Introducción al Procesamiento de Lenguaje Natural

Grupo de PLN - InCo

Análisis Léxico

Análisis Léxico

Objetivo: dado un texto, atribuir a cada constituyente su categoría gramatical.

Por ejemplo:

```
El/DET gato/NOM come/VB pescado/NOM ./SP
```

¿Cómo podemos automatizarlo? ¿Problemas?

Análisis Léxico

¿Y si queremos más información?

El	gato	come	pescado	.
el	*gato*	*comer*	*pescado*	.
DA0MS0	NCMS000	VMIP3S0	NCMS000	Fp
1	1	0.75	0.833333	1
		comer	*pescar*	
		VMM02S0	VMP00SM	
		0.25	0.166667	

Análisis Léxico

Algunos ejemplos de ambigüedad:

Bajo con el hombre bajo a tocar el bajo bajo la escalera.

El señor Mesa se mesa la barba al lado de la mesa.

Dejá en la barra la barra de pan, para que el mozo barra.

Ana y Laura están leyendo a Herrera y Reissig.

Análisis Léxico

- Motivaciones
  - Pronunciación. Inglés:
    - DIScount (nombre) vs. disCOUNT (verbo)
    - OBject (nombre) vs. obJECT (verbo)
  - Lematización (stemming) para R.I.
    - Búsqueda del verbo *cantar*.
  - Resumen Automático
    - Pesos a constituyentes textuales según categoría gramatical.

Análisis Léxico

Motivaciones

Estudios de fenómenos lingüísticos:

<verbo> + <prep>

<nomprop> “y” <nomprop>

Ejercicio:

buscar formas del verbo servir + <prep> en

http://www.corpusdelespanol.org/

sólo textos periodísticos, siglo XX

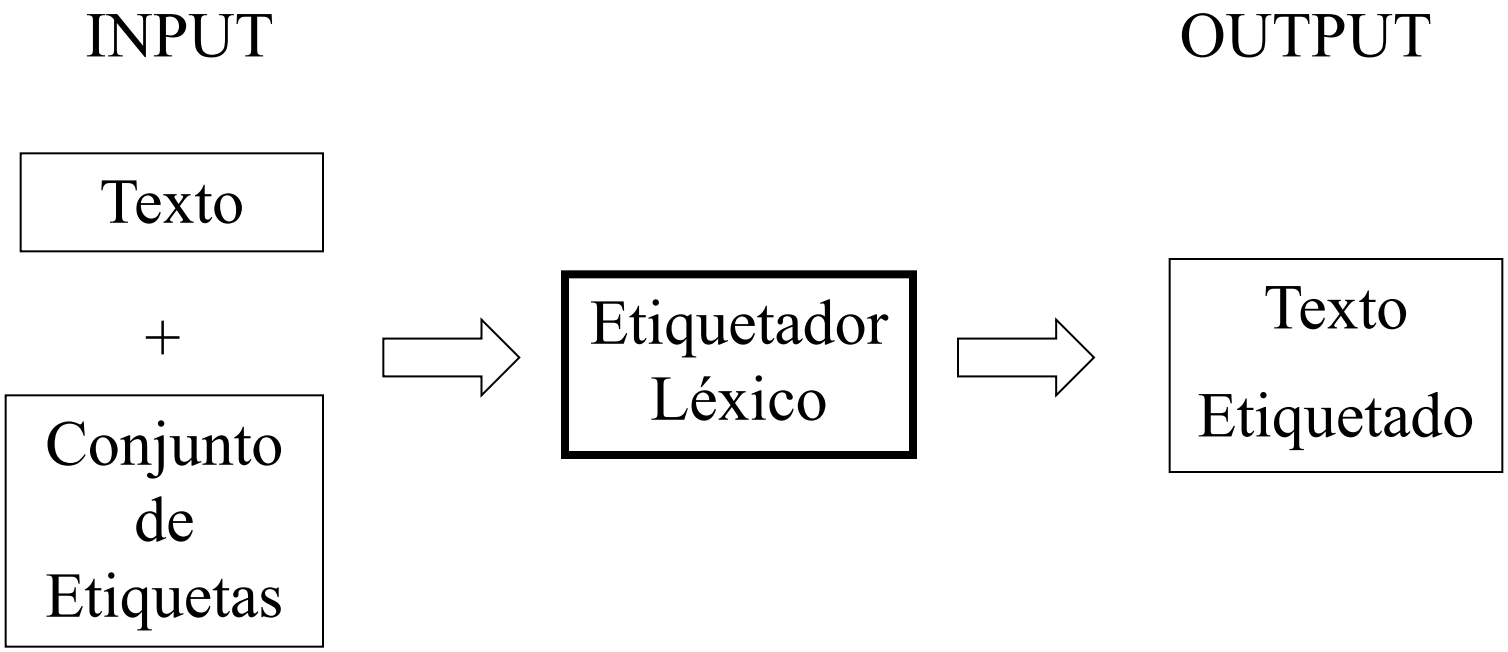
Análisis Léxico

Resultados (verificar)

- 387 ocurrencias de "servir de" (La parte inferior de la estructura servía de tren de aterrizaje.)
- 354 ocurrencias de "servir para" (La vidriera servía para transformar la luz natural exterior en una luz "no natural" diferenciada, ...)
- 74 ocurrencias de "servir a"
- 31 ocurrencias de "servir en"
- 9 ocurrencias de "servir con"

Motivación: Allanar camino para realizar un análisis sintáctico.

Etiquetado léxico



Categorías gramaticales

determinante (*el*, *la*, *un*, *unos*, *esos*, *nuestro*, *algún*)
preposición (*a*, *de*, *por*, *contra*)
pronombre (*yo*, *él*, *mí*, *se*, *aquello*, *donde*, *que*)
conjunción (*y*, *o*, *si*, *que*)
nombre (*casa*, *casas*, *felicidad*)
verbo (*cantar*, *cantando*, *cantaremos*, *sabe*, *sé*)
adjetivo (*alto*, *alta*, *lindos*, *inteligente*, *solo*)
adverbio (*medio*, *simplemente*, *sólo*)

Categorías Eagle

1. Adjetivos
2. Adverbios
3. Determinantes
4. Nombres
5. Verbos
6. Pronombres
7. Conjunciones
8. Preposiciones
9. Interjecciones (ah, ejem, eh…)
10. Signos De Puntuación (*Preprocesamiento*)
11. Cifras (222, cuarenta y ocho, cinco pesos…)
12. Fechas Y Horas

Categorías Eagle

Para cada categoría define atributos y valores posibles.

Lematiza y etiqueta combinando atributos.

NOMBRES			
Pos.	**Atributo**	**Valor**	**Código**
1	Categoría	Nombre	N
2	Tipo	Común	C
		Propio	P
3	Género	Masculino	M
		Femenino	F
		Común	C
4	Número	Singular	S
		Plural	P
		Invariable	N
5-6	Clasificación semántica	-	0
7	Grado	Apreciativo	A

Categorías Brown Corpus

BE	verb "to be", infinitive or imperative	be
BED	verb "to be", past tense, 2nd person singular or all persons plural	were
BED*	verb "to be", past tense, 2nd person singular or all persons plural, negated	weren't
FW-JJ	foreign word: adjective	serieuses royaux anticus Sovietskaya
VBN	verb, past participle	conducted charged won
VBZ	verb, present tense , 3rd person singular	deserves believes receives takes

Universal POS Tags

- Parte del proyecto "Universal Dependencies" de colaboración abierta
- Busca tener un formato consistente a través de diferentes idiomas para anotación de Treebanks
- Basado en las dependencias de Stanford y los Google Universal POS-tags

Universal POS Tags

- Etiquetas de POS (POS-tags) universales
  - Abiertas (ADJ, ADV, NOUN, etc), Cerradas (ADP, AUX, etc), Otras (PUNCT, SYM). Detalles

- Características universales: Género, Número, Forma Verbal, etc. Detalles

- Características específicas para algunos idiomas
- Formas de conversión

Análisis Léxico

- Dos categorías principales:
  - Enfoques por reglas:
    - Grandes bases de conocimiento con reglas de desambiguación
  - Enfoques estadísticos:
    - Entrenamiento de modelos a partir de grandes corpus
    - Estimación de la probabilidad de que una palabra tenga determinada categoría gramatical dado un contexto.
- Los enfoques híbridos son plausibles
  - P.ej el etiquetador basado en transformaciones (Brill)

Análisis Léxico

Primeros intentos en los años 60.

Arquitectura en dos capas:

- Capa 1: asigna categorías potenciales a partir de un diccionario.
- Capa 2: utiliza reglas de desambiguación para eliminar categorías candidatas.

Los enfoques modernos utilizan esta arquitectura, pero poseen recursos lingüísticos más completos.

EngCG tagger

Lexicon basado en morfología a dos niveles.
Aproximadamente 56000 entradas.
Cada entrada se anota con propiedades morfológicas y sintácticas.
Supera el 99,5% de correctitud.

Word	POS	Additional POS features
smaller	ADJ	COMPARATIVE
entire	ADJ	ABSOLUTE ATTRIBUTIVE
fast	ADV	SUPERLATIVE
that	DET	CENTRAL DEMONSTRATIVE SG
all	DET	PREDETERMINER SG/PL QUANTIFIER
dog's	N	GENITIVE SG
furniture	N	NOMINATIVE SG NOINDEFDETERMINER
one-third	NUM	SG
she	PRON	PERSONAL FEMININE NOMINATIVE SG3
show	V	PRESENT -SG3 VFIN
show	N	NOMINATIVE SG
shown	PCP2	SVOO SVO SV
occurred	PCP2	SV
occurred	V	PAST VFIN SV

EngCG tagger

Ejemplo: Pavlov had shown that salivation...

Pavlov	**PAVLOV N NOM SG PROPER**
had	**HAVE V PAST VFIN SVO**
	HAVE PCP2 SVO
shown	**SHOW PCP2 SVOO SVO SV**
that	ADV
	PRON DEM SG
	DET CENTRAL DEM SG
	CS
salivation	**N NOM SG**
...	

EngCG tagger

Se aplican 3744 reglas para quitar las etiquetas incorrectas.

P.ej: regla para “that” como adverbio.

ADVERBIAL-THAT RULE
Given input: “that”
if
(+1 A/ADV/QUANT); /* *if next word is adj, adverb, or quantifier* */
(+2 SENT-LIM); /* *and following which is a sentence boundary,* */
(NOT -1 SVOC/A); /* *and the previous word is not a verb like* */
/* *‘consider’ which allows adjs as object complements* */
then eliminate non-ADV tags
else eliminate ADV tag

POS-tagging con Hidden Markov Models

Uso de modelos markovianos ocultos (caso particular de la inferencia bayesiana).

Clasificación: dada una observación, se intenta determinar a cuál conjunto de clases pertenece.

El POS-Tagging se puede concebir como una tarea de clasificación de secuencias:

- Dada una secuencia de palabras, se debe inferir la secuencia apropiada de categorías gramaticales.

POS-tagging con HMM

Ejemplo:

"Secretariat/NNP is/BEZ expected/VBN to/TO **race/{VB,NN}** tomorrow/NR"

Idea:

considerar todas las posibles secuencias de etiquetas

elegir la secuencia de etiquetas que es la más probable para la secuencia de observación de n palabras: w_1, w_2,...w_n

POS-tagging con HMM

Es decir: de todas las secuencias de n etiquetas $t_1, t_2, \ldots, t_n$, queremos aquella tal que $P(t_1 \ldots t_n \mid w_1 \ldots w_2)$ es máxima.

$$\hat{t}_1^n = \arg\max_{t_1^n} P(t_1^n | w_1^n)$$

El ^ significa "nuestro estimado del mejor"

argmax_x f(x) significa "el x tal que f(x) es maximizada"

POS-tagging con HMM

- ¿Cómo computar la fórmula?

$$\hat{t}_1^n = \arg\max_{t_1^n} P(t_1^n | w_1^n)$$

- Usando la regla de Bayes para transformar las probabilidades en otras más simples de computar

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

POS-tagging con HMM

$$\hat{t}_1^n = \arg\max_{t_1^n} \frac{P(w_1^n|t_1^n)P(t_1^n)}{P(w_1^n)}$$

$$\hat{t}_1^n = \arg\max_{t_1^n} P(w_1^n|t_1^n)P(t_1^n)$$

POS-tagging con HMM

La probabilidad de que una palabra ocurra depende sólo de su etiqueta (y no de otras palabras y etiquetas "alrededor")

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

La probabilidad de que una etiqueta ocurra depende sólo de la etiqueta previa (hipótesis de bigrama).

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

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

POS-tagging con HMM

Probabilidad de la transición $P(t_i \mid t_{i-1})$:

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

Los determinantes suelen preceder adjetivos y nombres. Entonces: P(NN | DT) y P(JJ | DT) deberían ser altas

Ejemplo a partir del Brown Corpus.

$$P(NN|DT) = \frac{C(DT, NN)}{C(DT)} = \frac{56,509}{116,454} = 0.49$$

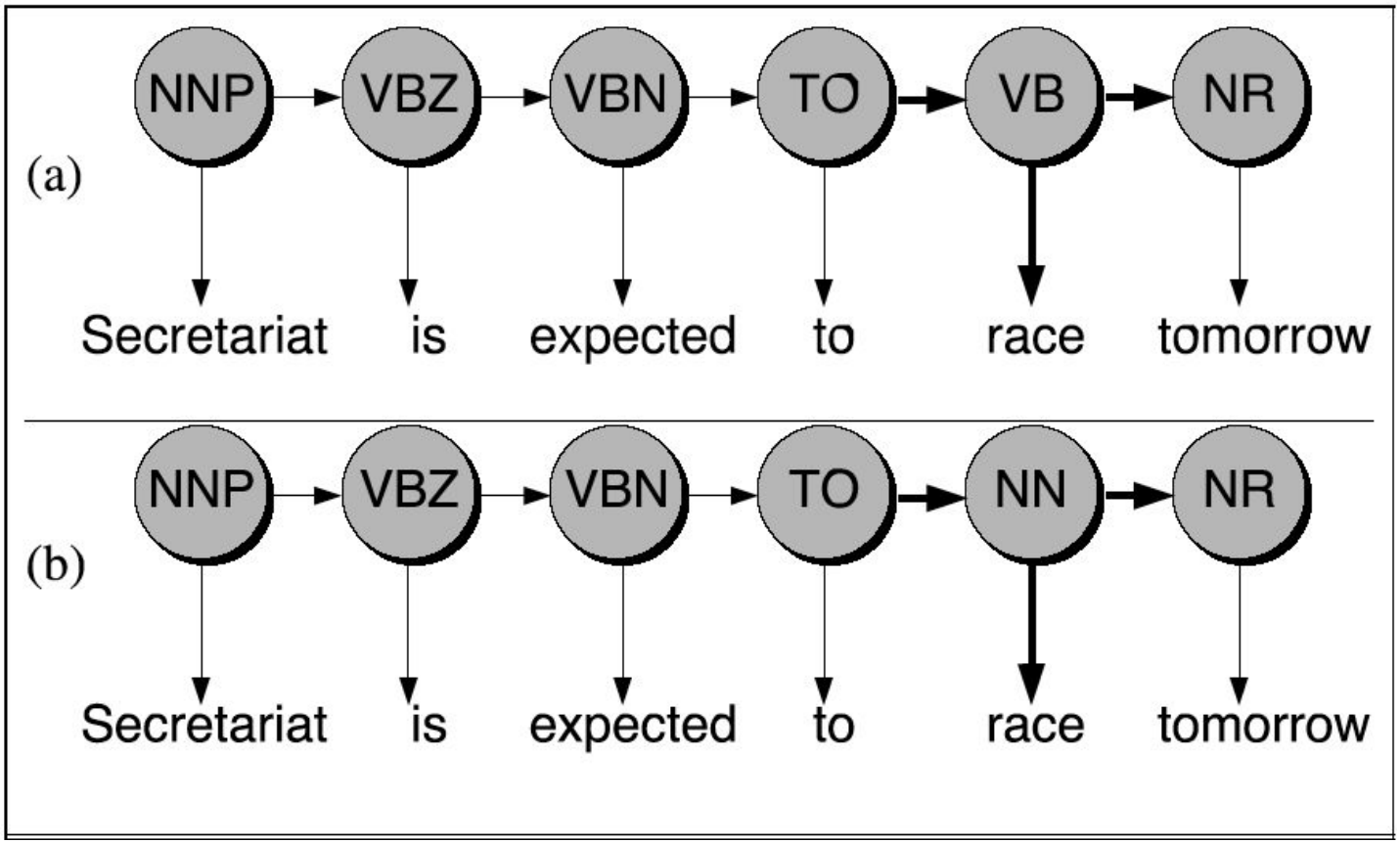
POS-tagging con HMM

Probabilidad de emisión de la palabra $P(w_i|t_i)$:

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

$$P(is|VBZ) = \frac{C(VBZ, is)}{C(VBZ)} = \frac{10,073}{21,627} = 0.47$$

POS-tagging con HMM



POS-tagging con HMM

Desambiguar “race”

Datos tomados del Brown Corpus

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

POS-tagging con HMM

Opción 1: “race” funciona como verbo

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

Opción 2: “race” funciona como nombre

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

Hidden Markov Models

$Q = q_1 q_2 q_3 \ldots q_n$ {conjunto de estados}

$A = a_{o1} a_{o2} \ldots a_{n1} \ldots a_{nn}$ { probs de transición}

$O = o_1 o_2 \ldots o_N$ {conjunto de observaciones}

$B = b_i(o_t)$ {probabilidades de emisión}

$q_o q_{end}$ {estados inicial y final}

Hidden Markov Models

Problemas fundamentales (Rabiner, 1989)

- **Decodificar**: dado un HMM λ y una secuencia O, calcular la mejor secuencia Q de estados ocultos
- **Calcular verosimilitud**: dado un HMM λ y una secuencia O, calcular P(O| λ)
- **Aprender**: dado O y los estados de un HMM, aprender los parámetros A y B del HMM

Hidden Markov Models

Algoritmo de Viterbi: permite determinar cuáles son las variables asociadas a una secuencia de observaciones (decodificar).

Es el algoritmo más usado en HMM.

Algoritmo de programación dinámica.

Idea: computar la secuencia que maximiza la probabilidad de estar en el estado j después de t observaciones.

Algoritmo de Viterbi

El algoritmo llena una tabla $T_{i,j}$ con probabilidades hasta la palabra *i* en el estado *j*

- Comienza el estado inicial con probabilidad 1
- Para *i* entre 1 y *n*
  - Para cada estado *j*

    $T_{i,j} = \max_k(T_{i-1,k} * a_{kj} * b_j(w_i))$

    (recordar además el *k* del máximo)
- Devolver la secuencia de *k*s máximos

Ejemplo

fruit	flies	fast
NN	NNS	VB
	VB	RB
		JJ

$b_j(w)$

$b_2(\text{fruit}) = P(\text{fruit}|\text{NN}) = 0.1$

$b_3(\text{flies}) = P(\text{flies}|\text{NNS}) = 0.01$

$b_4(\text{flies}) = P(\text{flies}|\text{VB}) = 0.1$

$b_4(\text{fast}) = P(\text{fast}|\text{VB}) = 0.01$

$b_5(\text{fast}) = P(\text{fast}|\text{RB}) = 0.3$

$b_1(\text{fast}) = P(\text{fast}|\text{JJ}) = 0.05$

$a_{ij} = P(t_j|t_i)$

	j	0	1	2	3	4	5
i		</s>	JJ	NN	NNS	VB	RB
0	<s>	0	0.3	0.2	0.2	0.2	0.1
1	JJ	0.2	0.1	0.3	0.2	0.1	0.1
2	NN	0.2	0.1	0.2	0.2	0.2	0.1
3	NNS	0.2	0.1	0.1	0.2	0.3	0.1
4	VB	0.2	0.1	0.2	0.2	0	0.3
5	RB	0.2	0.1	0.2	0.1	0.2	0.2

Hidden Markov Models

Algoritmo forward: permite calcular la probabilidad de estar en un estado j después de t observaciones (verosimilitud)

Idea: igual que Viterbi, pero en vez de obtener la probabilidad máxima, sumamos en todos los caminos posibles.

Hidden Markov Models

Y si no tenemos un corpus etiquetado?

Algoritmo Baum-Welch o forward-backward: permite entrenar un HMM, calculando las probabilidades de transición y de emisión

Idea: Expectation-Maximization, aprendizaje no supervisado

Tagger de Brill

Enfoque híbrido: se usan reglas pero se infieren a partir de grandes corpus.

Se utilizan patrones de reglas para limitar el conjunto de tipos de reglas a inferir.

El algoritmo etiqueta inicialmente según etiquetas más probables y luego hace un ciclo en el que examina cada posible transformación (regla) y selecciona la que maximiza la probabilidad, hasta que la mejora no sea significativa.

Misceláneas

Palabras desconocidas: ¿qué hacer?

- suponer que es ambigua en relación equiprobable con todas las etiquetas.
- usar morfología (-s, -ando, etc.)

Otros idiomas

Combinación de taggers

- combinar salidas:
  - votación
  - entrenar un clasificador que elija en cuál tagger confiar según el contexto

Referencias

J.Martin & D.Jurafsky. Speech and Language Processing. Capítulos 8 y 9 (Tercera Edición)

Apuntes del Curso “Biological and Linguistic Sequence Analysis” - Brian Roark (Facultad de Ingeniería, 2008)