NLTK

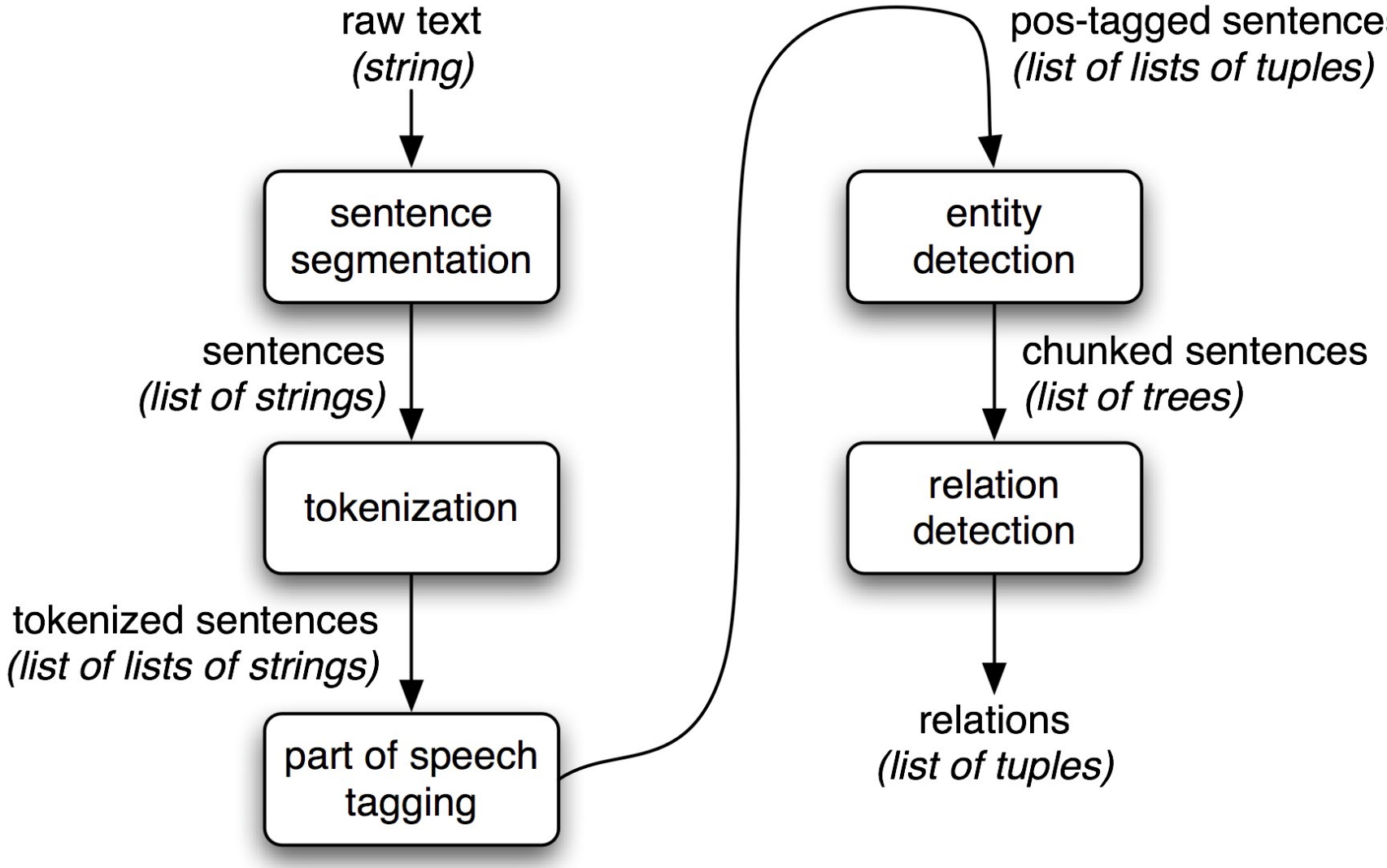
Natural Language Toolkit

http://www.nltk.org/

NLTK

Es una herramienta que permite realizar tareas relacionadas con el procesamiento de lenguaje natural.

NLTK – Extraer Información



Segmentación en Sentencias

sentences = nltk.sent_tokenize(document)

```
['We saw the yellow dog.']
```

Tokenización

sentences = [nltk.word_tokenize(sent) for sent in sentences]

```
[['We', 'saw', 'the', 'yellow', 'dog', '.']]
```

Etiquetado (Part-of-Speech Tagging)

sentences_tagged = [nltk.pos_tag(sent) for sent in sentences]

```
[[('We', 'PRP'), ('saw', 'VBD'), ('the', 'DT'),
 ('yellow', 'JJ'), ('dog', 'NN'), ('.', '.')]]
```

Etiquetado (Part-of-Speech Tagging)

part-of-speech tags used in the Penn Treebank Project

Number	Tag	Description
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential *there*
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRPS	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	*to*
26.	UH	Interjection

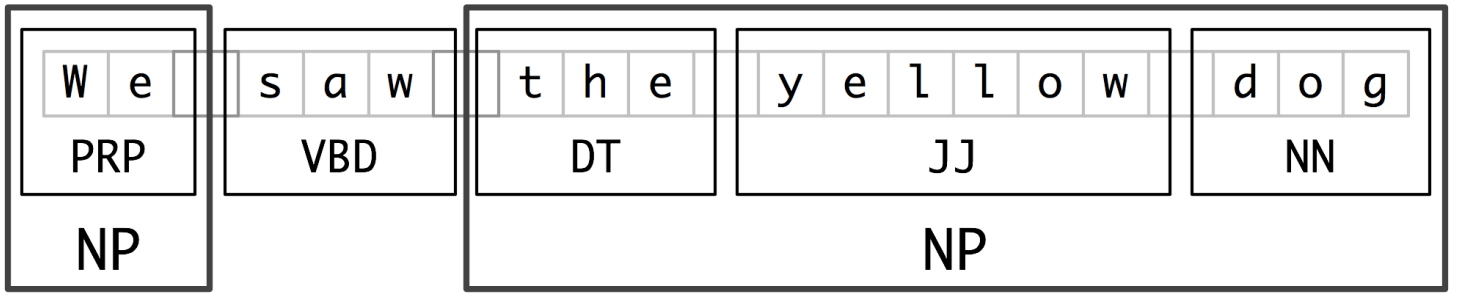
Detección de Entidades (Chunking)

```
grammar = """NP: {<DT>?<JJ>*<NN|PRP>}"""
...
cp = nltk.RegexpParser(grammar)
    for sentece in sentences_tagged:
        entities = cp.parse(sentece)
        for subtree in entities.subtrees():
            if subtree.label() == 'NP':
                print subtree
```

```
(NP We/PRP)
(NP the/DT yellow/JJ dog/NN)
```

Detección de Entidades (Chunking)

document = """We saw the yellow dog."""



Relación de Extracciones

```
>>> IN = re.compile(r'.*\bin\b(?!\b.+ing)')
>>> for doc in nltk.corpus.ieer.parsed_docs('NYT_19980315'):
...     for rel in nltk.sem.extract_rels('ORG', 'LOC', doc,
...                                      corpus='ieer', pattern =
IN):
...         print nltk.sem.show_raw_rtuple(rel)
```

[ORG: 'WHYY'] 'in' [LOC: 'Philadelphia']
[ORG: 'McGlashan & Sarrail'] 'firm in' [LOC: 'San Mateo']
[ORG: 'Freedom Forum'] 'in' [LOC: 'Arlington']
[ORG: 'Brookings Institution'] ', the research group in' [LOC: 'Washington']
[ORG: 'Idealab'] ', a self-described business incubator based in' [LOC: 'Los Angeles']
[ORG: 'Open Text'] ', based in' [LOC: 'Waterloo']
[ORG: 'WGBH'] 'in' [LOC: 'Boston']
[ORG: 'Bastille Opera'] 'in' [LOC: 'Paris']
[ORG: 'Omnicom'] 'in' [LOC: 'New York']
[ORG: 'DDB Needham'] 'in' [LOC: 'New York']
[ORG: 'Kaplan Thaler Group'] 'in' [LOC: 'New York']
[ORG: 'BBDO South'] 'in' [LOC: 'Atlanta']
[ORG: 'Georgia-Pacific'] 'in' [LOC: 'Atlanta']

Bibliografía

http://www.nltk.org/book_1ed/

Gracias