Lab1-Assignment

Copyright: Vrije Universiteit Amsterdam, Faculty of Humanities, CLTL

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 [3]:
```
from pathlib import Path
```

In [4]:
```
nltk.download('averaged_perceptron_tagger')
nltk.download('maxent_ne_chunker')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-4-b3ef775cd85f> in <module>
----> 1 nltk.download('averaged_perceptron_tagger')
      2 nltk.download('maxent_ne_chunker')

NameError: name 'nltk' is not defined
```

In [5]:
```
cur_dir = Path().resolve() # this should provide you with the folder in which this no
path_to_file = Path.joinpath(cur_dir, 'Lab1-apple-samsung-example.txt')
print(path_to_file)
print('does path exist? ->', Path.exists(path_to_file))
```

```
C:\Users\Marlon\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 [6]:
```
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 [7]:
```
import nltk
from nltk.tokenize import sent_tokenize
from nltk import word_tokenize
```

In [8]:
```
sentences_nltk = sent_tokenize(text)
```

In [9]:
```
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 [10]:
```
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 Bean system, the Galaxy Tab 8.9 Wifi tablet, the Galaxy Tab 2 10.1, 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', 'system', ',', 'the', 'Galaxy', 'Tab', '8.9', 'Wifi', 'tablet', ',', 'the', 'Galaxy', '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 [11]:
```
pos_tags_per_sentence = []
for tokens in tokens_per_sentence:
    print(nltk.pos_tag(tokens))
    pos_tags_per_sentence.append(nltk.pos_tag(tokens))
```

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

In [12]:
```
print(pos_tags_per_sentence)
```

```
[[('https', 'NN'), (':', ':'), ('//www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more-products-under-scrutiny.html', 'JJ'), ('Documents', 'NNS'), ('filed', 'VBN'), ('to', 'TO'), ('the', 'DT'), ('San', 'NNP'), ('Jose', 'NNP'), ('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$'), ('patents', 'NNS'), ('.', '.')], [('The', 'DT'), ('six', 'CD'), ('phones', 'NNS'), ('and', 'CC'), ('tablets', 'NNS'), ('affected', 'VBN'), ('are', 'VBP'), ('the', 'DT'), ('Galaxy', 'NNP'), ('S', 'NNP'), ('III', 'NNP'), (',', ','), ('running', 'VBG'), ('the', 'DT'), ('new', 'JJ'), ('Jelly', 'NNP'), ('Bean', 'NNP'), ('system', 'NN'), (',', ','), ('the', 'DT'), ('Galaxy', 'NNP'), ('Tab', 'NNP'), ('8.9', 'CD'), ('Wifi', 'NNP'), ('tablet', 'NN'), (',', ','), ('the', 'DT'), ('Galaxy', 'NNP'), ('Tab', 'NNP'), ('2', 'CD'), ('10.1', 'CD'), (',', ','), ('Galaxy', 'NNP'), ('Rugby', 'NNP'), ('Pro', 'NNP'), ('and', 'CC'), ('Galaxy', 'NNP'), ('S', 'NNP'), ('III', 'NNP'), ('mini', 'NN'), ('.', '.')], [('Apple', 'NNP'), ('stated', 'VBD'), ('it', 'PRP'), ('had', 'VBD'), ('â€œacted', 'VBN'), ('quickly', 'RB'), ('and', 'CC'), ('diligently', 'RB'), ("''", "''"), ('in', 'IN'), ('order', 'NN'), ('to', 'TO'), ('``', '``'), ('determine', 'VB'), ('that', 'IN'), ('these', 'DT'), ('newly', 'RB'), ('released', 'VBN'), ('products', 'NNS'), ('do', 'VBP'), ('infringe', 'VB'), ('many', 'JJ'), ('of', 'IN'), ('the', 'DT'), ('same', 'JJ'), ('claims', 'NNS'), ('already', 'RB'), ('asserted', 'VBN'), ('by', 'IN'), ('Apple', 'NNP'), ('.', '.'), ("''", "''")], [('In', 'IN'), ('August', 'NNP'), (',', ','), ('Samsung', 'NNP'), ('lost', 'VBD'), ('a', 'DT'), ('US', 'NNP'), ('patent', 'NN'), ('case', 'NN'), ('to', 'TO'), ('Apple', 'NNP'), ('and', 'CC'), ('was', 'VBD'), ('ordered', 'VBN'), ('to', 'TO'), ('pay', 'VB'), ('its', 'PRP$'), ('rival', 'JJ'), ('$', '$'), ('1.05bn', 'CD'), ('(', '('), ('Â£0.66bn', 'NN'), (')', ')'), ('in', 'IN'), ('damages', 'NNS'), ('for', 'IN'), ('copying', 'VBG'), ('features', 'NNS'), ('of', 'IN'), ('the', 'DT'), ('iPad', 'NN'), ('and', 'CC'), ('iPhone', 'NN'), ('in', 'IN'), ('its', 'PRP$'), ('Galaxy', 'NNP'), ('range', 'NN'), ('of', 'IN'), ('devices', 'NNS'), ('.', '.')], [('Samsung', 'NNP'), (',', ','), ('which', 'WDT'), ('is', 'VBZ'), ('the', 'DT'), ('world', 'NN'), ("'s", 'POS'), ('top', 'JJ'), ('mobile', 'NN'), ('phone', 'NN'), ('maker', 'NN'), (',', ','), ('is', 'VBZ'), ('appealing', 'VBG'), ('the', 'DT'), ('ruling', 'NN'), ('.', '.')], [('A', 'DT'), ('similar', 'JJ'), ('case', 'NN'), ('in', 'IN'), ('the', 'DT'), ('UK', 'NNP'), ('found', 'VBD'), ('in', 'IN'), ('Samsung', 'NNP'), ("'s", 'POS'), ('favour', 'NN'), ('and', 'CC'), ('ordered', 'VBD'), ('Apple', 'NNP'), ('to', 'TO'), ('publish', 'VB'), ('an', 'DT'), ('apology', 'NN'), ('making', 'VBG'), ('clear', 'JJ'), ('that', 'IN'), ('the', 'DT'), ('South', 'JJ'), ('Korean', 'JJ'), ('firm', 'NN'), ('had', 'VBD'), ('not', 'RB'), ('copied', 'VBN'), ('its', 'PRP$'), ('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 [13]:
```
from nltk.chunk import ne_chunk
ner_tags_per_sentence = []
for sentence in pos_tags_per_sentence:
    ner_tags_per_sentence.append(nltk.chunk.ne_chunk(sentence))
```

In [14]:
```
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'), 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$'), ('patents', 'NNS'), ('.', '.')]), Tree('S', [('The', 'DT'), ('six', 'CD'), ('phones', 'NNS'), ('and', 'CC'), ('tablets', 'NNS'), ('affected', 'VBN'), ('are', 'VBP'), ('the', 'DT'), Tree('ORGANIZATION', [('Galaxy', 'NNP')]), ('S', 'NNP'), ('III', 'NNP'), (',', ','), ('running', 'VBG'), ('the', 'DT'), ('new', 'JJ'), Tree('PERSON', [('Jelly', 'NNP'), ('Bean', 'NNP')]), ('system', 'NN'), (',', ','), ('the', 'DT'), Tree('ORGANIZATION', [('Galaxy', 'NNP')]), ('Tab', 'NNP'), ('8.9', 'CD'), ('Wifi', 'NNP'), ('tablet', 'NN'), (',', ','), ('the', 'DT'), Tree('ORGANIZATION', [('Galaxy', 'NNP')]), ('Tab', 'NNP'), ('2', 'CD'), ('10.1', 'CD'), (',', ','), Tree('PERSON', [('Galaxy', 'NNP'), ('Rugby', 'NNP'), ('Pro', 'NNP')]), ('and', 'CC'), Tree('PERSON', [('Galaxy', 'NNP'), ('S', 'NNP')]), ('III', 'NNP'), ('mini', 'NN'), ('.', '.')]), Tree('S', [Tree('PERSON', [('Apple', 'NNP')]), ('stated', 'VBD'), ('it', 'PRP'), ('had', 'VBD'), ('â€œacted', 'VBN'), ('quickly', 'RB'), ('and', 'CC'), ('diligently', 'RB'), ("''", "''"), ('in', 'IN'), ('order', 'NN'), ('to', 'TO'), ('``', '``'), ('determine', 'VB'), ('that', 'IN'), ('these', 'DT'), ('newly', 'RB'), ('released', 'VBN'), ('products', 'NNS'), ('do', 'VBP'), ('infringe', 'VB'), ('many', 'JJ'), ('of', 'IN'), ('the', 'DT'), ('same', 'JJ'), ('claims', 'NNS'), ('already', 'RB'), ('asserted', 'VBN'), ('by', 'IN'), Tree('PERSON', [('Apple', 'NNP')]), ('.', '.'), ("''", "''")]), Tree('S', [('In', 'IN'), Tree('GPE', [('August', 'NNP')]), (',', ','), Tree('PERSON', [('Samsung', 'NNP')]), ('lost', 'VBD'), ('a', 'DT'), Tree('GSP', [('US', 'NNP')]), ('patent', 'NN'), ('case', 'NN'), ('to', 'TO'), Tree('GPE', [('Apple', 'NNP')]), ('and', 'CC'), ('was', 'VBD'), ('ordered', 'VBN'), ('to', 'TO'), ('pay', 'VB'), ('its', 'PRP$'), ('rival', 'JJ'), ('$', '$'), ('1.05bn', 'CD'), ('(', '('), ('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$'), Tree('GPE', [('Galaxy', 'NNP')]), ('range', 'NN'), ('of', 'IN'), ('devices', 'NNS'), ('.', '.')]), Tree('S', [Tree('GPE', [('Samsung', 'NNP')]), (',', ','), ('which', 'WDT'), ('is', 'VBZ'), ('the', 'DT'), ('world', 'NN'), ("'s", 'POS'), ('top', 'JJ'), ('mobile', 'NN'), ('phone', 'NN'), ('maker', 'NN'), (',', ','), ('is', 'VBZ'), ('appealing', 'VBG'), ('the', 'DT'), ('ruling', 'NN'), ('.', '.')]), Tree('S', [('A', 'DT'), ('similar', 'JJ'), ('case', 'NN'), ('in', 'IN'), ('the', 'DT'), Tree('ORGANIZATION', [('UK', 'NNP')]), ('found', 'VBD'), ('in', 'IN'), Tree('GPE', [('Samsung', 'NNP')]), ("'s", 'POS'), ('favour', 'NN'), ('and', 'CC'), ('ordered', 'VBD'), Tree('PERSON', [('Apple', 'NNP')]), ('to', 'TO'), ('publish', 'VB'), ('an', 'DT'), ('apology', 'NN'), ('making', 'VBG'), ('clear', 'JJ'), ('that', 'IN'), ('the', 'DT'), Tree('LOCATION', [('South', 'JJ'), ('Korean', 'JJ')]), ('firm', 'NN'), ('had', 'VBD'), ('not', 'RB'), ('copied', 'VBN'), ('its', 'PRP$'), ('iPad', 'NN'), ('when', 'WRB'), ('designing', 'VBG'), ('its', 'PRP$'), ('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 [16]:
```
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 [17]:
```
constituency_output_per_sentence = []
for token in pos_tags_per_sentence:
    constituency_output_per_sentence.append(constituent_parser.parse(token))
```

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

```
[Tree('S', [Tree('NP', [('https', 'NN')]), (':', ':'), Tree('NP', [('//www.telegraph.co.uk/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', 'JJ'), ('court', 'NN')]), 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('VP', [Tree('V', [('operating', 'VBG')])]), ('systems', 'NNS'), (',', ','), ('which', 'WDT'), ('Apple', 'NNP'), Tree('VP', [Tree('V', [('claims', 'VBZ')])]), Tree('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', [Tree('V', [('are', 'VBP')]), Tree('NP', [('the', 'DT')])]), ('Galaxy', 'NNP'), ('S', 'NNP'), ('III', 'NNP'), (',', ','), Tree('VP', [Tree('V', [('running', 'VBG')]), Tree('NP', [('the', 'DT'), ('new', 'JJ')])]), ('Jelly', 'NNP'), ('Bean', 'NNP'), Tree('NP', [('system', 'NN')]), (',', ','), Tree('NP', [('the', 'DT')]), ('Galaxy', 'NNP'), ('Tab', 'NNP'), ('8.9', 'CD'), ('Wifi', 'NNP'), Tree('NP', [('tablet', 'NN')]), (',', ','), Tree('NP', [('the', 'DT')]), ('Galaxy', 'NNP'), ('Tab', 'NNP'), ('2', 'CD'), ('10.1', 'CD'), (',', ','), ('Galaxy', 'NNP'), ('Rugby', 'NNP'), ('Pro', 'NNP'), ('and', 'CC'), ('Galaxy', 'NNP'), ('S', 'NNP'), ('III', 'NNP'), Tree('NP', [('mini', 'NN')]), ('.', '.')]), Tree('S', [('Apple', 'NNP'), Tree('VP', [Tree('V', [('stated', 'VBD')])]), ('it', 'PRP'), Tree('VP', [Tree('V', [('had', 'VBD')])]), 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', [('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', 'VBP')])]), 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', 'VBD')]), Tree('NP', [('a', 'DT')])]), ('US', 'NNP'), Tree('NP', [('patent', 'NN'), ('case', 'NN')]), ('to', 'TO'), ('Apple', 'NNP'), ('and', 'CC'), Tree('VP', [Tree('V', [('was', 'VBD')])]), Tree('VP', [Tree('V', [('ordered', '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', 'NNS'), Tree('PP', [Tree('P', [('of', 'IN')]), Tree('NP', [('the', 'DT'), ('iPad', 'NN')])]), ('and', 'CC'), Tree('NP', [('iPhone', 'NN')]), Tree('P', [('in', 'IN')]), ('its', 'PRP$'), ('Galaxy', 'NNP'), Tree('NP', [('range', 'NN')]), Tree('P', [('of', 'IN')]), ('devices', 'NNS'), ('.', '.')]), Tree('S', [('Samsung', 'NNP'), (',', ','), ('which', 'WDT'), Tree('VP', [Tree('V', [('is', 'VBZ')]), Tree('NP', [('the', 'DT'), ('world', 'NN')])]), ("'s", 'POS'), Tree('NP', [('top', 'JJ'), ('mobile', 'NN'), ('phone', 'NN'), ('maker', 'NN')]), (',', ','), Tree('VP', [Tree('V', [('is', 'VBZ')])]), Tree('VP', [Tree('V', [('appealing', 'VBG')]), Tree('NP', [('the', 'DT'), ('ruling', 'NN')])]), ('.', '.')]), Tree('S', [Tree('NP', [('A', 'DT'), ('similar', 'JJ'), ('case', 'NN')]), Tree('PP', [Tree('P', [('in', 'IN')]), Tree('NP', [('the', 'DT')])]), ('UK', 'NNP'), Tree('VP', [Tree('V', [('found', 'VBD')])]), Tree('P', [('in', 'IN')]), ('Samsung', 'NNP'), ("'s", 'POS'), Tree('NP', [('favour', 'NN')]), ('and', 'CC'), Tree('VP', [Tree('V', [('ordered', 'VBD')])]), ('Apple', 'NNP'), ('to', 'TO'), Tree('VP', [Tree('V', [('publish', 'VB')]), Tree('NP', [('an', 'DT'), ('apology', 'NN')])]), Tree('VP', [Tree('V', [('making', 'VBG')]), Tree('NP', [('clear', 'JJ')])]), Tree('PP', [Tree('P', [('that', 'IN')]), Tree('NP', [('the', 'DT'), ('South', 'JJ'), ('Korean', 'JJ'), ('firm', 'NN')])]), Tree('VP', [Tree('V', [('had', 'VBD')])]), ('not', 'RB'), Tree('VP', [Tree('V', [('copied', 'VBN')])]), ('its', 'PRP$'), Tree('NP', [('iPad', 'NN')]), ('when', 'WRB'), Tree('VP', [Tree('V', [('designing', 'VBG')])]), ('its', '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 [26]:
```
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>*}              # ???''')
```

In [27]:
```
constituency_v2_output_per_sentence = []
for token in pos_tags_per_sentence:
    constituency_v2_output_per_sentence.append(constituent_parser.parse(token))
```

In [28]:
```
print(constituency_v2_output_per_sentence)
```

```
[Tree('S', [Tree('NP', [('https', 'NN')]), (':', ':'), Tree('NP', [('//www.telegraph.co.uk/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', 'JJ'), ('court', 'NN')]), 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('VP', [Tree('V', [('operating', 'VBG')])]), ('systems', 'NNS'), (',', ','), ('which', 'WDT'), ('Apple', 'NNP'), Tree('VP', [Tree('V', [('claims', 'VBZ')])]), Tree('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', [Tree('V', [('are', 'VBP')]), Tree('NP', [('the', 'DT')])]), ('Galaxy', 'NNP'), ('S', 'NNP'), ('III', 'NNP'), (',', ','), Tree('VP', [Tree('V', [('running', 'VBG')]), Tree('NP', [('the', 'DT'), ('new', 'JJ')])]), ('Jelly', 'NNP'), ('Bean', 'NNP'), Tree('NP', [('system', 'NN')]), (',', ','), Tree('NP', [('the', 'DT')]), ('Galaxy', 'NNP'), ('Tab', 'NNP'), ('8.9', 'CD'), ('Wifi', 'NNP'), Tree('NP', [('tablet', 'NN')]), (',', ','), Tree('NP', [('the', 'DT')]), ('Galaxy', 'NNP'), ('Tab', 'NNP'), ('2', 'CD'), ('10.1', 'CD'), (',', ','), ('Galaxy', 'NNP'), ('Rugby', 'NNP'), ('Pro', 'NNP'), ('and', 'CC'), ('Galaxy', 'NNP'), ('S', 'NNP'), ('III', 'NNP'), Tree('NP', [('mini', 'NN')]), ('.', '.')]), Tree('S', [('Apple', 'NNP'), Tree('VP', [Tree('V', [('stated', 'VBD')])]), ('it', 'PRP'), Tree('VP', [Tree('V', [('had', 'VBD')])]), 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', [('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', 'VBP')])]), 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', 'VBD')]), Tree('NP', [('a', 'DT')])]), ('US', 'NNP'), Tree('NP', [('patent', 'NN'), ('case', 'NN')]), ('to', 'TO'), ('Apple', 'NNP'), ('and', 'CC'), Tree('VP', [Tree('V', [('was', 'VBD')])]), Tree('VP', [Tree('V', [('ordered', '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', 'NNS'), Tree('PP', [Tree('P', [('of', 'IN')]), Tree('NP', [('the', 'DT'), ('iPad', 'NN')])]), ('and', 'CC'), Tree('NP', [('iPhone', 'NN')]), Tree('P', [('in', 'IN')]), ('its', 'PRP$'), ('Galaxy', 'NNP'), Tree('NP', [('range', 'NN')]), Tree('P', [('of', 'IN')]), ('devices', 'NNS'), ('.', '.')]), Tree('S', [('Samsung', 'NNP'), (',', ','), ('which', 'WDT'), Tree('VP', [Tree('V', [('is', 'VBZ')]), Tree('NP', [('the', 'DT'), ('world', 'NN')])]), ("'s", 'POS'), Tree('NP', [('top', 'JJ'), ('mobile', 'NN'), ('phone', 'NN'), ('maker', 'NN')]), (',', ','), Tree('VP', [Tree('V', [('is', 'VBZ')])]), Tree('VP', [Tree('V', [('appealing', 'VBG')]), Tree('NP', [('the', 'DT'), ('ruling', 'NN')])]), ('.', '.')]), Tree('S', [Tree('NP', [('A', 'DT'), ('similar', 'JJ'), ('case', 'NN')]), Tree('PP', [Tree('P', [('in', 'IN')]), Tree('NP', [('the', 'DT')])]), ('UK', 'NNP'), Tree('VP', [Tree('V', [('found', 'VBD')])]), Tree('P', [('in', 'IN')]), ('Samsung', 'NNP'), ("'s", 'POS'), Tree('NP', [('favour', 'NN')]), ('and', 'CC'), Tree('VP', [Tree('V', [('ordered', 'VBD')])]), ('Apple', 'NNP'), ('to', 'TO'), Tree('VP', [Tree('V', [('publish', 'VB')]), Tree('NP', [('an', 'DT'), ('apology', 'NN')])]), Tree('VP', [Tree('V', [('making', 'VBG')]), Tree('NP', [('clear', 'JJ')])]), Tree('PP', [Tree('P', [('that', 'IN')]), Tree('NP', [('the', 'DT'), ('South', 'JJ'), ('Korean', 'JJ'), ('firm', 'NN')])]), Tree('VP', [Tree('V', [('had', 'VBD')])]), ('not', 'RB'), Tree('VP', [Tree('V', [('copied', 'VBN')])]), ('its', 'PRP$'), Tree('NP', [('iPad', 'NN')]), ('when', 'WRB'), Tree('VP', [Tree('V', [('designing', 'VBG')])]), ('its', '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 [21]:
```
import spacy
nlp = spacy.load('en_core_web_sm')
```

In [22]:
```
doc = nlp(text) # insert code here

pos_tags_per_sentence_spacy = []
ner_tags_per_sentence_spacy = []
constituency_output_per_sentence_spacy = []

for sentence in doc.sents:
    POS = [(token.text,token.tag_) for token in sentence]
    NER = [ (ent.text,ent.label_) for ent in sentence.ents]
    constituency = [(token.text, token.dep_, token.head) for token in sentence]

    pos_tags_per_sentence_spacy.append(POS)
    ner_tags_per_sentence_spacy.append(NER)
    constituency_output_per_sentence_spacy.append(constituency)
```

In [90]:
```
```

In [23]:
```
print(pos_tags_per_sentence_spacy)
```

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

In [24]:
```
print(ner_tags_per_sentence_spacy)
```

```
[[('San Jose', 'GPE'), ('California', 'GPE'), ('November 23', 'DATE'), ('six', 'CARDINAL'), ('Samsung', 'ORG'), ('Jelly Bean', 'WORK_OF_ART'), ('Apple', 'ORG')], [], [('six', 'CARDINAL'), ('the Galaxy S III', 'GPE'), ('Jelly Bean', 'ORG'), ('the Galaxy Tab 2 10.1', 'ORG')], [('Apple', 'ORG')], [('August', 'DATE'), ('Samsung', 'ORG'), ('US', 'GPE'), ('Apple', 'ORG'), ('1.05bn', 'MONEY'), ('iPad', 'ORG'), ('iPhone', 'ORG')], [('Samsung', 'ORG')], [('UK', 'GPE'), ('Samsung', 'ORG'), ('Apple', 'ORG'), ('South Korean', 'NORP'), ('iPad', 'ORG')]]
```

In [97]:
```
print(constituency_output_per_sentence_spacy)
```

```
[[('https://www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more-products-under-scrutiny.html', 'compound', 

), ('\n\n', 'dep', 

), ('Documents', 'appos', 

), ('filed', 'acl', Documents), ('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', 'appos', 

), ('six', 'nummod', products), ('Samsung', 'compound', products), ('products', 'appos', list), ('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', 'conj', Bean), ('"', 'punct', Sandwich), ('operating', 'compound', systems), ('systems', 'appos', list), (',', 'punct', systems), ('which', 'nsubj', infringe), ('Apple', 'compound', claims), ('claims', 'nsubj', infringe), ('infringe', 'relcl', systems), ('its', 'poss', patents), ('patents', 'dobj', infringe), ('.', 'punct', 

)], [('\n', 'dep', 
)], [('The', 'det', phones), ('six', 'nummod', phones), ('phones', 'nsubj', are), ('and', 'cc', phones), ('tablets', 'conj', phones), ('affected', 'acl', phones), ('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', Tab), ('Galaxy', 'compound', Tab), ('Tab', 'appos', system), ('8.9', 'nummod', tablet), ('Wifi', 'compound', tablet), ('tablet', 'appos', Tab), (',', 'punct', Tab), ('the', 'det', Tab), ('Galaxy', 'compound', Tab), ('Tab', 'appos', Tab), ('2', 'nummod', Tab), ('10.1', 'nummod', Tab), (',', 'punct', system), ('Galaxy', 'compound', Rugby), ('Rugby', 'appos', system), ('Pro', 'appos', Rugby), ('and', 'cc', Pro), ('Galaxy', 'conj', Pro), ('S', 'conj', Pro), ('III', 'nummod', mini), ('mini', 'attr', are), ('.', 'punct', are)], [('\n', 'dep', stated), ('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), ('"', 'punct', stated)], [('\n', 'dep', lost), ('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', 1.05bn), ('rival', 'amod', 1.05bn), ('$', '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', '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)], [('\n', 'dep', case), ('A', 'det', case), ('similar', 'amod', case), ('case', 'ROOT', case), ('in', 'prep', case), ('the', 'det', UK), ('UK', 'pobj', in), ('found', 'acl', case), ('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', case)]]
```

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. Print for each token the output from NLTK and spaCy next to each other (align possible tokenization differences). Are there any differences? This is not a trick question; it is possible that there are no differences.

In [48]:
```
#Comparing sentence splitting for NLTK and for spaCy.
print(pos_tags_per_sentence,'\n')
print(pos_tags_per_sentence_spacy)
```

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

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

In [41]:
```
#We can see that there are indeed some differences for nltk and spacy.
#For example, in one of the first sentences the word 'to' is labeled as 'TO' with NLT
#we can also observe that some of the splitting is different.
```

In [54]:
```
print(pos_tags_per_sentence[0], '\n')
print(pos_tags_per_sentence_spacy[0])

#We choose the sentence below because we are interested to see how the website url is
#We think that NLTK and spacy might split the url in a different way.
#In addition, we think that some of the words will be labeled differently.
```

```
[('https', 'NN'), (':', ':'), ('//www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more-products-under-scrutiny.html', 'JJ'), ('Documents', 'NNS'), ('filed', 'VBN'), ('to', 'TO'), ('the', 'DT'), ('San', 'NNP'), ('Jose', 'NNP'), ('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'), ('.', '.')] 

[('https://www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more-products-under-scrutiny.html', 'NNP'), ('\n\n', '_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', 'NNP'), ('"', "''"), ('operating', 'NN'), ('systems', 'NNS'), (',', ','), ('which', 'WDT'), ('Apple', 'NNP'), ('claims', 'NNS'), ('infringe', 'VBP'), ('its', 'PRP$'), ('patents', 'NNS'), ('.', '.')]
```

In [53]:
```
for item_a, item_b in zip(pos_tags_per_sentence[0], pos_tags_per_sentence_spacy[0]):
    print(item_a, item_b)
```

```
('https', 'NN') ('https://www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more-products-under-scrutiny.html', 'NNP')
(':', ':') ('\n\n', '_SP')
('//www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more-products-under-scrutiny.html', 'JJ') ('Documents', 'NNS')
('Documents', 'NNS') ('filed', 'VBD')
('filed', 'VBN') ('to', 'IN')
('to', 'TO') ('the', 'DT')
('the', 'DT') ('San', 'NNP')
('San', 'NNP') ('Jose', 'NNP')
('Jose', 'NNP') ('federal', 'JJ')
('federal', 'JJ') ('court', 'NN')
('court', 'NN') ('in', 'IN')
('in', 'IN') ('California', 'NNP')
('California', 'NNP') ('on', 'IN')
('on', 'IN') ('November', 'NNP')
('November', 'NNP') ('23', 'CD')
('23', 'CD') ('list', 'NN')
('list', 'NN') ('six', 'CD')
('six', 'CD') ('Samsung', 'NNP')
('Samsung', 'NNP') ('products', 'NNS')
('products', 'NNS') ('running', 'VBG')
('running', 'VBG') ('the', 'DT')
('the', 'DT') ('"', '``')
('``', '``') ('Jelly', 'NNP')
('Jelly', 'RB') ('Bean', 'NNP')
('Bean', 'NNP') ('"', "''")
("''", "''") ('and', 'CC')
('and', 'CC') ('"', '``')
('``', '``') ('Ice', 'NNP')
('Ice', 'NNP') ('Cream', 'NNP')
('Cream', 'NNP') ('Sandwich', 'NNP')
('Sandwich', 'NNP') ('"', "''")
("''", "''") ('operating', 'NN')
('operating', 'VBG') ('systems', 'NNS')
('systems', 'NNS') (',', ',')
(',', ',') ('which', 'WDT')
('which', 'WDT') ('Apple', 'NNP')
('Apple', 'NNP') ('claims', 'NNS')
('claims', 'VBZ') ('infringe', 'VBP')
('infringe', 'VB') ('its', 'PRP$')
('its', 'PRP$') ('patents', 'NNS')
('patents', 'NNS') ('.', '.')
```

In [ ]:
```
# In the sentences above we can observe some differences between the behavior in NLTK
#First, we can see that not every words is split the same way. The URL at the beginni
#while spacy retains the whole URL. Secondly,
#we can see that the words 'jelly', 'operating' and the word 'claims' are labeled dif
```

[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 [64]:
```
for i in ner_tags_per_sentence:
    print(i)

for i in ner_tags_per_sentence_spacy:
    print(i)
```

```
(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
  (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
  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)
  ./.
  ''/'')
(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
    ./.)
  (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
  ./.)
[('San Jose', 'GPE'), ('California', 'GPE'), ('November 23', 'DATE'), ('six', 'CARDINA
L'), ('Samsung', 'ORG'), ('Jelly Bean', 'WORK_OF_ART'), ('Apple', 'ORG')]
[]
[('six', 'CARDINAL'), ('the Galaxy S III', 'GPE'), ('Jelly Bean', 'ORG'), ('the Galaxy
Tab 2 10.1', 'ORG')]
[('Apple', 'ORG')]
[('August', 'DATE'), ('Samsung', 'ORG'), ('US', 'GPE'), ('Apple', 'ORG'), ('1.05bn',
'MONEY'), ('iPad', 'ORG'), ('iPhone', 'ORG')]
[('Samsung', 'ORG')]
[('UK', 'GPE'), ('Samsung', 'ORG'), ('Apple', 'ORG'), ('South Korean', 'NORP'), ('iPa
d', 'ORG')]
```

In [65]:

```
# SpaCy finds the relations betweeen words instead of seeing all the words as separate
# a word like 'jelly bean' is split into 'jelly' and 'bean' in NLTK and spacy classif
# Another example is the word 'the Galaxy S III' which is seen as an individual word
# Eventhough spacy incorrectly labeled 'the Galaxy S III' as 'GPE' meaning Geopolitica
# important distinction that some words that should be seen as one entity are correct
# Therefore, we think that spaCy perfroms better on Named Entity Recognition.
```

[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.

In [68]:

```
print(constituency_output_per_sentence[0], '\n')
print(constituency_output_per_sentence_spacy[0])
```

```
(S
  (NP https/NN)
  :/:
  (NP
    //www.telegraph.co.uk/technology/apple/9702716/Apple-Samsung-lawsuit-six-more-prod
ucts-under-scrutiny.html/JJ)
  Documents/NNS
  (VP (V filed/VBN))
  to/TO
  (NP the/DT)
  San/NNP
  Jose/NNP
  (NP federal/JJ court/NN)
  (P in/IN)
  California/NNP
  (P on/IN)
  November/NNP
  23/CD
  (NP list/NN)
  six/CD
  Samsung/NNP
  products/NNS
  (VP (V running/VBG) (NP the/DT))
  ``/``
  Jelly/RB
  Bean/NNP
  ''/''
  and/CC
  ``/``
  Ice/NNP
  Cream/NNP
  Sandwich/NNP
  ''/''
```

```
  (VP (V operating/VBG))
  systems/NNS
  ,/,
  which/WDT
  Apple/NNP
  (VP (V claims/VBZ))
  (VP (V infringe/VB))
  its/PRP$
  patents/NNS
  ./.)

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

), ('\n\n', 'dep',

), ('Documents', 'appos',

), ('filed', 'acl', Documents), ('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), ('Novem
ber', 'pobj', on), ('23', 'nummod', November), ('list', 'appos',

), ('six', 'nummod', products), ('Samsung', 'compound', products), ('products', 'appo
s', list), ('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', 'compoun
d', Sandwich), ('Sandwich', 'conj', Bean), ('"', 'punct', Sandwich), ('operating', 'co
mpound', systems), ('systems', 'appos', list), (',', 'punct', systems), ('which', 'nsu
bj', infringe), ('Apple', 'compound', claims), ('claims', 'nsubj', infringe), ('infrin
ge', 'relcl', systems), ('its', 'poss', patents), ('patents', 'dobj', infringe), ('.',
'punct',

)]
```

In [ ]:

```
# Dependency parsing will show the relationships between words and their constitutes,
# while constituency parsing will showcase the the whole sentence structure and the re
# NLTK is a library that processes strings. It takes strings as input and it returns s
# In contrast, SpaCy uses an object-oriented approach. When a text is parsed withSpaCy
# it returns document object whose words and sentences are objects.

#Looking at the above output, we can observe differences between NLTK and spaCy.
# For example, the word 'cream' is labeled NNP in NLTK while in spaCy it is labeled as
# some words seem to be labeled the same, but overall there are many differences betwe
```

End of this notebook