082057 – Procesamiento del Lenguaje Natural

Lab 2

Parte del Discurso (Part of Speech) (POS)

1 De que 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

2 NLTK environment

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

```
>>> import nltk

>>> nltk.download()
```

En el cuadro de diálogo, elija “book” que bajará todo el corpora usado en los ejemplos del libro de NLTK (http://www.nltk.org/book/). Clickea en download y cuando termine, cerra el cuadro de diálogo.

Pudes importar el corpora, tipeando:

```
>>> from nltk.book import *
```

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?

```
>>> nltk.help.upenn_tagset(".*")
```

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.

$: dollar
    $ -$ --$ A$ C$ HK$ M$ NZ$ S$ U.S.$ US$

'': closing quotation mark
    ' ''

...
MD: modal auxiliary
    can cannot could couldn't dare may might must need ought shall should...
NN: noun, common, singular or mass
    common-carrier cabbage knuckle-duster casino afghan shed thermostat investment ...

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.

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

```
>>> nltk.help.brown tagset("NN*")
```

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.

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:

```
>>> from nltk.corpus import treebank
>>> treebank.fileids()
```

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:

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

(u‘Pierre’, u‘NNP’), (u‘Vinken’, u‘NNP’), (u‘,’, u’,’), (u‘61’, u‘CD’), (u‘years’, u‘NNS’),...
```

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

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

>>> for p in x100:
        print p
```

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

5 Contando desde un corpora

In class last week, we discussed an example sentence “The Secretariat is expected to race next week.” and we hypothesized in general English (not in this particular sentence, but general use) whether the word ‘race’ is more frequently a noun or a verb. Get counts from corpora to empirically see the result.

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:

```
>>> race1 = nltk.tag.str2tuple(‘race/NN’)
```

```
>>> 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.

```
>>> from nltk.corpus import brown

>>> len(brown.tagged_words())
```

Cuál es el tamaño del Brown corpus?

Ahora estimá cuán frecuentemente aparecen estas tuplas en el Brown corpus.

```
>>> brown.tagged_words().count(race1)

>>> brown.tagged words().count(race2)
```

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

Era lo que te imaginabas?

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?

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.

```
>>> from nltk.corpus import brown

>>> 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?)

```
>>> from nltk import word_tokenize

>>> S = "The Secretariat is expected to race tomorrow."

>>> S_tok = word_tokenize(S)

>>> unigram_tagger.tag(S_tok)
```

Acertaste?

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

```
>>> hmm_tagger =
nltk.hmm.HiddenMarkovModelTrainer().train_supervised(brown.tagged_sents(categories="news")[:50
00])

>>> hmm_tagger.tag(S tok)
```

Qué tag le asignó este modelo a la palabra “race”?

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

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.

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