTK's straightforward, punctuation-driven approach to identifying sentence boundaries performs better, compared to spaCy's context-aware model.
- The first sentence has been selected for analysis due to its potential to reveal differences in how NLTK and spaCy process URLs. Additionally the unusual semantics and vocabulary in this sentence, 'Samsung running the Jelly Bean and Ice Cream Sandwich Operating Systems', mean that NLTK and spaCy could yield different and interesting results when part of speech tagging.
- After comparing the output in token.tag from spaCy to the part of speech tagging from NLTK for each token, there appear to be some differences, most notably, the URL seems to be split up by NLTK into:

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

    Meanwhile, spaCy treats the URL as follows, indicating that it may have been trained on a wider variety of content, including web-based content:

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

    #### Also noteworthy are the following differences in tagging for the tokens

    NLTK:

  - filed VBN
  - to TO
  - Jelly RB
  - Sandwich NNP
  - operating VBG
  - infringe VB

    spaCy:

  - filed VBD
  - to IN
  - Jelly NNP
  - Sandwich NN
  - operating NN
  - infringe VBP

In [29]:

```
for i,sent in enumerate(sentences_nltk, 1):
    print(i,sent,'\n')
    print()
for i,sent in enumerate(doc.sents, 1):
    print(i,sent)
```

```
1 https://www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more-prod
```

```
ucts-under-scrutiny.html

Documents filed to the San Jose federal court in California on November 23 list six Samsun
g products running the "Jelly Bean" and "Ice Cream Sandwich" operating systems, which Appl
e claims infringe its patents.


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


3 Apple stated it had â€œacted quickly and diligently" in order to "determine that these n
ewly released products do infringe many of the same claims already asserted by Apple."


4 In August, Samsung lost a US patent case to Apple and was ordered to pay its rival $1.05
bn (Â£0.66bn) in damages for copying features of the iPad and iPhone in its Galaxy range o
f devices.


5 Samsung, which is the world's top mobile phone maker, is appealing the ruling.


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


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

Documents filed to the San Jose federal court in California on November 23 list six Samsun
g products running the "Jelly Bean" and "Ice Cream Sandwich" operating systems, which Appl
e claims infringe its patents.

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

3 Apple stated it had â€œacted quickly and diligently" in order to "determine that these n
ewly released products do infringe many of the same claims already asserted by Apple.
4 "
In August, Samsung lost a US patent case to Apple and was ordered to pay its rival $1.05bn
(Â£0.66bn) in damages for copying features of the iPad and iPhone in its Galaxy range of d
evices.
5 Samsung, which is the world's top mobile phone maker, is appealing the ruling.

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

In [30]:

```python
# sentences_nltk

## CHOOSE A SENTENCE
chosen_sentence = word_tokenize(sentences_nltk[0])  # Tokenize the first sentence
tagged_tokens = pos_tag(chosen_sentence)

for token, tag in tagged_tokens:  # Iterate over the list of tagged tokens
    print(token, tag)
```

```
https NN
: :
//www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more-products-und
er-scrutiny.html JJ
```

```
Documents NNS
filed VBN
to TO
the DT
San NNP
Jose NNP
federal JJ
court NN
in IN
California 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
infringe VB
its PRP$
patents NNS
. .
```

In [31]:
```python
# doc

## CHOOSE A SENTENCE
for token in sents[0]:  # the first sentence is selected
    print(token.text, token.tag_)
```

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


 _SP
Documents NNS
filed VBD
to IN
the DT
San NNP
Jose NNP
federal JJ
court NN
in IN
California NNP
on IN
November NNP
23 CD
list NN
```

```
six CD
Samsung NNP
products NNS
running VBG
the DT
" ``
Jelly NNP
Bean NNP
" ''
and CC
" ``
Ice NNP
Cream NNP
Sandwich NN
" ''
operating NN
systems NNS
, ,
which WDT
Apple NNP
claims VBZ
infringe VBP
its PRP$
patents NNS
. .

  _SP
```

[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?

Exercise 3b answer:

NLTK struggles with URLs and specific names, sometimes breaking them down incorrectly or assigning odd categorizations, for example, misclassifying 'Apple' as a (PERSON) instead of an organization. spaCy handles the URL better, but strangely tags the URL as 'TIME'. It occasionally misclassifies names, for example, labeling 'Galaxy S III' as a "PERSON". spaCy appears to perform better in this comparison, as it offers a slightly more robust NER algorithm

[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.

Exercise 3c answer

```
- Constituency parsing breaks down a sentence into its constituent parts, also
known as phrases or syntactic categories. These constituents are represented in a
tree structure, where each node represents a phrase, and leaves represent the words
in the sentence. Dependency parsing, on the other hand, focuses on the
relationships between words in a sentence. It represents these relationships in a
tree structure where each node is a word, and edges are the grammatical
relationships (dependencies) between the words. Each dependency has a direction and
a type that indicates how two words are related, with one word acting as the "head"
```

of the relationship and the other as the "dependent".
- The output for NLTK shows how sentences can be decomposed into nested phrases e.g (NEP Galaxy/NNP S/NNP III/NNP), This example highlights the hierarchical structure of a sentence and identifies the roles played by each phrase within the sentence. The output for spaCY shows how each word in the sentence is connected to others, indicating the type of grammatical relationship that exists between them e.g 'six nummod phones'and 'phones nsubj are', where 'phones' is the head for 'six', but de dependent for 'are'.

In [32]:
```
## CHOOSE A SENTENCE
chosen_sentence = word_tokenize(sentences_nltk[1])  # Tokenize the second sentence
tagged_tokens = pos_tag(chosen_sentence)
constituent_structure = constituent_parser_v2.parse(tagged_tokens)
print(constituent_structure)
```

```
(S
  (NP The/DT)
  six/CD
  phones/NNS
  and/CC
  tablets/NNS
  (VP (V affected/VBN))
  (VP (V are/VBP) (NP the/DT))
  (NEP Galaxy/NNP S/NNP III/NNP)
  ,/,
  (VP (V running/VBG) (NP the/DT new/JJ))
  (NEP Jelly/NNP Bean/NNP)
  (NP system/NN)
  ,/,
  (NP the/DT)
  (NEP Galaxy/NNP Tab/NNP 8.9/CD)
  (NEP Wifi/NNP)
  (NP tablet/NN)
  ,/,
  (NP the/DT)
  (NEP Galaxy/NNP Tab/NNP 2/CD)
  10.1/CD
  ,/,
  (NEP Galaxy/NNP Rugby/NNP Pro/NNP)
  and/CC
  (NEP Galaxy/NNP S/NNP III/NNP)
  (NP mini/NN)
  ./.)
```

In [33]:
```
## CHOOSE A SENTENCE
for token in sents[1]:  # the second sentence is selected
    print(token.text, token.dep_, token.head)
```

```
The det phones
six nummod phones
phones nsubj are
and cc phones
tablets conj phones
affected acl tablets
are ROOT are
the det III
Galaxy compound III
S compound III
III attr are
, punct are
running advcl are
the det system
new amod system
```

```
Jelly compound Bean
Bean compound system
system dobj running
, punct system
the det tablet
Galaxy compound tablet
Tab nmod tablet
8.9 nummod tablet
Wifi compound tablet
tablet appos system
, punct tablet
the det Tab
Galaxy compound Tab
Tab conj tablet
2 compound 10.1
10.1 nummod Tab
, punct Tab
Galaxy compound Pro
Rugby compound Pro
Pro conj Tab
and cc Pro
Galaxy compound III
S compound III
III conj Pro
mini appos Pro
. punct are

 dep .
```

End of this notebook