

# Índice

- 1 Conceptos de PLN: Análisis Morfológico y Etiquetación
- 2 Conceptos de PLN: Análisis Sintáctico Superficial
- 3 Conceptos de PLN: Semántica Léxica
- 4 Extracción de Información
- 5 Búsqueda de Respuestas

# Procesamiento del Lenguaje Natural (PLN)

- a.k.a. *Natural Language Processing (NLP)*
- **Def.:** tratamiento computacional del lenguaje humano
  - **Objetivo:** computadora comprenda el lenguaje humano
- **Aplicaciones:**
  1. Procesamiento de texto escrito:
    - Ayudas a la producción de texto: correctores ortográficos y gramaticales (por ej. de estilo) y OCR.
    - Traducción automática de textos.
    - Extracción de información.
    - Generación automática de resúmenes.
    - Clasificación, recuperación y filtrado de documentos
  2. Interacción Hombre-Máquina:
    - Interfaces en lenguaje natural: BBDD, aplicaciones educativas, etc.
    - Reconocimiento y síntesis de voz: servicios de atención al cliente, control de máquinas por la voz, interfaces para discapacitados.

# Niveles de Análisis

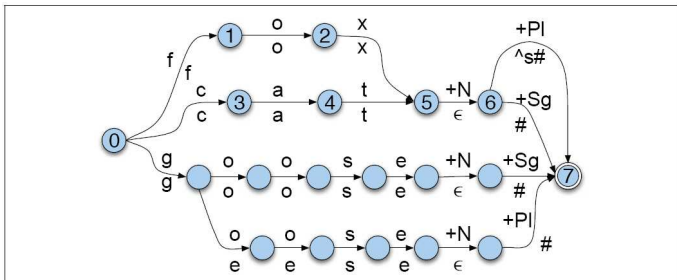
1. **Morfológico:** determinar las palabras que integran un texto, su estructura morfémica y su etiqueta morfosintáctica
2. **Sintáctico:** cómo se combinan las palabras en *sintagmas* y *frases*
3. **Semántico:** determinar el *significado* de cada palabra y cómo se construye el significado de una frase a partir de los significados de las palabras que la constituyen
4. **Pragmático/de discurso:** cómo se relaciona el lenguaje con su *contexto* de uso. Ej.: establece la identidad de las personas y objetos que aparecen en los textos, gestiona el diálogo en un entorno conversacional, etc.

# Análisis Morfológico

- **Def.:** dada la *forma* de una palabra (considerada de forma aislada), obtener los diferentes **rasgos morfológicos** asociados a ella: categoría gramatical, género, número, persona, etc.

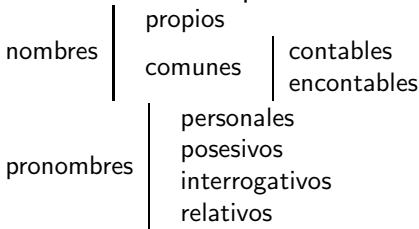
*cats* → cat +N +PL

- Por lo general implementados mediante **traductores de estado finito** (*finite-state transducers, FST*):



# Clases de Palabras

- Las palabras pueden clasificarse en **clases**/categorías (*Part of Speech*, *PoS*) en base a su función dentro de las frases del lenguaje.
- Clases de palabras:
  - Clases cerradas: número fijo de palabras (no se pueden añadir más).  
Ej: preposiciones, pronombres, conjunciones...
  - Clases abiertas: permiten añadir nuevas palabras (mediante flexión derivación, inclusión de neologismos...)  
Ej: nombres, adjetivos, verbos, adverbios...
- Dentro de una clase pueden existir **subclases**:



# Tagsets

- **Def.:** conjuntos de **etiquetas** (*PoS tags*) posibles.
- Una etiqueta codifica:
  - Categoría léxica (clase de palabra).
  - Información de los rasgos gramaticales (género, número...).
- La información codificada (y por tanto su complejidad) depende del lenguaje y del tipo de aplicación.
  - Tagsets pequeños: decenas de etiquetas
  - Tagsets grandes: cientos

# Tagsets (cont.): Penn Treebank

Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	<i>and, but, or</i>	SYM	symbol	<i>+, %, &amp;</i>
CD	cardinal number	<i>one, two, three</i>	TO	“to”	<i>to</i>
DT	determiner	<i>a, the</i>	UH	interjection	<i>ah, oops</i>
EX	existential ‘there’	<i>there</i>	VB	verb, base form	<i>eat</i>
FW	foreign word	<i>mea culpa</i>	VBD	verb, past tense	<i>ate</i>
IN	preposition/sub-conj	<i>of, in, by</i>	VBG	verb, gerund	<i>eating</i>
JJ	adjective	<i>yellow</i>	VBN	verb, past participle	<i>eaten</i>
JJR	adj., comparative	<i>bigger</i>	VBP	verb, non-3sg pres	<i>eat</i>
JJS	adj., superlative	<i>wildest</i>	VBZ	verb, 3sg pres	<i>eats</i>
LS	list item marker	<i>1, 2, One</i>	WDT	wh-determiner	<i>which, that</i>
MD	modal	<i>can, should</i>	WP	wh-pronoun	<i>what, who</i>
NN	noun, sing. or mass	<i>llama</i>	WP\$	possessive wh-	<i>whose</i>
NNS	noun, plural	<i>llamas</i>	WRB	wh-adverb	<i>how, where</i>
NNP	proper noun, singular	<i>IBM</i>	\$	dollar sign	<i>\$</i>
NNPS	proper noun, plural	<i>Carolinas</i>	#	pound sign	<i>#</i>
PDT	predeterminer	<i>all, both</i>	“	left quote	<i>‘ or “</i>
POS	possessive ending	<i>'s</i>	”	right quote	<i>’ or ”</i>
PRP	personal pronoun	<i>I, you, he</i>	(	left parenthesis	<i>[, (, {, &lt;</i>
PRP\$	possessive pronoun	<i>your, one's</i>	)	right parenthesis	<i>], ), }, &gt;</i>
RB	adverb	<i>quickly, never</i>	,	comma	<i>,</i>
RBR	adverb, comparative	<i>faster</i>	.	sentence-final punc	<i>. ! ?</i>
RBS	adverb, superlative	<i>fastest</i>	:	mid-sentence punc	<i>: ; ... --</i>
RP	particle	<i>up, off</i>			

# Etiquetación

- a.k.a. *PoS tagging*
- **Def.:** proceso de asignar etiquetas a las palabras de un texto

$tager f : W \longrightarrow T = f(W)$

$W = w_1 w_2 \dots w_n$  cadena de palabras (i.e. texto a etiquetar)

$T = t_1 t_2 \dots t_n$  cadena de etiquetas

Yo	bajo	con	el	hombre	bajo	a	tocar	el	bajo	bajo	la	escalera
PPers	Verb	Prep	Det	NCom	Adj	Prep	Verb	Det	NCom	Prep	Det	NCom

- **Aplicaciones:**

- Frecuentemente fase previa en tareas de PLN.
- Reconocimiento de voz (homófonas): *te* vs. *té*.
- Generación de voz (pronunciación): object (N) vs. object (V).
- Fase previa al análisis sintáctico.
- *Text Mining*: ponderar palabras con contenido, encontrar nombres, entidades, relaciones, etc.
- Traducción automática: *object* (N)  $\rightarrow$  *objeto* vs. *object* (V)  $\rightarrow$  *objetar*.

# Etiquetación (cont.)

- **Problema:** una palabra aislada puede tener varias etiquetas candidatas (**ambigüedad**):

Yo	bajo	con	el	hombre	bajo	a	tocar	el	bajo	bajo	la	escalera
PPers	<b>Verb</b>	Prep	Det	NCom	Verb	Prep	Verb	Det	Verb	Verb	Det	NCom
	Adj				<b>Adj</b>				Adj	Adj		
	Ncom				NCom				<b>NCom</b>	NCom		
	Prep				Prep				Prep	<b>Prep</b>		

# Etiquetación (cont.)

- **Solución: desambiguación morfosintáctica:** asigna a cada palabra la categoría léxica más apropiada en función del contexto local
  - El contexto da información acerca de la categoría concreta de esa palabra en ese momento
    - Ej. es más probable que la palabra que siga a un artículo sea un sustantivo que un verbo: *un coche* vs. *un comíamos\**
  - La mayoría de las palabras no son ambiguas y nos proporcionan un contexto fijo sobre el que aplicar algoritmos.
- Opcionalmente también **lematización**
  - **Lema:** forma canónica de la palabra (i.e. su entrada del diccionario).  
Ej.: *rojas*→*rojo*; *cantábamos*→*cantar*
- **Tipos** de etiquetadores-desambiguadores:
  - Basados en reglas.
  - Estocásticos.
  - Híbridos.
  - Otros paradigmas: modelos de máxima entropía, árboles de decisión, RNA, SVM, algoritmos evolutivos, etc.

# Recursos Lingüísticos para Etiquetación

- **Texto/corpus etiquetado.** Todas sus palabras aparecen con su etiqueta correcta (opcionalmente también su lema).
  - Como *corpus de entrenamiento*

la	DAFS	el
selecci'on	NCFS	selecci'on
femenina	AQFS	femenino
afronta	V3SRI	afrontar
esta	DDFS	este
tarde	NCFS	tarde

- **Diccionario (a.k.a. lexicón).** Una lista de palabras acompañada de sus posibles etiquetas (opcionalmente también sus lemas).

arom'atica	AQ+FS	arom'atico
arom'aticas	AQ+FP	arom'atico
arom'atico	AQ+MS	arom'atico
arom'aticos	AQ+MP	arom'atico

# Etiquetadores Basados en Reglas

## ● **Arquitectura de 2 niveles**

1. Se asigna a cada palabra un conjunto de etiquetas candidatas (diccionario o analizador morfológico).
2. Se usan listas de reglas de desambiguación para reducir el número de etiquetas por palabra a 1.

## ● **Reglas construidas manualmente** por un experto (lingüista).

- Conocimiento lingüístico (*Knowledge-driven taggers*).
- Número limitado de reglas (< 1000).

## ● **Ventajas:**

- Reglas motivadas lingüísticamente.
- Alta precisión ( $\equiv$  99%).

## ● **Inconvenientes:**

- Alto coste de desarrollo.
- No portable (i.e. sólo sirve para el idioma para el que se creó).
- Mayor coste de etiquetación.

# Ejemplo: EngCG ENGTWOL

- **Paso 1:** obtención de las etiquetas candidatas mediante un analizador morfológico

Pavlov	PAVLOV N NOM SG PROPER
had	HAVE V PAST VFIN SVO HAVE PCP2 SVO
shown	SHOW PCP2 SV00 SVO SV
that	ADV PRON DEM SG DET CENTRAL DEM SG CS
salivation	N NOM SG

## Ejemplo: EngCG ENGTWOL (cont.)

- **Paso 2:** aplicación de reglas de restricción para eliminar las etiquetas incorrectas:

Regla THAT-ADVERBIAL

Entrada: *"that"*

If

(+1 A/ADV/CUANT); Si la siguiente palabra es un ADJ, ADV o cuantificador...

(+2 SENT-LIM); ... seguida por un End-of-Sentence

(NOT -1 SVOC/A); ... y la palabra anterior no es un verbo como "consider"/"believe" que permiten adjetivos como objetos complementos (se utiliza para evitar considerar "that" como adverbio en frases como:

"I consider that odd" )

then eliminate non-ADV tags

else eliminate ADV tag

# Etiquetadores Estocásticos

- Calculan la **probabilidad** de que una determinada palabra tenga una determinada etiqueta en un contexto dado.
- **Modelo generado automáticamente** a partir de un *corpus de entrenamiento*.
  - Probabilidades estimadas a partir de los datos (*Data-Driven Taggers*).
- Ventajas:
  - Marco teórico bien fundamentado.
  - Aproximación clara y métodos simples.
  - Precisión aceptable ( $> 97\%$ ).
  - Rapidez.
  - Portabilidad (basta volver a entrenar).

# Etiquetadores Estocásticos (cont.)

- Inconvenientes:
  - Aprendizaje del modelo (i.e. necesitamos *corpus de entrenamiento*).
    - Preciso sean similares a los textos a etiquetar (dependencia).
  - Conocimiento no explícito: implícito en los parámetros estimados.
    - No se pueden "leer" ni modificar.
  - Menor precisión que los etiquetadores basados en reglas.
  - Contextos pequeños.

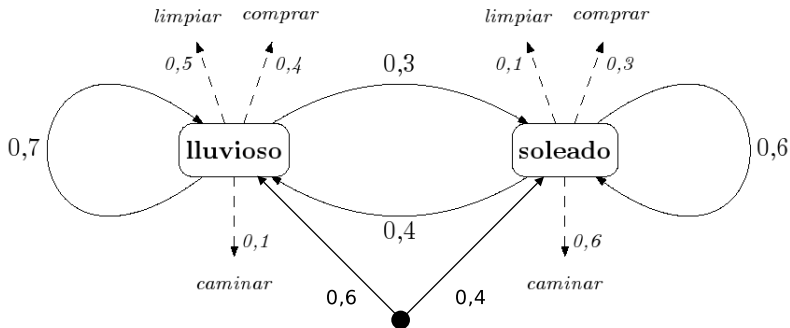
# Modelos de Markov Ocultos (HMM)

- a.k.a. *Hidden Markov Models (HMM)*
- Herramienta estadística de propósito general originalmente para modelizar secuencias de palabras en ruso (*A. Markov, 1913*)
- Sistema modelizado como un conjunto finito de estados donde, pasado un intervalo de tiempo, el sistema cambia de estado de acuerdo a unas probabilidades de transiciones entre estados:
  - Modelo del lenguaje: representado por un autómata finito probabilista (los estados del autómata se asocian a las etiquetas).
  - Modelo de comunicación: representado por la probabilidad de "emisión" de una palabra en un estado dado (la probabilidad de la palabra depende sólo de la etiqueta).
- **Etiquetación:** proceso doblemente aleatorio parametrizable
- **Objetivo:** calcular la secuencia de etiquetas **más probable** para el texto de entrada

# Ejemplo HMM

- Queremos saber qué tiempo ha hecho en Lugo (estados *lluvioso* y *soleado*).
- Allí vive un amigo que nos cuenta lo que ha hecho cada día (*caminar*, *comprar* o *limpiar*), pero nunca nada acerca del tiempo.
- Sin embargo sabemos que, dependiendo sólo del tiempo que hace ese día, existe una determinada probabilidad de que nuestro amigo realice alguna de esas tres actividades.
- Con los datos de los últimos meses (*corpus de entrenamiento*) que nos proporciona el pronóstico meteorológico (tiempo que ha hecho cada día) y nuestro amigo (qué actividades ha hecho cada día) podemos construir un HMM para determinar el estado del tiempo en su ciudad en función de las actividades que realiza.

## Ejemplo HMM (cont.)

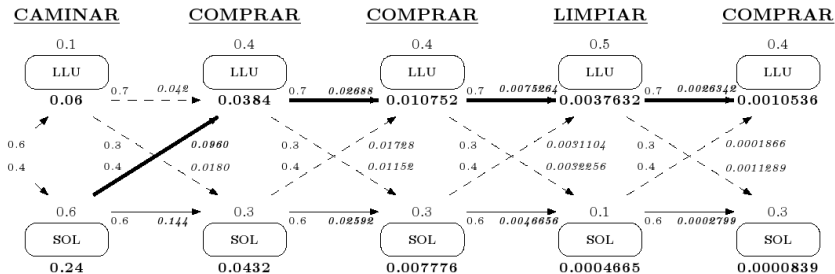


## Ejemplo HMM (cont.)

- Si nuestro amigo nos ha dicho que esta semana ha realizado las siguientes actividades:

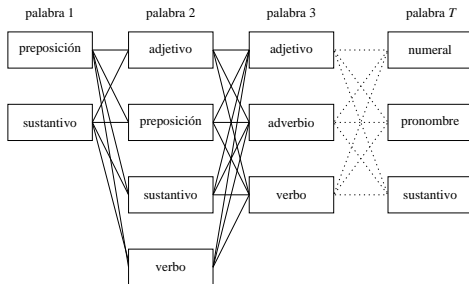
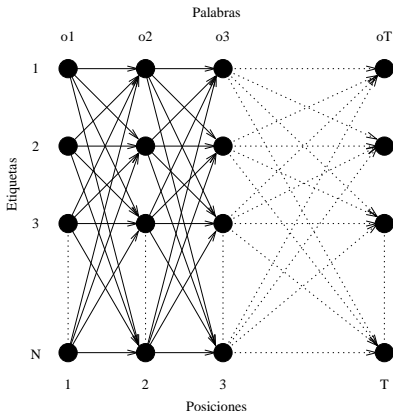
Lunes	Martes	Miércoles	Jueves	Viernes
caminar	comprar	comprar	limpiar	caminar

¿qué tiempo ha hecho esos días (con mayor probabilidad)?



= (sol, llu, llu, llu, llu)

# HMM: Adaptación a Etiquetación



## Parámetros del modelo a estimar:

1. Probabilidad de estado inicial.
2. Probabilidades de las transiciones.
3. Probabilidad de emisión de una palabra.

# Etiquetadores Híbridos: Etiquetador de Brill

- Comparte características con los etiquetadores estadísticos y los basados en reglas.
- Etiquetación **mediante reglas**.
- Reglas **generadas automáticamente** a partir de *corpus de entrenamiento*.
  - Técnicas de aprendizaje supervisado: Aprendizaje Basado en Transformaciones (*Transformation Based Learning*).
- **Información lingüística explícita** en forma de reglas simples:
  - Permiten interdependencias complejas entre palabras y etiquetas.
  - Permiten contextos más amplios que los etiq. estadísticos.

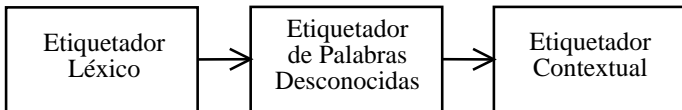
# Etiquetadores Híbridos: Etiquetador de Brill (cont.)

- Ventajas:
  - Evita la generación manual de reglas (lento y costoso).
  - Conocimiento explicitado: posible corregir/añadir reglas manualmente.
  - Portabilidad (basta volver a entrenar).
  - Detecta gran variedad de regularidades léxicas y sintácticas.
- Inconvenientes:
  - Más lento que etiq. estocásticos para entrenar y etiquetar.

# Etiquetador de Brill: Arquitectura

Tres módulos, con inferencia automática a partir del *corpus*:

1. Etiquetador Léxico.
2. Etiquetador de Palabras Desconocidas.
3. Etiquetador Contextual.



# Etiquetador de Brill: Arquitectura (cont.)

## 1. Etiquetador Léxico

- Etiqueta inicialmente cada palabra con la etiqueta más probable (estimadas a partir del *corpus*).

## 2. Etiquetador de Palabras Desconocidas

- Intenta etiquetarlas en base a su prefijo, sufijo, etc. y/o su contexto.
- *Regla = descripción de contexto + regla de reescritura* (reemplazo etiquetas).
- Reglas (18 plantillas):

A X fhassuf 1 B      Si la etiqueta actual es A y los 1 últimos caracteres son X, reemplazar la etiqueta por B. Ejemplo:

r hassuf 1 V000f0      Si acaba en "r", se etiqueta como verbo infinitivo.

# Etiquetador de Brill: Arquitectura (cont.)

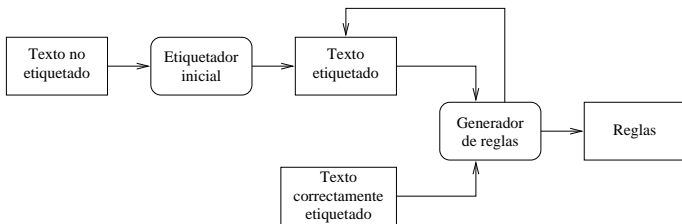
## 3. Etiquetador Contextual

- Aplica **en orden** una serie de reglas contextuales inferidas a partir del corpus de entrenamiento.
- *Regla = descripción de contexto + regla de reescritura*
  - Descripción del contexto: de etiquetas y/o reglas
  - 9 tipos de entornos y 26 plantillas de reglas:

A B prevtag C                      cambia A por B si la anterior palabra está etiquetada con C.

Afp0 Amp0 prevtag scmp            Cambia de Afp0 (adjetivo femenino plural) a Amp0 (adjetivo masculino plural) si la etiqueta anterior es scmp (sustantivo común masculino plural).

# Etiquetador de Brill: Aprendizaje



Algoritmo basado en transformaciones y dirigido por el error:

1. Etiquetación inicial: etiqueta más probable.
  2. Comparar con texto etiquetado manualmente + identificar errores.
  3. Probar cada regla de transformación posible y seleccionar la que elimine más errores.
  4. Se aplica dicha regla y volvemos a 2.
- Condición de parada: el error cae por debajo de un umbral.

# Recursos en la Web

## ● Demos on-line:

- Freeling (multilingüe, gallego y castellano incl.): Centro de Tecnologías y Aplicaciones del Lenguaje y del Habla, Univ. Politécnica de Cataluña: <http://garraf.epsevg.upc.es/freeling/demo.php>
- Cognitive Computation Group, Univ. of Illinois at Urbana-Champaign: [http://l2r.cs.uiuc.edu/~cogcomp/pos\\_demo.php](http://l2r.cs.uiuc.edu/~cogcomp/pos_demo.php)
- CST's PoS tagger (Brill adaptado): Centre for Language Technology, Univ. of Copenhagen: [http://cst.dk/online/pos\\_tagger/uk/index.html](http://cst.dk/online/pos_tagger/uk/index.html)

## ● Descargas:

- Freeling (multilingüe, gallego y castellano incl.):  
<http://www.lsi.upc.edu/~nlp/freeling/>
- CST's PoS tagger (Brill adaptado: inglés y lenguas nórdicas):  
<http://cst.dk/download/uk/index.html>
- TnT (estocástico: inglés y alemán):  
<http://www.coli.uni-saarland.de/~thorsten/tnt/>
- TreeTagger (multilingüe, español incl.):  
<http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html>

# Referencias

- [Jurafsky & Martin, 2009a] Jurafsky, D. & Martin, J.H. (2009). Chapter 3: Words and Transducers. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (2nd ed.)*. Pearson–Prentice Hall.
- [Jurafsky & Martin, 2009b] Jurafsky, D. & Martin, J.H. (2009). Chapter 5: Part-of-Speech Tagging. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (2nd ed.)*. Pearson–Prentice Hall.
- [Graña, 2000] Graña, J. (2000). *Técnicas de Análisis Sintáctico Robusto para la Etiquetación del Lenguaje Natural*, PhD. Thesis, Detpo. de Computación, Universidade da Coruña. Disponible en <http://www.dc.fi.udc.es/~grana/> (visitada en marzo de 2010).