Laboratorio 2

November 19, 2022

0.1 Laboratorio 2: Parte del Discurso (Part of Speech) (POS)

0.1.1 1. ¿De qué se trata?

Este lab se enfoca en aplicar el tagging (marcado o etiquetado, en español) Part of Speech (POS) y de explorar los métodos que ofrece NLTK para asingar POS automáticamente. Para más prácticas e información sobre este tópico, puede visitar: http://www.nltk.org/book/ch05.html

0.1.2 2. NLTK enviroment

El primer paso es importar NLTK y algún corpora como en los otros Labs:

[1]:
```
import nltk
```

[2]:
```
from nltk.book import *
```

```
*** Introductory Examples for the NLTK Book ***
Loading text1, …, text9 and sent1, …, sent9
Type the name of the text or sentence to view it.
Type: 'texts()' or 'sents()' to list the materials.
text1: Moby Dick by Herman Melville 1851
text2: Sense and Sensibility by Jane Austen 1811
text3: The Book of Genesis
text4: Inaugural Address Corpus
text5: Chat Corpus
text6: Monty Python and the Holy Grail
text7: Wall Street Journal
text8: Personals Corpus
text9: The Man Who Was Thursday by G . K . Chesterton 1908
```

0.1.3 3. Part of speech tagsets

Los POS tagger asignan etiquetas a palabras. Los tags son tomados de un tagset. Vimos dos tagsets utilizados en la clase. Te acordas sus nombres?

Penn Treebank y Brown

[3]:
```
nltk.help.upenn_tagset(".*")
```

$: dollar
    $ -$ --$ A$ C$ HK$ M$ NZ$ S$ U.S.$ US$
'': closing quotation mark
    ' ''
(: opening parenthesis
    ([{
): closing parenthesis
)] }
,: comma
    ,
--: dash
    --
.: sentence terminator
    . ! ?
:: colon or ellipsis
    : ; …
CC: conjunction, coordinating
    & 'n and both but either et for less minus neither nor or plus so
    therefore times v. versus vs. whether yet
CD: numeral, cardinal
    mid-1890 nine-thirty forty-two one-tenth ten million 0.5 one forty-
    seven 1987 twenty '79 zero two 78-degrees eighty-four IX '60s .025
    fifteen 271,124 dozen quintillion DM2,000 …
DT: determiner
    all an another any both del each either every half la many much nary
    neither no some such that the them these this those
EX: existential there
    there
FW: foreign word
    gemeinschaft hund ich jeux habeas Haementeria Herr K'ang-si vous
    lutihaw alai je jour objets salutaris fille quibusdam pas trop Monte
    terram fiche oui corporis …
IN: preposition or conjunction, subordinating
    astride among uppon whether out inside pro despite on by throughout
    below within for towards near behind atop around if like until below
    next into if beside …
JJ: adjective or numeral, ordinal
    third ill-mannered pre-war regrettable oiled calamitous first separable
    ectoplasmic battery-powered participatory fourth still-to-be-named
    multilingual multi-disciplinary …
JJR: adjective, comparative
    bleaker braver breezier briefer brighter brisker broader bumper busier
    calmer cheaper choosier cleaner clearer closer colder commoner costlier
    cozier creamier crunchier cuter …
JJS: adjective, superlative
    calmest cheapest choicest classiest cleanest clearest closest commonest
    corniest costliest crassest creepiest crudest cutest darkest deadliest
    dearest deepest densest dinkiest …

LS: list item marker
A A. B B. C C. D E F First G H I J K One SP-44001 SP-44002 SP-44005 SP-44007 Second Third Three Two * a b c d first five four one six three two

MD: modal auxiliary
can cannot could couldn't dare may might must need ought shall should shouldn't will would

NN: noun, common, singular or mass
common-carrier cabbage knuckle-duster Casino afghan shed thermostat investment slide humour falloff slick wind hyena override subhumanity machinist …

NNP: noun, proper, singular
Motown Venneboerger Czestochwa Ranzer Conchita Trumplane Christos Oceanside Escobar Kreisler Sawyer Cougar Yvette Ervin ODI Darryl CTCA Shannon A.K.C. Meltex Liverpool …

NNPS: noun, proper, plural
Americans Americas Amharas Amityvilles Amusements Anarcho-Syndicalists Andalusians Andes Andruses Angels Animals Anthony Antilles Antiques Apache Apaches Apocrypha …

NNS: noun, common, plural
undergraduates scotches bric-a-brac products bodyguards facets coasts divestitures storehouses designs clubs fragrances averages subjectivists apprehensions muses factory-jobs …

PDT: pre-determiner
all both half many quite such sure this

POS: genitive marker
' 's

PRP: pronoun, personal
hers herself him himself hisself it itself me myself one oneself ours ourselves ownself self she thee theirs them themselves they thou thy us

PRP$: pronoun, possessive
her his mine my our ours their thy your

RB: adverb
occasionally unabatingly maddeningly adventurously professedly stirringly prominently technologically magisterially predominately swiftly fiscally pitilessly …

RBR: adverb, comparative
further gloomier grander graver greater grimmer harder harsher healthier heavier higher however larger later leaner lengthier less-perfectly lesser lonelier longer louder lower more …

RBS: adverb, superlative
best biggest bluntest earliest farthest first furthest hardest heartiest highest largest least less most nearest second tightest worst

RP: particle
aboard about across along apart around aside at away back before behind by crop down ever fast for forth from go high i.e. in into just later low more off on open out over per pie raising start teeth that through under unto up up-pp upon whole with you

```
SYM: symbol
    % & ' '' ''. ) ). * + ,. < = > @ A[fj] U.S U.S.S.R * ** ***
TO: "to" as preposition or infinitive marker
    to
UH: interjection
    Goodbye Goody Gosh Wow Jeepers Jee-sus Hubba Hey Kee-reist Oops amen
    huh howdy uh dammit whammo shucks heck anyways whodunnit honey golly
    man baby diddle hush sonuvabitch …
VB: verb, base form
    ask assemble assess assign assume atone attention avoid bake balkanize
    bank begin behold believe bend benefit bevel beware bless boil bomb
    boost brace break bring broil brush build …
VBD: verb, past tense
    dipped pleaded swiped regummed soaked tidied convened halted registered
    cushioned exacted snubbed strode aimed adopted belied figgered
    speculated wore appreciated contemplated …
VBG: verb, present participle or gerund
    telegraphing stirring focusing angering judging stalling lactating
    hankerin' alleging veering capping approaching traveling besieging
    encrypting interrupting erasing wincing …
VBN: verb, past participle
    multihulled dilapidated aerosolized chaired languished panelized used
    experimented flourished imitated reunifed factored condensed sheared
    unsettled primed dubbed desired …
VBP: verb, present tense, not 3rd person singular
    predominate wrap resort sue twist spill cure lengthen brush terminate
    appear tend stray glisten obtain comprise detest tease attract
    emphasize mold postpone sever return wag …
VBZ: verb, present tense, 3rd person singular
    bases reconstructs marks mixes displeases seals carps weaves snatches
    slumps stretches authorizes smolders pictures emerges stockpiles
    seduces fizzes uses bolsters slaps speaks pleads …
WDT: WH-determiner
    that what whatever which whichever
WP: WH-pronoun
    that what whatever whatsoever which who whom whosoever
WP$: WH-pronoun, possessive
    whose
WRB: Wh-adverb
    how however whence whenever where whereby whereever wherein whereof why
``: opening quotation mark
    ` ``
```

Esto desplegará el Penn Treebank tagset, el cual contiene 45 tags! Vas a ver los tags seguidos de una lista de palabras representitivas para su categoría.

Podes usar expresiones regulares para mostrar un subset de los tags. Por ejemplo, reemplazando .* en el comando de arriba con NN*, podes obtener filtrando solo los tags de Nouns (sustantivos). Usando VB*, podes obtener todas las categorías de verbos.

¿Cuántas categorías de verbos y de sustantivos hay? Recordá que tienen significados distintos. Escribí todas las categorías de tipo Noun (sustantivo) y una breve frase describiendo qué es.

En el penn treebank tagset hay 4 categorías de sustantivos y 6 categorías de adjetivos:

Sustantivos:

-NN (Noun, singular or mass): Son aquellos con la propiedad de que cualquier cantidad de lo que

-NNP(Noun, proper, singular): Son aquellos que representan el nombre o título específico de una

-NNPS(Noun, proper, plural): Representan el nombre o título de varias personas, lugares, cosas

-NNS (Noun, plural): Son aquellos que refieren a más de una persona, lugar, cosa o idea.

Verbos:

-VB (verb, base form): Es la forma no personal más simple de un verbo. En inglés es similar al

-VBD(verb, past tense): Verbos en tiempo pasado

-VBG(verb, present participle or gerund): Es una forma verbal no finita que tiene algunas de la

-VBN(verb, past participle): Son verbos que refieren a acciones que comenzaron y terminaron en

-VBP(verb, present tense, not 3rd person singular): Verbos en presente que no están en tercera

-VBZ(verb, present tense, 3rd person singular): Vergos en tercera persona del presente

Repetí este ejercicio con el tagset de "Brown" que tiene 87 tags.

Sustantivos

-NN: noun, singular, common

-NNS: noun, singular, common, genitive

-NN+BEZ: noun, singular, common + verb 'to be', present tense, 3rd person singular

-NN+HVD: noun, singular, common + verb 'to have', past tense

-NN+HVZ: noun, singular, common + verb 'to have', present tense, 3rd person singular

-NN+IN: noun, singular, common + preposition

-NN+MD: noun, singular, common + modal auxillary

-NN+NN: noun, singular, common, hyphenated pair

-NNS: noun, plural, common

-NNSS: noun, plural, common, genitive

-NNS+MD: noun, plural, common + modal auxillary

-NP: noun, singular, proper

-NPS: noun, singular, proper, genitive

-NP+BEZ: noun, singular, proper + verb 'to be', present tense, 3rd person singular

-NP+HVZ: noun, singular, proper + verb 'to have', present tense, 3rd person singular

-NP+MD: noun, singular, proper + modal auxillary

-NPS: noun, plural, proper

-NPSS: noun, plural, proper, genitive

-NR: noun, singular, adverbial

-NRS: noun, singular, adverbial, genitive

-NR+MD: noun, singular, adverbial + modal auxillary

-NRS: noun, plural, adverbial

Verbos

-VB: verb, base: uninflected present, imperative or infinitive

-VB+AT: verb, base: uninflected present or infinitive + article

-VB+IN: verb, base: uninflected present, imperative or infinitive + preposition

-VB+JJ: verb, base: uninflected present, imperative or infinitive + adjective

-VB+PPO: verb, uninflected present tense + pronoun, personal, accusative

-VB+RP: verb, imperative + adverbial particle

-VB+TO: verb, base: uninflected present, imperative or infinitive + infinitival to

-VB+VB: verb, base: uninflected present, imperative or infinitive; hypenated pair

-VBD: verb, past tense

-VBG: verb, present participle or gerund

-VBG+TO: verb, present participle + infinitival to

-VBN: verb, past participle

-VBN+TO: verb, past participle + infinitival to

-VBZ: verb, present tense, 3rd person singular

No hace falta que escribas los tags, pero tomate un tiempo para buscar a través de las diferencias que hay entre el Penn Treebank y el Brown tagger. El tagset Penn Treebank está basado en el Brown, pero reduce la cantidad de etiquetas significativamente para eliminar la redundancia léxica y sintáctica.

0.1.4 4. Explorando el tagged corpora

Las probabilidades de un algoritmos de POS tagging son estimadas con un corpora. Este corpora fue anotado (etiquetado) a mano por lingüistas. Para visualizar cómo es un corpora de estos, hace lo siguiente:

[4]:
```
from nltk.corpus import treebank
```

[5]:
```
treebank.fileids()
```

[5]:
```
['wsj_0001.mrg',
 'wsj_0002.mrg',
 'wsj_0003.mrg',
 'wsj_0004.mrg',
 'wsj_0005.mrg',
 'wsj_0006.mrg',
 'wsj_0007.mrg',
 'wsj_0008.mrg',
 'wsj_0009.mrg',
 'wsj_0010.mrg',
 'wsj_0011.mrg',
 'wsj_0012.mrg',
 'wsj_0013.mrg',
 'wsj_0014.mrg',
 'wsj_0015.mrg',
 'wsj_0016.mrg',
 'wsj_0017.mrg',
 'wsj_0018.mrg',
 'wsj_0019.mrg',
 'wsj_0020.mrg',
 'wsj_0021.mrg',
 'wsj_0022.mrg',
 'wsj_0023.mrg',
 'wsj_0024.mrg',
 'wsj_0025.mrg',
 'wsj_0026.mrg',
```

```
'wsj_0027.mrg',
'wsj_0028.mrg',
'wsj_0029.mrg',
'wsj_0030.mrg',
'wsj_0031.mrg',
'wsj_0032.mrg',
'wsj_0033.mrg',
'wsj_0034.mrg',
'wsj_0035.mrg',
'wsj_0036.mrg',
'wsj_0037.mrg',
'wsj_0038.mrg',
'wsj_0039.mrg',
'wsj_0040.mrg',
'wsj_0041.mrg',
'wsj_0042.mrg',
'wsj_0043.mrg',
'wsj_0044.mrg',
'wsj_0045.mrg',
'wsj_0046.mrg',
'wsj_0047.mrg',
'wsj_0048.mrg',
'wsj_0049.mrg',
'wsj_0050.mrg',
'wsj_0051.mrg',
'wsj_0052.mrg',
'wsj_0053.mrg',
'wsj_0054.mrg',
'wsj_0055.mrg',
'wsj_0056.mrg',
'wsj_0057.mrg',
'wsj_0058.mrg',
'wsj_0059.mrg',
'wsj_0060.mrg',
'wsj_0061.mrg',
'wsj_0062.mrg',
'wsj_0063.mrg',
'wsj_0064.mrg',
'wsj_0065.mrg',
'wsj_0066.mrg',
'wsj_0067.mrg',
'wsj_0068.mrg',
'wsj_0069.mrg',
'wsj_0070.mrg',
'wsj_0071.mrg',
'wsj_0072.mrg',
'wsj_0073.mrg',
```

```
'wsj_0074.mrg',
'wsj_0075.mrg',
'wsj_0076.mrg',
'wsj_0077.mrg',
'wsj_0078.mrg',
'wsj_0079.mrg',
'wsj_0080.mrg',
'wsj_0081.mrg',
'wsj_0082.mrg',
'wsj_0083.mrg',
'wsj_0084.mrg',
'wsj_0085.mrg',
'wsj_0086.mrg',
'wsj_0087.mrg',
'wsj_0088.mrg',
'wsj_0089.mrg',
'wsj_0090.mrg',
'wsj_0091.mrg',
'wsj_0092.mrg',
'wsj_0093.mrg',
'wsj_0094.mrg',
'wsj_0095.mrg',
'wsj_0096.mrg',
'wsj_0097.mrg',
'wsj_0098.mrg',
'wsj_0099.mrg',
'wsj_0100.mrg',
'wsj_0101.mrg',
'wsj_0102.mrg',
'wsj_0103.mrg',
'wsj_0104.mrg',
'wsj_0105.mrg',
'wsj_0106.mrg',
'wsj_0107.mrg',
'wsj_0108.mrg',
'wsj_0109.mrg',
'wsj_0110.mrg',
'wsj_0111.mrg',
'wsj_0112.mrg',
'wsj_0113.mrg',
'wsj_0114.mrg',
'wsj_0115.mrg',
'wsj_0116.mrg',
'wsj_0117.mrg',
'wsj_0118.mrg',
'wsj_0119.mrg',
'wsj_0120.mrg',
```

```
'wsj_0121.mrg',
'wsj_0122.mrg',
'wsj_0123.mrg',
'wsj_0124.mrg',
'wsj_0125.mrg',
'wsj_0126.mrg',
'wsj_0127.mrg',
'wsj_0128.mrg',
'wsj_0129.mrg',
'wsj_0130.mrg',
'wsj_0131.mrg',
'wsj_0132.mrg',
'wsj_0133.mrg',
'wsj_0134.mrg',
'wsj_0135.mrg',
'wsj_0136.mrg',
'wsj_0137.mrg',
'wsj_0138.mrg',
'wsj_0139.mrg',
'wsj_0140.mrg',
'wsj_0141.mrg',
'wsj_0142.mrg',
'wsj_0143.mrg',
'wsj_0144.mrg',
'wsj_0145.mrg',
'wsj_0146.mrg',
'wsj_0147.mrg',
'wsj_0148.mrg',
'wsj_0149.mrg',
'wsj_0150.mrg',
'wsj_0151.mrg',
'wsj_0152.mrg',
'wsj_0153.mrg',
'wsj_0154.mrg',
'wsj_0155.mrg',
'wsj_0156.mrg',
'wsj_0157.mrg',
'wsj_0158.mrg',
'wsj_0159.mrg',
'wsj_0160.mrg',
'wsj_0161.mrg',
'wsj_0162.mrg',
'wsj_0163.mrg',
'wsj_0164.mrg',
'wsj_0165.mrg',
'wsj_0166.mrg',
'wsj_0167.mrg',
```

```
 'wsj_0168.mrg',
 'wsj_0169.mrg',
 'wsj_0170.mrg',
 'wsj_0171.mrg',
 'wsj_0172.mrg',
 'wsj_0173.mrg',
 'wsj_0174.mrg',
 'wsj_0175.mrg',
 'wsj_0176.mrg',
 'wsj_0177.mrg',
 'wsj_0178.mrg',
 'wsj_0179.mrg',
 'wsj_0180.mrg',
 'wsj_0181.mrg',
 'wsj_0182.mrg',
 'wsj_0183.mrg',
 'wsj_0184.mrg',
 'wsj_0185.mrg',
 'wsj_0186.mrg',
 'wsj_0187.mrg',
 'wsj_0188.mrg',
 'wsj_0189.mrg',
 'wsj_0190.mrg',
 'wsj_0191.mrg',
 'wsj_0192.mrg',
 'wsj_0193.mrg',
 'wsj_0194.mrg',
 'wsj_0195.mrg',
 'wsj_0196.mrg',
 'wsj_0197.mrg',
 'wsj_0198.mrg',
 'wsj_0199.mrg']
```

Esto desplegará una lista de archivos con extensión .mrg que son los archivos anotados del Penn Treebank. (NLTK solo contiene el 10% del corpus total de Penn Treebank, el corpus total tiene aproximadamente 1 millón de palabras) Ahora, para ver el archivo junto con los POS tags, podes usar:

[6]:
```
treebank.tagged_words("wsj_0001.mrg")[0:]
```

[6]:
```
[('Pierre', 'NNP'),
 ('Vinken', 'NNP'),
 (',', ','),
 ('61', 'CD'),
 ('years', 'NNS'),
 ('old', 'JJ'),
 (',', ','),
 ('will', 'MD'),
```

```
 ('join', 'VB'),
 ('the', 'DT'),
 ('board', 'NN'),
 ('as', 'IN'),
 ('a', 'DT'),
 ('nonexecutive', 'JJ'),
 ('director', 'NN'),
 ('Nov.', 'NNP'),
 ('29', 'CD'),
 ('.', '.'),
 ('Mr.', 'NNP'),
 ('Vinken', 'NNP'),
 ('is', 'VBZ'),
 ('chairman', 'NN'),
 ('of', 'IN'),
 ('Elsevier', 'NNP'),
 ('N.V.', 'NNP'),
 (',', ','),
 ('the', 'DT'),
 ('Dutch', 'NNP'),
 ('publishing', 'VBG'),
 ('group', 'NN'),
 ('.', '.')]
```

El Penn Treebank está compuesto por noticias financieras del Wall Street Journal publicadas alrededor de 1989. Dado que este archivo es solo el primero de los archivos anotados, todo el mundo académico que hace research en Procesamiento del Lenguja e Natural está familiarizado con este contenido. Podes proeguntar sobre la edad de Pierre Vinkin a casi cualquier investigador de NLP y van entender el chiste :) Mira todas las palabras taggeadas como sustantivos (NN). Basado en las distinciones que observaste en la sección previa, vescómo diferentes tipos de tags de sustantivos fueron asignados a la data? En este caso, [0 :] hace la query en el archivo entero wsj_0001.mrg. Para archivos más largos, usa un rango más corto. Tratá de ver las primeras 100 palabras de wsj_0003.mrg

[7]:
```
x100 = treebank.tagged_words("wsj_0003.mrg")[0:100]
```

[8]:
```
for p in x100:
    print(p)
```

```
('A', 'DT')
('form', 'NN')
('of', 'IN')
('asbestos', 'NN')
('once', 'RB')
('used', 'VBN')
('*', '-NONE-')
('*', '-NONE-')
```

```
('to', 'TO')
('make', 'VB')
('Kent', 'NNP')
('cigarette', 'NN')
('filters', 'NNS')
('has', 'VBZ')
('caused', 'VBN')
('a', 'DT')
('high', 'JJ')
('percentage', 'NN')
('of', 'IN')
('cancer', 'NN')
('deaths', 'NNS')
('among', 'IN')
('a', 'DT')
('group', 'NN')
('of', 'IN')
('workers', 'NNS')
('exposed', 'VBN')
('*', '-NONE-')
('to', 'TO')
('it', 'PRP')
('more', 'RBR')
('than', 'IN')
('30', 'CD')
('years', 'NNS')
('ago', 'IN')
(',', ',')
('researchers', 'NNS')
('reported', 'VBD')
('0', '-NONE-')
('*T*-1', '-NONE-')
('.', '.')
('The', 'DT')
('asbestos', 'NN')
('fiber', 'NN')
(',', ',')
('crocidolite', 'NN')
(',', ',')
('is', 'VBZ')
('unusually', 'RB')
('resilient', 'JJ')
('once', 'IN')
('it', 'PRP')
('enters', 'VBZ')
('the', 'DT')
('lungs', 'NNS')
(',', ',')
```

```
('with', 'IN')
('even', 'RB')
('brief', 'JJ')
('exposures', 'NNS')
('to', 'TO')
('it', 'PRP')
('causing', 'VBG')
('symptoms', 'NNS')
('that', 'WDT')
('*T*-1', '-NONE-')
('show', 'VBP')
('up', 'RP')
('decades', 'NNS')
('later', 'JJ')
(',', ',')
('researchers', 'NNS')
('said', 'VBD')
('0', '-NONE-')
('*T*-2', '-NONE-')
('.', '.')
('Lorillard', 'NNP')
('Inc.', 'NNP')
(',', ',')
('the', 'DT')
('unit', 'NN')
('of', 'IN')
('New', 'JJ')
('York-based', 'JJ')
('Loews', 'NNP')
('Corp.', 'NNP')
('that', 'WDT')
('*T*-2', '-NONE-')
('makes', 'VBZ')
('Kent', 'NNP')
('cigarettes', 'NNS')
(',', ',')
('stopped', 'VBD')
('using', 'VBG')
('crocidolite', 'NN')
('in', 'IN')
('its', 'PRP$')
('Micronite', 'NN')
('cigarette', 'NN')
('filters', 'NNS')
```

Esto va a imprimir las primeras 100 palabras en formato bonito.

0.1.5 5. Contando desde una corpora

A veces, nos preguntamos (para entender mejor el texto que estamos procesando, entre otras cosas), si una palabra en particular, aparece más frecuentemente como sustantivo o como verbo. Podemos entonces contar la frecuencia en un corpora para ver los resultados empíricos de nuestra duda. En este caso, vamos a ver si la palabra inglesa 'race' (carrera) aparece más veces como verbo o como sustantivo, ya que puede adoptar los dos significados. En NLTK, un par word-tag es representado como una tupla (word, tag). Vamos a crear primero dos tuplas correspondiendo a nuestras dos hipótesis:

[9]:
```
race1 = nltk.tag.str2tuple('race/NN')
```

[10]:
```
race2 = nltk.tag.str2tuple('race/VB')
```

Por defecto, NLTK contiene solo ejemplos del corpues Penn Treebank, así que usaremos también el Brown corpus para los estimados.

[11]:
```
from nltk.corpus import brown
```

[12]:
```
len(brown.tagged_words())
```

[12]: 1161192

¿Cuál es el tamaño del Brown corpus?

El tamaño es 1161192

[13]:
```
brown.tagged_words().count(race1)
```

[13]: 94

[14]:
```
brown.tagged_words().count(race2)
```

[14]: 4

Cuál es el uso más frecuente, cómo verbo o como sustantivo en el Brown corpus?

El uso más frecuente de "race" es como sustantivo.

El POS tagger HMM que se detalla en las diapositivas de la clase, usa dos fuentes de información. Una es la probabilidad de una palabra dado un tag particular, p(w i |t i). Cuál es la otra fuente de información que te ayuda a tagear una oración?

La otra fuente de información es P(Ti|Ti-1) que significa la probabilidad de un tag dado un tag anterior

0.1.6 Aplicando POS tagging a una nueva oración

Estudiamos el algoritmo HMM Viterbi para POS tagging. NLTK incluye una implementación del HMM tagger así como un par de otros.

El primer tagger que usaremos sería el UnigramTagger . Basado en el nombre, podrías adivinar que va a estar haciendo algo realmente simple y probablemente no se usa en ningún contexto. Así es! El UnigramTagger guarda el tag más frecuente para cada palabra sobre los datos de entretamiento (trainning set). Cuando ve que una palabra en una oración, le asignará el tag más frecuente a la misma. Así que, podemos decir que no tiene en cuenta el contexto.

Primero vamos a entregar un tagger de unigrams usando 5000 oraciones del Brown corpus como training data. Usando el argumento categories=news seleccionamos solo los documentos de noticias de la colección diversa que tiene el corpus Brown.

[15]:
```
from nltk.corpus import brown
```

[16]:
```
unigram_tagger = nltk.tag.UnigramTagger(brown.tagged_sents(categories='news')[:
↪5000])
```

Ahora unigram_tagger es un objeto que contiene el modelo entrenado. Aplicamos este modelo a nuestra frase de prueba "The Secretariat is expected to race tomorrow." PARA! Antes de ejecutar este código, qué TAG te parece que le será asignado a la palabra "race" en nuestro modelo? (sustantivo o verbo?)

[17]:
```
from nltk import word_tokenize
```

[18]:
```
S = "The Secretariat is expected to race tomorrow."
```

[19]:
```
S_tok = word_tokenize(S)
```

[20]:
```
unigram_tagger.tag(S_tok)
```

[20]:
```
[('The', 'AT'),
 ('Secretariat', 'NN-TL'),
 ('is', 'BEZ'),
 ('expected', 'VBN'),
 ('to', 'TO'),
 ('race', 'NN'),
 ('tomorrow', 'NR'),
 ('.', '.')]
```

El unigram tagger asignó un sustantivo a la palabra "race", pero en este caso se trata de un verbo.

Ahora vamos a usar el HMM tagger que toma en cuenta el contexto.

[21]:
```
hmm_tagger = nltk.hmm.HiddenMarkovModelTrainer().train_supervised(brown.
↪tagged_sents(categories="news")[:5000])
```

[22]:
```
hmm_tagger.tag(S_tok)
```

[22]:
```
[('The', 'AT'),
 ('Secretariat', 'NN-TL'),
 ('is', 'BEZ'),
```

```
 ('expected', 'VBN'),
 ('to', 'TO'),
 ('race', 'VB'),
 ('tomorrow', 'NR'),
 ('.', '.')]
```

Qué tag le asignó este modelo a la palabra "race"? En este caso, el HMM tagger identificó correctamente a la palabra "race" como verbo

Probá los dos modelos Unigram y HMM para otras sentencias ambiguas que se te ocurran (por ejemplo bank, duck, etc!)

Pruebo con la oración "The man water the plants"

[23]:
```
S2 = "The man water the plants"
```

[24]:
```
S2_tok = word_tokenize(S2)
```

[25]:
```
unigram_tagger.tag(S2_tok)
```

[25]:
```
[('The', 'AT'),
 ('man', 'NN'),
 ('water', 'NN'),
 ('the', 'AT'),
 ('plants', 'NNS')]
```

[26]:
```
hmm_tagger.tag(S2_tok)
```

[26]:
```
[('The', 'AT'),
 ('man', 'NN'),
 ('water', 'NN'),
 ('the', 'AT'),
 ('plants', 'NNS')]
```

En este caso, ambos taggers consideran erróneamente a "water" como un sustantivo.

Pruebo con la oración "Larry made a quick duck to avoid hitting his head"

[27]:
```
S3 = "Larry made a quick duck to avoid hitting his head"
```

[28]:
```
S3_tok = word_tokenize(S3)
```

[29]:
```
unigram_tagger.tag(S3_tok)
```

[29]:
```
[('Larry', 'NP'),
 ('made', 'VBN'),
 ('a', 'AT'),
 ('quick', 'JJ'),
 ('duck', None),
 ('to', 'TO'),
```

```
 ('avoid', 'VB'),
 ('hitting', 'VBG'),
 ('his', 'PP$'),
 ('head', 'NN')]
```

[30]:
```
hmm_tagger.tag(S3_tok)
```

[30]:
```
[('Larry', 'NP'),
 ('made', 'VBD'),
 ('a', 'AT'),
 ('quick', 'JJ'),
 ('duck', 'AT'),
 ('to', 'AT'),
 ('avoid', 'AT'),
 ('hitting', 'AT'),
 ('his', 'AT'),
 ('head', 'AT')]
```

En este caso, ninguno logra identificar a "duck" como verbo. Unigram no le asigna ningún tag y HMM le asigna un artículo

Ahora hagamos este ejercicio con el NLTK book. Es posible encontrar títulos de noticias ambiguos como este: "Juvenile Court to Try Shooting Defendant" Manualmente (a mano) taggea este titular para ver si tu conocimiento del Part of speech tagging remueve la ambigüedad. Una aproximación está bien, no hace falta que uses exactamente los tags del tagset.

-"Juvenile" : JJ (adjetivo)

-"Court": NN (sustantivo)

-"To": AT (artículo)

-"Try": VB (verbo)

-"Shooting": VBG (gerundio usado como adjetivo)

-"Defendant": NN (sustantivo)

Ahora corré el HMM tagger en esta oración, te dio los mismos tags?

[31]:
```
S4 = "Juvenile court to try shooting defendant."
```

[32]:
```
S4_tok = word_tokenize(S4)
```

[33]:
```
hmm_tagger.tag(S4_tok)
```

[33]:
```
[('Juvenile', 'JJ-TL'),
 ('court', 'NN-TL'),
```

```
('to', 'TO'),
('try', 'VB'),
('shooting', 'VBG'),
('defendant', 'NN'),
('.', '.')]
```

Todos excepto ‘to’