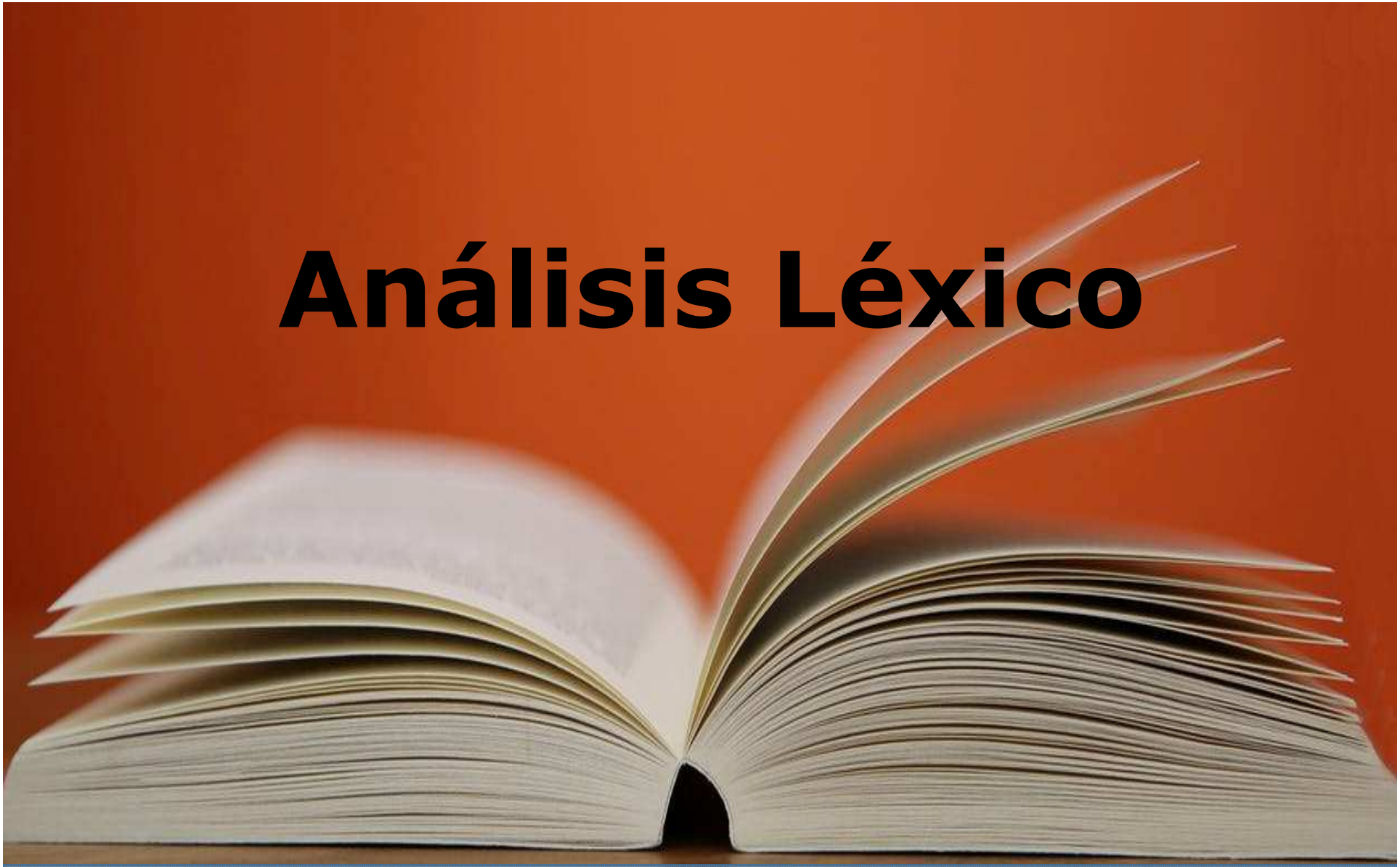
Procesamiento del Lenguaje Natural

Dr. John Atkinson

iUAI
UNIVERSIDAD ADOLFO IBÁÑEZ
FACULTAD DE INGENIERÍA Y CIENCIAS

Análisis Léxico



Procesamiento de Lenguaje Natural

iUAI
UNIVERSIDAD ADOLFO IBÁÑEZ
FACULTAD DE INGENIERÍA Y CIENCIAS

Introducción

Análisis léxico es el proceso de convertir una secuencia de caracteres de un texto en una secuencia de palabras ó "*tokens*" que poseen cierto significado lingüístico individual.

Unidades Léxicas (UL)

- ✓ Una UL es una *palabra* que es la unidad básica de un diccionario (*lexicón*).

- ✓ Pero esto es ambigüo: los strings "*teatro*" y "*teatros*" son diferentes formas de la misma entidad en el diccionario!!.

¿Deberíamos tratarlas igual?

Respuesta (1): SI

- ✓ Si estamos analizando información textual donde las variaciones morfológicas no son de interés, estas se deberían tratar como equivalentes:

- ✓ ¿Cómo se realiza?
  - ***Stemming***: tarea morfológica de reducir las formas derivadas de una palabra a su "tronco" (stem) ó forma raíz.
  - Ejemplos:

    policía → polici

    trataron → trat

Respuesta (2): NO

- ✓ Suponga que debemos construir un *Sistema de Pregunta-Respuesta* para interactuar con un cliente, y recibimos las siguientes consultas (*queries*):

A: "Encuentre los teatros más cercanos"

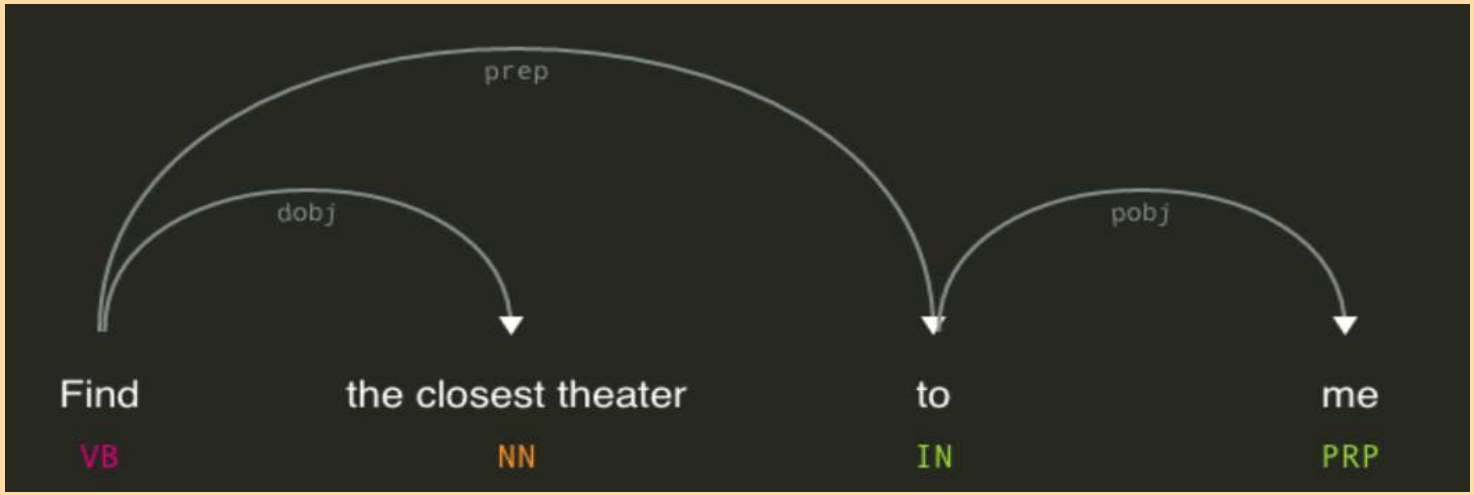
Ó bien,

B: "Encuentre el teatro más cercano"

Problemas

En A, el cliente está implicando que desea ver múltiples teatros, mientras que en B, sólo desea el teatro más cercano.

Claramente, eliminar la distinción *singular/plural* impactará negativamente nuestra aplicación.



Análisis Léxico

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

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

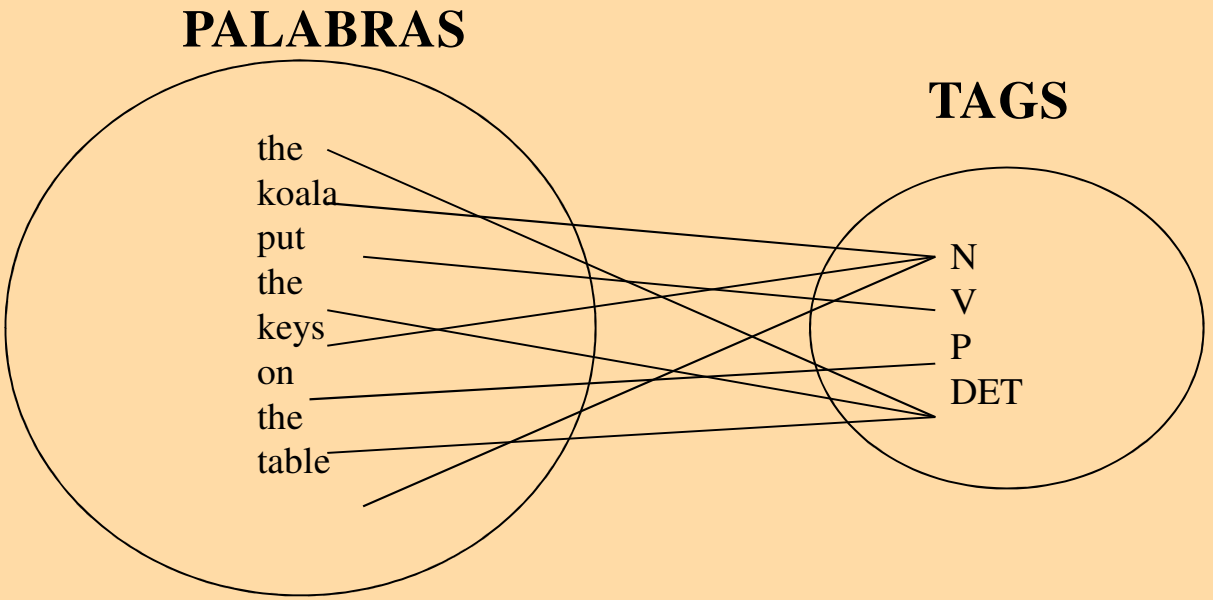
- ✓ Esta *función* se denomina "Parte del Habla" (*Part-Of-Speech* ó POS)

Ejemplo (*Español*):

PALABRA	POS tag
el	**ART**
cliente	**N**
puso	**V**
una	**ART**
queja	**N**
en	**P**
el	**ART**
mesón	**N**

POS Tagging

El proceso de asignar una etiqueta (*etiquetar*) POS a cada palabra en un corpus se denomina *POS tagging*:



Aplicaciones

- ✓ Auto-complete de palabras en mensajes de celulares.
- ✓ Predicción de "*movidas" de diálogo* de un cliente en una conversación.
- ✓ Análisis de sentimientos.
- ✓ Reconocimiento de nombres de entidades (NER) importantes de un texto.
- ✓ Muchas más..

Elección de un *TAG SET*

- ✓ 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
- ✓ El set de "grano fino" más común es el "*TreeBank*" tagset, 45 tags
  - PRP$, WRB, WP$, VBG, etc

Ejemplo de TAGSET WSJ

Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction	*and, but, or*	SYM	Symbol	*+,%, &*
CD	Cardinal number	*one, two, three*	TO	"to"	*to*
DT	Determiner	*a, the*	UH	Interjection	*ah, oops*
EX	Existential 'there'	*there*	VB	Verb, base form	*eat*
FW	Foreign word	*mea culpa*	VBD	Verb, past tense	*ate*
IN	Preposition/sub-conj	*of, in, by*	VBG	Verb, gerund	*eating*
JJ	Adjective	*yellow*	VBN	Verb, past participle	*eaten*
JJR	Adj., comparative	*bigger*	VBP	Verb, non-3sg pres	*eat*
JJS	Adj., superlative	*wildest*	VBZ	Verb, 3sg pres	*eats*
LS	List item marker	*1, 2, One*	WDT	Wh-determiner	*which, that*
MD	Modal	*can, should*	WP	Wh-pronoun	*what, who*
NN	Noun, sing. or mass	*llama*	WP$	Possessive wh-	*whose*
NNS	Noun, plural	*llamas*	WRB	Wh-adverb	*how, where*
NNP	Proper noun, singular	*IBM*	$	Dollar sign	*$*
NNPS	Proper noun, plural	*Carolinas*	#	Pound sign	*#*
PDT	Predeterminer	*all, both*	"	Left quote	(' or ")
POS	Possessive ending	*'s*	"	Right quote	(' or ")
PP	Personal pronoun	*I, you, he*	(	Left parenthesis	([, (, {, <)
PP$	Possessive pronoun	*your, one's*	)	Right parenthesis	( ],), }, >)
RB	Adverb	*quickly, never*	,	Comma	,
RBR	Adverb, comparative	*faster*	.	Sentence-final punc	(. ! ?)
RBS	Adverb, superlative	*fastest*	:	Mid-sentence punc	(: ; ... — -)
RP	Particle	*up, off*			

PRP
PRP$

Uso de un TAGSET en Tagging

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

Nota:

- ✓ El texto se "*etiqueta*" ó "*anota*" automáticamente con las etiquetas POS que mejor corresponden a cada palabra.

POS Tagging

- ✓ Las palabras tienen usualmente más de un POS: *back*
  - The *back* door = Adjetivo (JJ)
  - On my *back* = Sustantivo (NN)
  - Win the voters *back* = Adverbio (RB)
  - Promised to *back* the bill = Verbo (VB)

- ✓ *Tarea*: determinar el *tag correcto* para cada cada palabra.

Algunos Métodos de *POS Tagging*

1. *Tagging basado en Reglas*
2. *Tagging Estadístico*
3. *Tagging Estocástico*
   - Uso de HMM (*Hidden Markov Model*), ó relacionados (ej. CRF)
4. Tagging basado en Transformaciones
5. Tagging basado en Aprendizaje Automático

Tagging *basado en Reglas*

1. Comience con un *diccionario de palabras.*
2. Asigne todos los posibles *tags* a palabras del diccionario.
3. Escriba reglas a mano para *eliminar tags* selectivamente.
4. Asigne el *tag* correcto a cada palabra.

Problemas

- ✓ Es difícil codificar las reglas manualmente.
- ✓ Requiere muchas reglas.
- ✓ Existe mucha ambigüedad al etiquetar.
- ✓ Método no muy robusto.

Tagging *Estadístico*

- ✓ Calcule el tag más frecuente para una palabra (probabilidad).
- ✓ Se pierde el contexto: la misma palabra en contextos diferentes tiene el *mismo* POS.
- ✓ Generalmente no se usaría si se necesitara pocos datos etiquetados.
- ✓ Útil solamente como medida de comparación.

Evaluación de Rendimiento

- ✓ Asuma que tiene un set de palabras de prueba que ya fueron etiquetadas por un humano ("*Gold Standard*")

- ✓ Se podría aplicar un *POS tagger* sobre dicho set de prueba y revisar cuántos *tags* se etiquetaron correctamente (accuracy ó %correctas).

$$\%correct = \frac{\#\,of\ \ words\ \ tagged\ \ correctly\ \ in\ \ test\ \ set}{total\ \#\ \ of\ \ \ words\ \ in\ \ test\ \ set}$$

Tagging Estocástico

- ✓ Uso de *Hidden Markov Model* (HMM) para POS tagging.
- ✓ Es un caso especial de inferencia *Bayesiana*.
- ✓ También se le conoce como el modelo de "*canal con ruido*" visto previamente en análisis de voz.

Tagging como *Clasificación*

Tenemos una oración:

Secuencia de Observaciones (palabras)

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

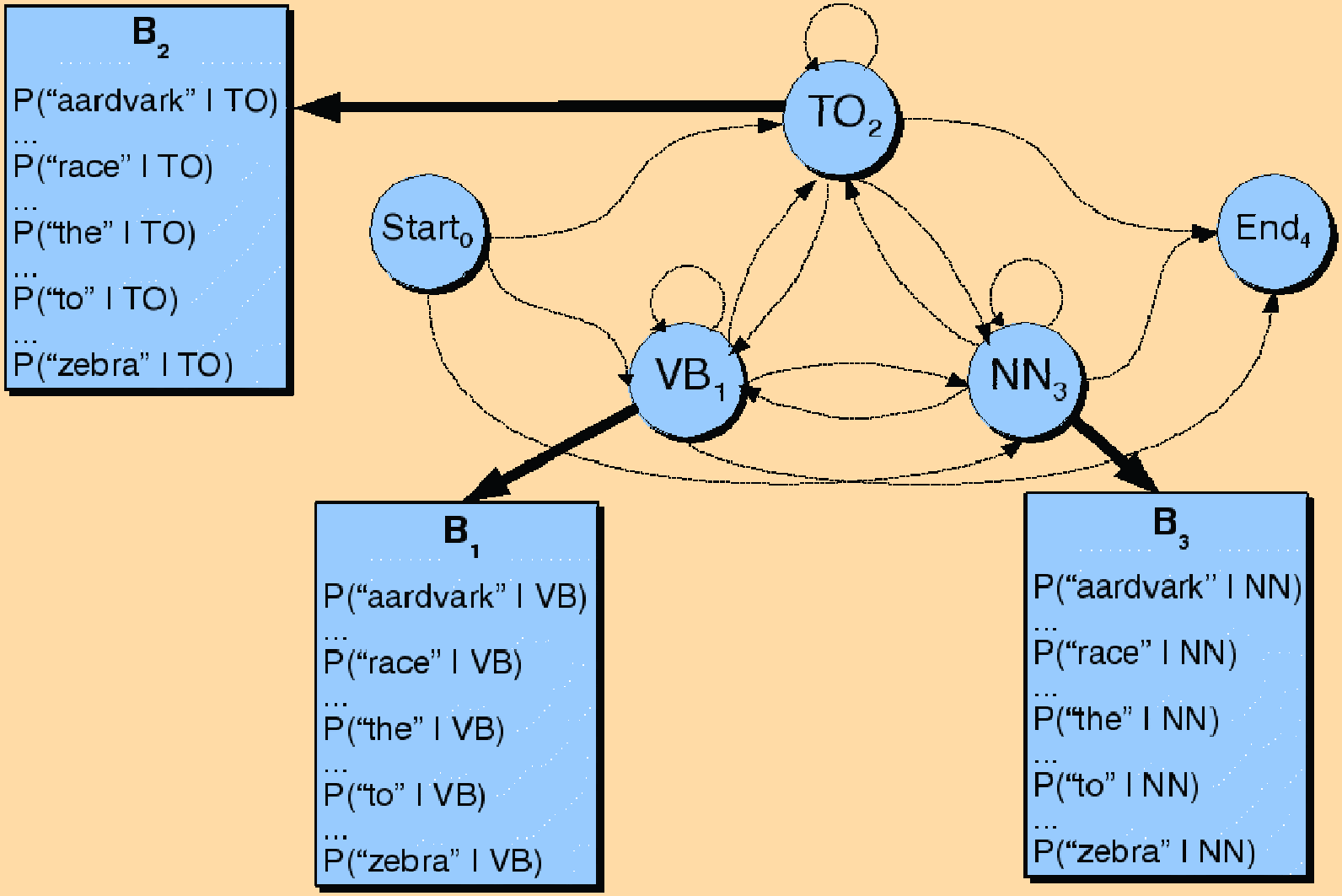
- Visión probabilística:
  - Considerar todas las secuencias posibles de *tags.*
  - Elegir la secuencia de *tags* que es más probable, dada una *secuencia de observaciones* de ***n*** palabras ***w1…wn.***

Modelo de Markov (HMM)

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

- ✓ Tareas posibles:
  - *Generar secuencias de estados de acuerdo a las probabilidades.*
  - *Computar la probabilidad de una secuencia.*

¿Cómo se vería una HMM?



¿Cómo se *Infiere*?

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

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

¿Cómo se estiman estos valores?

Intuición: *Clasificación Bayesiana!!*

Utilizar la regla de Bayes para transformar la ecuación principal en un conjunto de otras probabilidades que son más fáciles de estimar.

Uso de *Regla de Bayes*

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

Por regla de Bayes

$$P(w_1^n|t_1^n) \quad P(t_1^n)$$

$$\prod_{i=1}^{n} (w_i|t_i)$$

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

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

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

Clases de Probabilidades

Probabilidad de transición de tags: $P(t_i|t_{i-1})$

- Es más probable que artículos (DT) precedan adjetivos (JJ) y sustantivos (NN)
  - *That/DT flight/NN*
  - *The/DT yellow/JJ hat/NN*
  - Para calcular, por ejemplo, P(NN | DT) necesitamos:

$$P(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}$$

  - Por tanto,

$$P(NN|DT) = \frac{C(DT, NN)}{C(DT)} = \frac{56,509}{116,454} = .49$$

Clases de Probabilidades

Probabilidad de Palabras: **$P(w_i|t_i)$**

- La probabilidad que la palabra *"is"* tenga la etiqueta VBZ (*3sg pres verb*), usando

$$P(w_i|t_i) = \frac{C(t_i, w_i)}{C(t_i)}$$

- Sería, **P(is|VBZ)**:

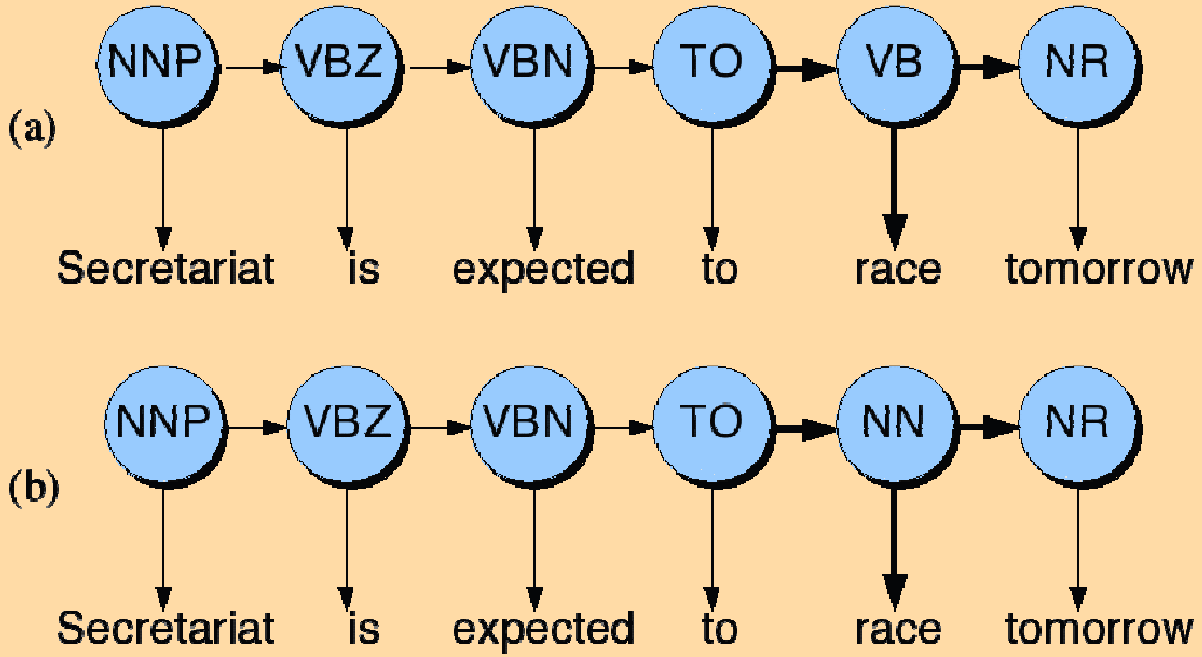
$$P(is|VBZ) = \frac{C(VBZ, is)}{C(VBZ)} = \frac{10,073}{21,627} = .47$$

Ejemplo: Tagging para "*race*"

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 la etiqueta correcta?

¿Cómo desambigüamos "*race*"?



¿Cómo desambigüamos "*race*"?

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

P(race|VB) P(VB|TO)P(NR|VB) = .00000027

P(race|NN) P(NN|TO)P(NR|NN) = .00000000032

Regla de la Cadena

Modelamiento del Lenguaje

- ✓ El modelo anterior también puede utilizarse para predecir secuencias de palabras.
- ✓ El modelo de "*canal con ruido" que estima* P(W), se denomina un *modelo de lenguaje*.

Modelo de Lenguaje -> *Gramática*
(secuencia de "*gramas*")

Aplicaciones

- ✓ *Extracción de información (IE) desde documentos.*
- ✓ *Clasificación de documentos.*
- ✓ *Agrupación de textos similares.*
- ✓ *Identificación de información clave en la interacción con un cliente/usuario.*

Ejemplo: *Reconocimiento de Entidades*

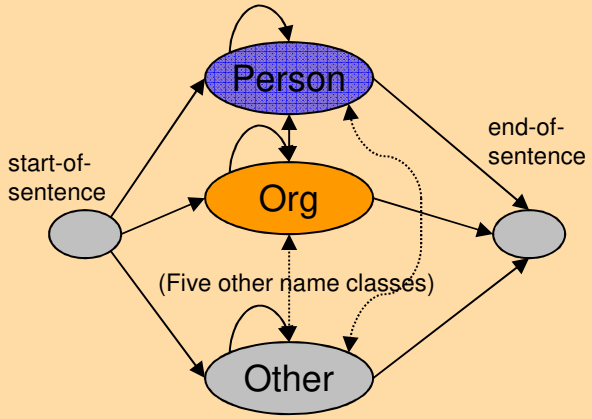
October 14, 2015, 4:00 a.m. PT

For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

"We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access."

Richard Stallman, founder of the Free Software Foundation, countered saying…



Una HMM para el texto

Ejemplo: *Reconocimiento de Entidades*

October 14, 2015, 4:00 a.m. PT

For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

"We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access."

Richard Stallman, founder of the Free Software Foundation, countered saying…

Entidades Reconocidas:

Microsoft Corporation
CEO
Bill Gates
Microsoft
Gates
Microsoft
Bill Veghte
Microsoft
VP
Richard Stallman
founder
Free Software Foundation

Tarea: *Named-Entity Recognition (NER)*

Resumen

- ✓ Análisis léxico es la tarea de identificar automáticamente una palabra y 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.
- ✓ La tarea de tagging tiene muchas aplicaciones, y consiste de un enfoque muy robusto, para extraer información básica desde textos (ej. NER).


Tiempo de Ejercicios

Ejercicio Grupal

1. Cargue en *Python* (vía *Spyder*) lo siguiente:
   a) tagging.py: funciones para realizar *POS tagging*.
   b) ner-spanish.py: funciones para realizar reconocimiento de nombres de entidades desde textos en Español.
2. Baje un texto en Español (ej. *noticia*) desde Internet a su directorio de trabajo.
3. Ajuste el nombre del directorio de entrada (PATH) en (a) y realice *POS tagging* de su documento.
4. Ajuste el nombre del directorio de entrada (PATH) en (b) y realice *NER* desde su documento con las funciones dadas.
5. ¿Qué problemas se observan? ¿Soluciones?