UNIVERSIDAD ADOLFO IBÁÑEZ

Procesamiento de Lenguaje Natural

DIPLOMA/MAGISTER EN INTELIGENCIA ARTIFICIAL
UNIVERSIDAD ADOLFO IBÁÑEZ

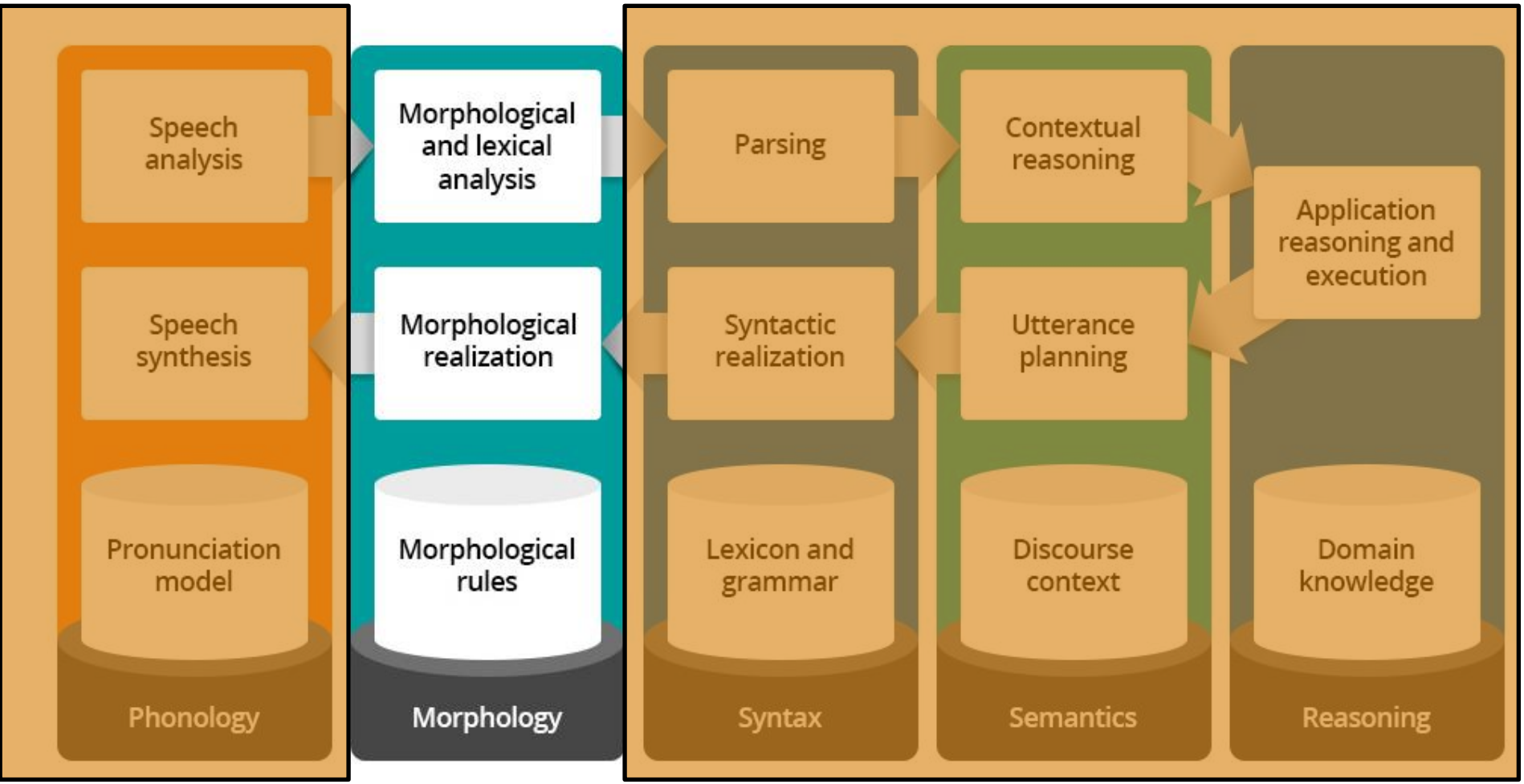
Profesor: Dr. John Atkinson

Análisis Léxico

OBJETIVO

Entender la forma en que se convierten secuencias de caracteres en secuencias de *tokens* o símbolos que poseen una función lingüística *individual*.

ETAPAS EN NLP



Análisis Léxico

- Tarea de identificar automáticamente **unidades léxicas** (UL) con significado individual, respecto a su función en un texto en lenguaje natural.

- ¿Porqué? El análisis morfológico no es suficiente, ya que podría generar unidades no válidas en el lenguaje y no ve la for a de las palabras en su context (ej. **militar**).

Unidad Léxica

- Una *UL* es una *palabra* que es la unidad básica de un diccionario (*lexicón*).
- Pero esto es ambigüo: los strings “*cajero*" y “*cajeros*" son diferentes formas morfológicas de la misma entidad en el diccionario!.

¿Deberíamos tratarlas igual?

Análisis Léxico

Luego:

1. ¿Es una palabra una UL válida?
2. Si es válida, ¿De qué *tipo* es?

Análisis Léxico

- Usualmente el *tipo* de una palabra está asociada a la función lingüística que esta cumple en el habla o *lenguaje*.

- Esta *función* se denomina Parte del Habla (*Part-Of-Speech* o POS)

Ejemplo (*Español*)

Texto:

El	cliente	puso	una	queja	en	el	mesón

Ejemplo (*Español*)

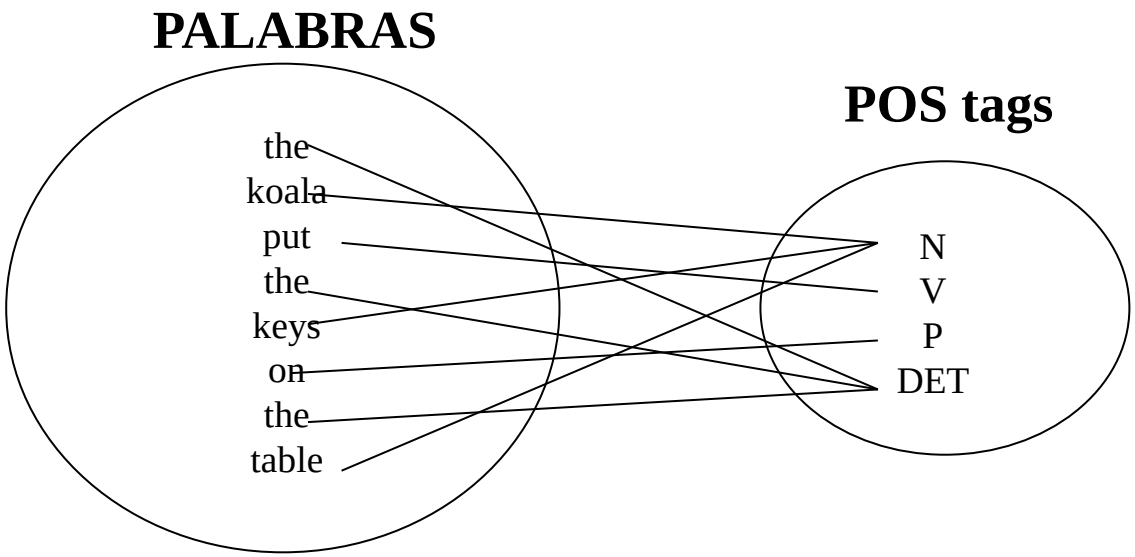
Texto:

El	cliente	puso	una	queja	en	el	mesón
Art.	Nombre	Verbo	Art.	Nombre	Prep.	Art.	Nombre

POS

Etiquetado POS

La tarea de asignar una etiqueta *POS* a cada palabra en un texto se denomina *POS tagging*.



Muchas Aplicaciones

- *Auto-complete de palabras en mensajes de celulares.*
- *Análisis de sentimientos de una opinión.*
- *Identificar y extraer información clave de oraciones de un cliente o de un conjunto de documentos.*
- *Reconocimiento de nombres de entidades importantes en un texto.*
- *Tarea clave preliminar para realizar análisis sintáctico.*
- *etc*

TAG SETS

- Para realizar *POS tagging*, necesitamos elegir un conjunto estándar de tags con los cuales trabajar (TAG SET)
- Podríamos utilizar *tagsets* de "grano grueso": N, V, Adj, Adv, etc
- Los sets de "*grano fino*" más comunes son *TreeBanks* (45 tipos de tags), WSJ, etc.

Uso de TAGSETS en POS Tagging

```
El          ganador                    de
pasapalabras      recibió
50    millones     .
```

Uso de TAGSETS en POS Tagging

El/*DT* ganador/*JJ* de/*IN*
pasapalabras/*NN* recibió/*VBD*
50/*CD* millones/*NNS* ./.

POS Tagging: *Ambigüedad*

Las palabras tienen usualmente más de un POS: ***back***

```
The back door = Adjetivo (JJ)
On my back = Nombre o Sustantivo (NN)
Win the voters back = Adverbio (RB)
Promised to back the bill = Verbo (VB)
```

Tarea: Determinar automáticamente el *mejor* tag de tipo POS para cada palabra en un texto.

Métodos de POS Tagging

- Tagging basado en Reglas.
- Tagging Estadístico.
- Tagging basado en Error.
- Tagging Estocástico.
- Tagging basado en Aprendizaje Automático.

Métodos de POS Tagging

- Tagging basado en Reglas.
- Tagging Estadístico.
- Tagging basado en Error.
- Tagging Estocástico.
- Tagging basado en Aprendizaje Automático.

Tagging Estocástico

- Basado en ***Modelos Ocultos de Markov*** (*Hidden Markov Model* o HMM) para asignar una secuencia de etiquetas POS a una secuencia de palabras (*oración de entrada*).
- Es un caso especial de inferencia *Bayesiana*.
- Útil para analizar datos *secuenciales*.
- También se le conoce como el modelo de *canal con ruido* visto previamente.

Tagging como Clasificación

Tenemos una oración de entrada:

Secuencia de “observaciones” (palabras)

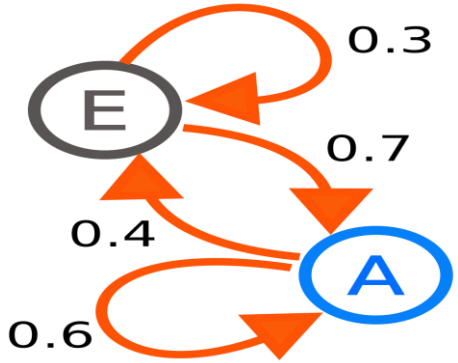
¿Cuál es la mejor secuencia de etiquetas POS que corresponde a una secuencia de observaciones dada?

Visión Probabilística:

- Considerar todas las secuencias posibles de *tags*.
- Elegir la secuencia de *tags* que es más probable, dada una *secuencia de observaciones* de ***n*** palabras ***w1…wn***.

Modelo de Markov simple

Un *modelo de Markov* es un autómata *probabilístico* compuesto de *transiciones* y *estados*, que permite realizar tareas de predicción y clasificación.



Formalmente

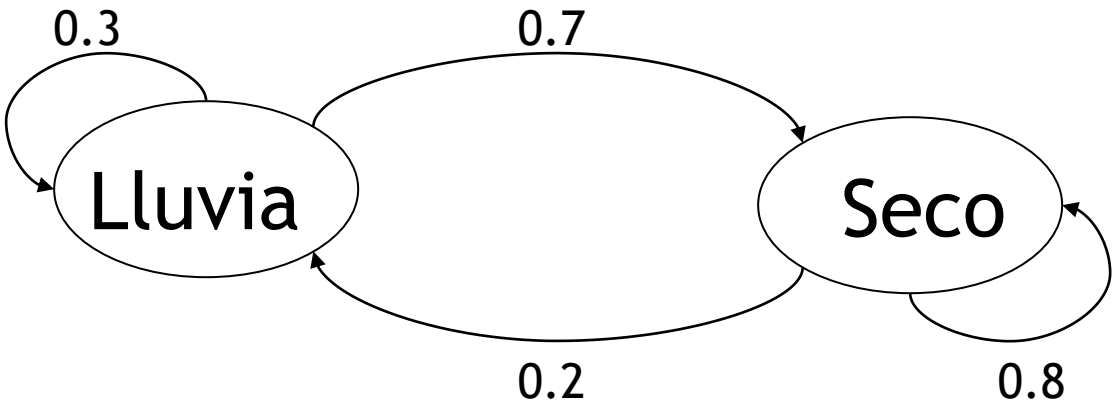
- Conjunto de estados: $\{s_1, s_2, \ldots, s_N\}$
- La máquina se mueve de un estado a otro generando una secuencia de estados: $s_{i1}, s_{i2}, \ldots, s_{ik}, \ldots$
- *Propiedad de la cadena* de Markov:

  La probabilidad de cada estado subsecuente depende solamente de su estado previo: $P(s_{ik} \mid s_{i1}, s_{i2}, \ldots, s_{ik-1}) = P(s_{ik} \mid s_{ik-1})$

- Para definir un **modelo de Markov**, se deben estimar las siguientes probabilidades:
  - *Probabilidades de transición:* $a_{ij} = P(s_i \mid s_j)$
  - *Probabilidades iniciales:* $\pi_i = P(s_i)$

EJEMPLO

Máquina de pronóstico del clima:



Dos estados : **‘Lluvia’ y ‘Seco’**
Probs. de Transición:
P(‘Lluvia’|‘LLuvia’)=0.3 P(‘Seco’|‘Lluvia’)=0.7
P(‘Lluvia’|‘Seco’)=0.2 P(‘Seco’|‘Seco’)=0.8
Probs. Iniciales: asumamos **P(‘Lluvia’)=0.4 P(‘Seco’)=0.6**

Consideraciones

1. ¿De dónde se obtienen las probabilidades iniciales y de transición?
2. ¿Cómo se calcula la probabilidad de una secuencia?

Consideraciones

1. ¿De dónde se obtienen las probabilidades iniciales y de transición?
2. ¿Cómo se estima la probabilidad de una secuencia?

Muestreo:

```
A B B ..
A A A B
..
A   B   A
B..
A A B B…
…
```

	Next		
Current	A	B	End
Start	0.7	0.3	0
A	0.2	0.7	0.1
B	0.7	0.2	0.1

Consideraciones

1. De dónde se obtienen las probabilidades iniciales y de transición?
2. ¿Cómo se estima la probabilidad de una secuencia?

Usando la propiedad de la cadena de *Markov*:

$$P(s_{i1}, s_{i2}, \ldots, s_{ik}) = P(s_{ik} \mid s_{i1}, s_{i2}, \ldots, s_{ik-1}) P(s_{i1}, s_{i2}, \ldots, s_{ik-1})$$
$$= P(s_{ik} \mid s_{ik-1}) P(s_{i1}, s_{i2}, \ldots, s_{ik-1}) = \ldots$$
$$= P(s_{ik} \mid s_{ik-1}) P(s_{ik-1} \mid s_{ik-2}) \ldots P(s_{i2} \mid s_{i1}) P(s_{i1})$$

Limitaciones

- El modelo de Markov simple es útil como una máquina predictora de secuencias.
- PERO necesitamos un modelo que además produzca una salida: *secuencia de etiquetas POS*.
- Debemos modificar levemente el modelo de *Markov* original.

Modelo Oculto de Markov (*Hidden Markov Model)*

- Conjunto de estados: $\{S_1, S_2, \ldots, S_N\}$
- La máquina se mueve de un estado a otro generando una secuencia de estados: $S_{i1}, S_{i2}, \ldots, S_{ik}, \ldots$
- Propiedad de la "cadena" de Markov:

$$P(s_{ik} \mid s_{i1}, s_{i2}, \ldots, s_{ik-1}) = P(s_{ik} \mid s_{ik-1})$$

- Los estados **NO son visibles** pero cada uno genera aleatóreamente una de **M** observaciones de salida:

$$\{v_1, v_2, \ldots, v_M\}$$

Formalizando un HMM

Para definir un **HMM *M***, se deben estimar las siguientes probabilidades:

- *Matriz de probabilidades de transición*:
  $A=(a_{ij})$ $a_{ij}= P(s_i \mid s_j)$
- *Vector de probs. Iniciales*: $\pi=(\pi_i)$ $\pi_i = P(s_i)$
- *Matriz de probabilidades de observación*:
  $B=(b_i (v_m))$ $b_i(v_m) = P(v_m \mid s_i)$

$$M=(A, B, \pi)$$

¿Cómo Infiere la HMM?

A partir de todas las secuencias posibles de *n* etiquetas $t_1 \ldots t_n$, deseamos la secuencia de tags POS, tal que $P(t_1 \ldots t_n \mid w_1 \ldots w_n)$ sea **máxima**:

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n | w_1^n)$$

¿Cómo Estimamos?

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n | w_1^n)$$

Por regla de Bayes

$$\frac{P(w^n{}_1 \mid t^n{}_1) P(t^n{}_1)}{P(w^n{}_1)}$$

$$P(w_1^n | t_1^n) \sim \prod_{i=1} P(w_i | t_i)$$

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} \overbrace{P(w_1^n | t_1^n)}^{\text{likelihood}} \overbrace{P(t_1^n)}^{\text{prior}}$$

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n | w_1^n) \approx \underset{t_1^n}{\operatorname{argmax}} \prod_{i=1}^{n} P(w_i | t_i) P(t_i | t_{i-1})$$

$$P(t_1^n) \approx \prod_{i=1}^{n} P(t_i | t_{i-1})$$

¿Cómo Estimamos?

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} \prod_{i=1}^{n} P(w_i|t_i) P(t_i|t_{i-1})$$

Necesito una colección de textos (*corpus*) etiquetados para estimar las probabilidades!!

EJEMPLO

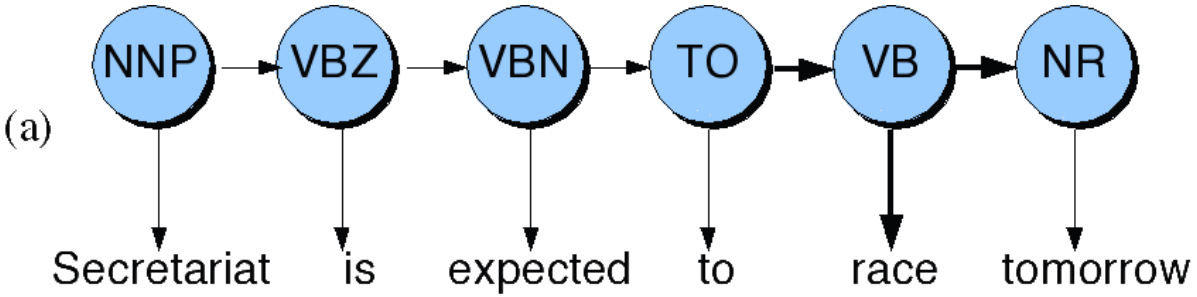
Texto Completo

Secretariat/NNP is/VBZ expected/VBN to/TO **race**/? tomorrow/NR ./.

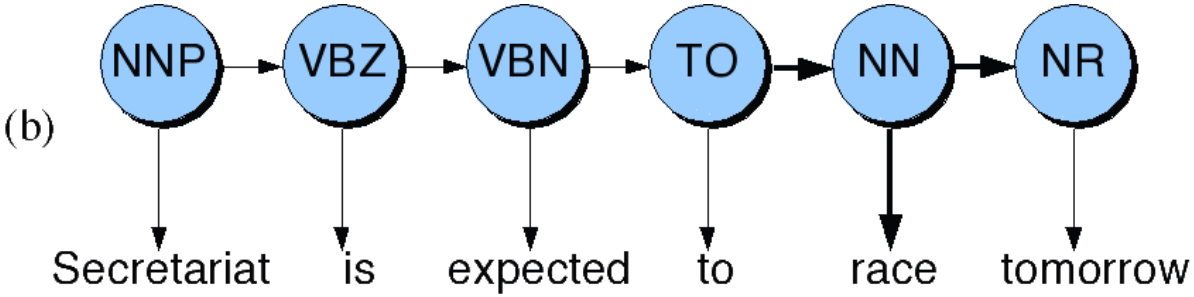
People/NNS continue/VB to/TO inquire/VB the/DT reason/NN for/IN the/DT **race**/? for/IN outer/JJ space/NN
…

¿Cómo seleccionamos el tag correcto para la palabra race?

¿Cómo desambigüamos race?



P(VB|race)



P(NN|race)

¿Cómo desambigüamos race?

Estimaciones desde textos de entrenamiento (etiquetados):

$P(NN|TO) = .00047$
$P(VB|TO) = .83$
$P(race|VB) = .00012$
$P(race|NN) = .00057$
$P(NR|VB) = .0027$
$P(NR|NN) = .0012$
…

Calculamos las probabilidades de dos tags posibles para "race":

$$P(\mathbf{VB}|race) = P(race|VB)P(VB|TO)P(NR|VB) = .00000027$$

o

$$P(\mathbf{NN}|race) = P(race|NN)P(NN|TO)P(NR|NN) = .00000000032$$

Luego, se le asigna la etiqueta **VB** a race

RESUMEN

- El **Análisis Léxico** es la tarea de identificar automáticamente una palabra y asociarla con su rol en un texto.
- Los métodos usuales para “etiquetar” (tagging) las palabras mediante su POS, incluyen los estadísticos, estocásticos, y basados en aprendizaje automático (ej. RNN, LSTM).
- La tarea de tagging tiene muchas aplicaciones, y consiste de un enfoque muy robusto, para extraer información desde textos en lenguaje natural.


Tiempo de Ejercicios

Ejercicio

- ✓ Cargue en *Google Colab* el programa **`POS_tagging`**.
- ✓ Cargue el texto de ejemplo “sample.txt”.