

Índice

- 1 Gramáticas de Unificación
- 2 Análisis Sintáctico Superficial**
- 3 Representación y Análisis Semántico
- 4 Semántica Léxica
- 5 Recuperación de Información
- 6 Extracción de Información
- 7 Búsqueda de Respuestas
- 8 Traducción Automática

Introducción

- Algunas tareas de PLN requieren **información sintáctica** del texto de entrada.
- Problemas del **análisis sintáctico completo/clásico** (*full parsing*):
 - Requiere conocimiento/recursos lingüísticos complejos (gramáticas, *treebanks*)
 - Escasa cobertura de las gramáticas
 - Escasa robustez
 - Alto coste
- Sin embargo no siempre es necesario que dicha información sea completa/exhaustiva pues sólo nos interesan ciertas estructuras o relaciones. P.ej.:
 - IR: más centrado en frases nominales
 - IE: sólo los segmentos de texto con información relevante

Introducción (cont.)

- Solución: **análisis sintáctico superficial** (*shallow parsing*; a.k.a. **chunking**, *partial parsing*):
 - Devuelve una representación "*superficial*" (i.e. aproximativa, incompleta) de la estructura sintáctica del texto:
 - Opera en base a **grupos de palabras** o **chunks**
 - Plana, i.e. no contempla estructuras arborescentes
 - Requerimientos menores
 - Mayor robustez
 - Bajo coste

Chunk

- **Def.:** grupo de palabras (segmento) que funcionan conjuntamente como un única *palabra con contenido*:
 - Nombre: funciona a modo de frase/grupo nominal (NP)
 - Adjetivo: a modo de frase/grupo adjetival (AP)
 - Verbo: a modo de frase/grupo verbal (VP)
 - Preposición*: a modo de frase/grupo preposicional (PP)
- Pero no son *frases* en el sentido estricto, sino aproximaciones.
- No hay estructuras recursivas (p.ej. *criador de caballos de carreras*).
 - Se simplifica el proceso de detección.
- (En inglés) Se devuelve el segmento desde la palabra inicial del grupo hasta el núcleo, desechando los modificadores posteriores
 - Influido por la sintaxis (en inglés los modificadores preceden al núcleo)
 - Se evita el problema de la ambigüedad en las adjunciones:

[VP vi] [PP a] [NP un hombre] [PP en] [NP una colina] [PP con] [NP un telescopio]

Como Etiquetación de Palabras

- Un proceso de *chunking* implica:
 - Localizar el segmento/grupo de palabras
 - Identificar su clase
- Puede verse como un **proceso de etiquetación** consistente en ...

(1) ... identificar las palabras que s/n integran el *chunk* (**IOB tagging**):

- Se etiquetan las **palabras**.
- Las **chunk tags** indican dónde comienza un nuevo *chunk*, qué palabras contiene (más sencillo que detectar dónde termina) y el tipo del *chunk* (*tagset* ampliable según categorías consideradas: NP, VP, PP ...):
 - B (*Beginning*): si es la palabra inicial del *chunk*
 - I (*Internal*): si está en el interior del *chunk*
 - O (*Outside*): si está fuera del *chunk*

The	morning	flight	from	Denver	has	arrived
B_NP	I_NP	I_NP	B_PP	B_NP	B_VP	I_VP
B_NP	I_NP	I_NP	O	B_NP	O	O

$[_{NP}$ *The morning flight*] *from* $[_{NP}$ *Denver*] *has arrived.*

Como Etiquetación de Separaciones entre Palabras

(2) ... delimitar el *chunk* mediante paréntesis (parentización):

- Se etiquetan las **separaciones entre palabras**
- Las **gap tags** indican los límites y clase del *chunk* (*tagset* ampliable según categorías consideradas)

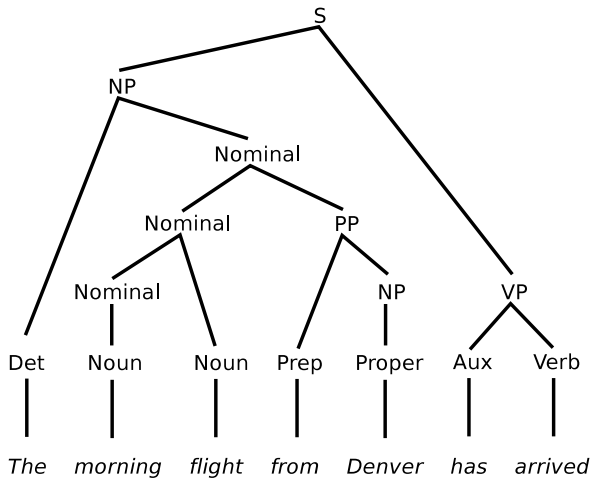
Beginning	End	Between	No bracket (outside)	No bracket (inside)
[<i>NP</i>	<i>NP</i>]	<i>NP</i>	[<i>NP</i> <i>Out</i>	<i>In</i>

[_{NP} The _{In} morning _{In} flight _{NP}] from [_{NP} Denver _{NP}] has _{Out} arrived.
 [_{NP} The morning flight] from [_{NP} Denver] has arrived.

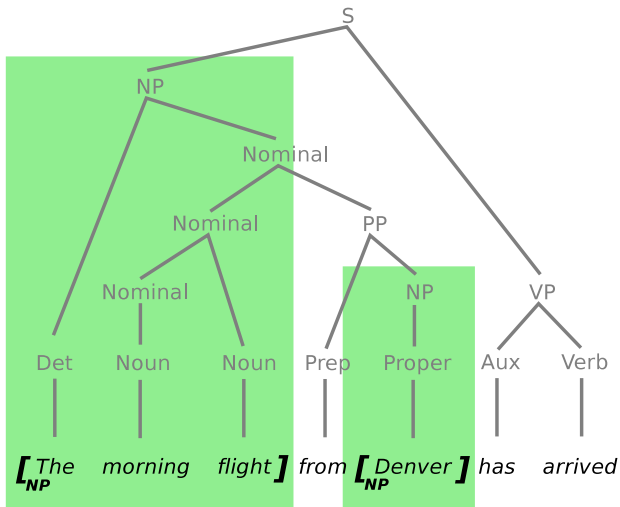
Ejemplo

The morning flight from Denver has arrived

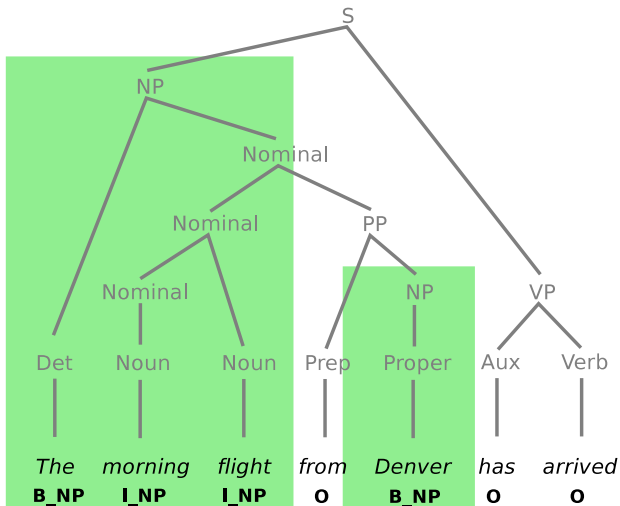
Ejemplo



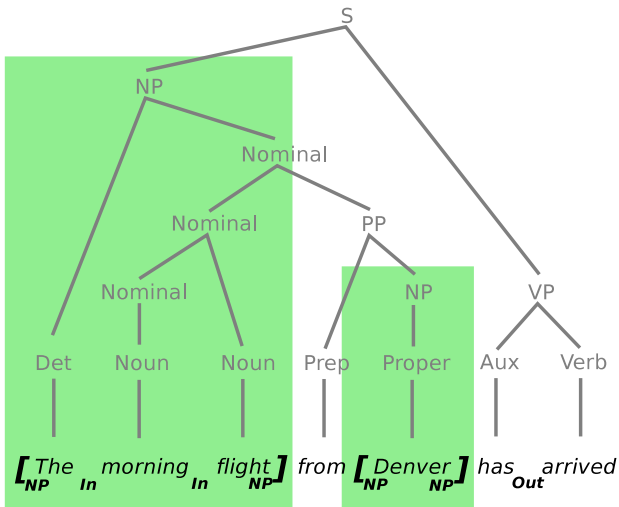
Ejemplo



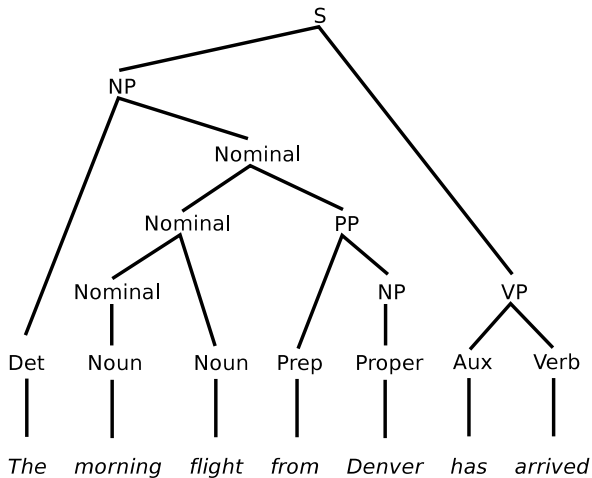
Ejemplo



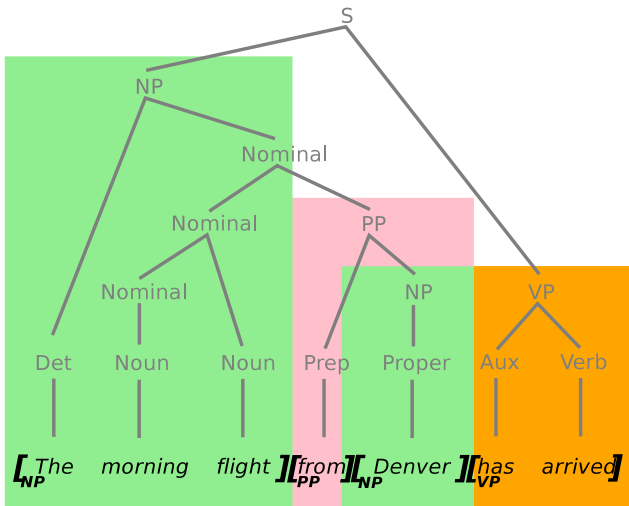
Ejemplo



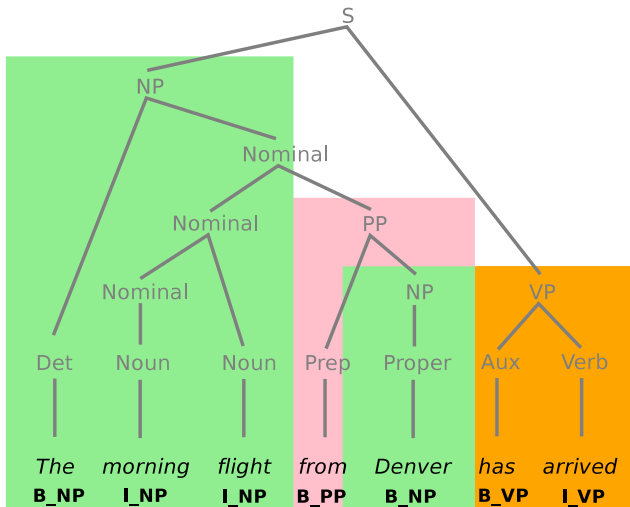
Ejemplo



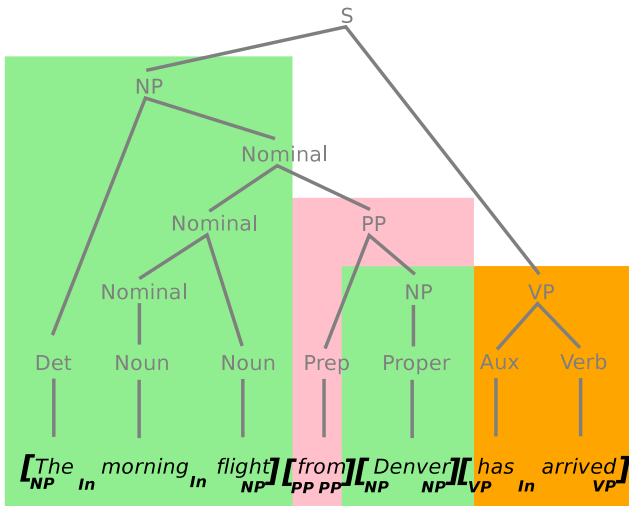
Ejemplo



Ejemplo



Ejemplo



Mediante Correspondencia de Patrones

- a.k.a. *finite-state role-based chunking*
- Se generan manualmente **patrones** que capturen las estructuras de interés:
 - De izqda. a drcha.
 - *Longest matching*
 - No superposición, i.e. el siguiente *matching* empieza justo a continuación del anterior
 - No se permiten estructuras recursivas: p.ej.

Nominal → Nominal PP

- Ejemplos:

NP → [Det] Noun* Noun

NP → Proper

VP → Verb

VP → Aux Verb

Mediante Correspondencia de Patrones (cont.)

- Pueden implementarse mediante **traductores finitos** (*finite-state transducers*, FSTs) emparentados con los autómatas finitos:
 - Eficiencia (complejidad lineal)
 - Simplicidad
- Pueden agruparse por niveles (i.e. en cascada) de forma que la salida del primer nivel sea la entrada al segundo, la salida del segundo la entrada al tercero... Esto permite:
 - Identificar estructuras cada vez más complejas
 - Generar estructuras arborescentes de altura limitada

Ejemplo (Vilares et al., 2008)

docenas de niños muy alegres han tenido que aprender hoy en el colegio una lección de historia

Etiquetador-Lematizador

N	P	N	W	A	V	C	V	W	P	D	N	D	N	P	N
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

docena de niño muy alegre tener que aprender hoy en el colegio un lección de historia

docenas de niños muy alegres han tenido que aprender hoy en el colegio una lección de historia

Nivel 0: Preprocesado

- Identificar expresiones numéricas y de cantidad (*NumP*)
 - e.g. algo más de dos millones
- Preprocesado de expresiones verbales: para simplificar el procesado en niveles superiores
 - e.g. tener_en_cuenta como unidad para evitar que en cuenta sea identificado como complemento del verbo

Nivel 0: Preprocesado

SNum	N	W	A	V	C	V	W	P	D	N	D	N	P	N
------	---	---	---	---	---	---	---	---	---	---	---	---	---	---

N	P	N	W	A	V	C	V	W	P	D	N	D	N	P	N
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

docena de niño muy alegre tener que aprender hoy en el colegio un lección de historia

docenas de niños muy alegres han tenido que aprender hoy en el colegio una lección de historia

Nivel 1: Frases Adverbiales y Verbos No-Perifrásticos

- Frases adverbiales ordinarias:

$$AdvP \rightarrow W^* W_1 \begin{cases} AdvP.lem \doteq W_1.lem \\ AdvP.tag \doteq W_1.tag \end{cases}$$

- Grupos adjetivales con función adverbial, e.g. de forma rápida = rápidamente:

$$AdvP \rightarrow \text{de (forma | manera | modo)} A \begin{cases} AdvP.lem \doteq A.lem \\ AdvP.tag \doteq A.tag \end{cases}$$

Nivel 1: Frases Adverbiales y Verbos No-Perifrásticos

- Formas activas y pasivas (verbo aux. ser)
- Tiempos simples y compuestos (verbo aux. haber)
- Ejemplo: formas compuestas pasivas

$$\text{VG1} \rightarrow \text{V}_1 \text{ V}_2 \text{ V}_3 \left\{ \begin{array}{l}
 \text{VG1.lem} \doteq \text{V}_3.\text{lem} \\
 \text{VG1.tag} \doteq \text{V}_1.\text{tag} \\
 \text{VG1.voice} \doteq \text{PASS} \\
 \text{V}_1.\text{lem} \doteq \text{haber} \\
 \text{V}_2.\text{lem} \doteq \text{ser} \\
 \text{V}_2.\text{tense} \doteq \text{PART} \\
 \text{V}_3.\text{tense} \doteq \text{PART}
 \end{array} \right.$$

Nivel 1: Frases Adverbiales y Verbos No-Perifrásticos



docena de niño muy alegre tener que aprender hoy en el colegio un lección de historia

docenas de niños muy alegres han tenido que aprender hoy en el colegio una lección de historia

Nivel 2: Frases Adjetivales y Perífrasis Verbales

- Su núcleo es un adjetivo, que podría venir precedido por una frase adverbial:

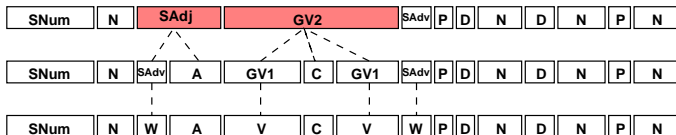
$$AdjP \rightarrow AdvP? A \begin{cases} AdjP.l\text{em} \doteq A.l\text{em} \\ AdjP.tag \doteq A.tag \end{cases}$$

Nivel 2: Frases Adjetivales y Perífrasis Verbales

- e.g. tener+que+*infinitivo*, ir+a+*infinitivo*
- Unión de dos o más formas verbales que funcionan como una unidad.
- Añaden matices de significado tales como obligación, grado de desarrollo de la acción, etc., que no pueden ser expresados mediante las formas verbales normales, simples o compuestas.
- Ejemplo: perífrasis de infinitivo

$$\begin{array}{l}
 VG2 \rightarrow VG1_1 \text{ (me|te|se)? (que|de|a)? } VG1_2 \\
 \left\{ \begin{array}{l}
 VG2.l\text{em} \doteq VG1_2.l\text{em} \\
 VG2.t\text{ag} \doteq VG1_1.t\text{ag} \\
 VG2.v\text{oic}\text{e} \doteq VG1_2.v\text{oic}\text{e} \\
 VG1_1.v\text{oic}\text{e} \doteq \text{ACT} \\
 VG2_2.t\text{ens}\text{e} \doteq \text{INF}
 \end{array} \right.
 \end{array}$$

Nivel 2: Frases Adjetivales y Perífrasis Verbales



docena de niño muy alegre tener que aprender hoy en el colegio un lección de historia

docenas de niños muy alegres han tenido que aprender hoy en el colegio una lección de historia

Nivel 3: Frases Nominales

- Existencia de *complementos partitivos (PC)*; e.g. ninguno de
- Secuencias/coordinaciones de frases adjetivales como post-modificadores (*AdjPostModif*)

$$\begin{array}{lcl}
 \text{AdjPostModif} & \rightarrow & \text{AdjP } Cc \text{ AdjP} \\
 & & | \text{ AdjP} \\
 & & | \text{ AdjP } \text{AdjP} \\
 & & | \text{ AdjP } \text{AdjP } \text{AdjP}
 \end{array}$$

- Existencia de posibles determinantes y frases adjetivales modificadoras antepuestos al núcleo nominal

$NP \rightarrow PC?$

$$\begin{array}{l}
 D^* \text{ (AdjP | Number | NumP)?} \\
 (N | Acronym | Proper)^* \\
 (N | Acronym | Proper)_1 \\
 \text{AdjPostModif?}
 \end{array}
 \left\{ \begin{array}{l}
 NP.lem \doteq ()_1.lem \\
 NP.tag \doteq ()_1.tag \\
 NP.num \doteq PC.num
 \end{array} \right.$$

Nivel 3: Frases Nominales

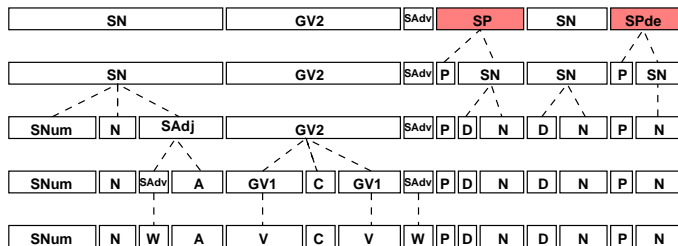


Nivel 4: Frases Preposicionales

- Para facilitar la extracción de términos en fases posteriores distinguiremos 3 tipos según la preposición:
 - *PPde*: preposición de
 - *PPpor*: preposición por
 - *PP*: otras
- Ejemplo: frases preposicionales introducidas mediante de

$$PPof \rightarrow P \ NP \left\{ \begin{array}{l} P.lem \doteq de \\ PP.lem \doteq NP.lem \\ PP.tag \doteq NP.tag \end{array} \right.$$

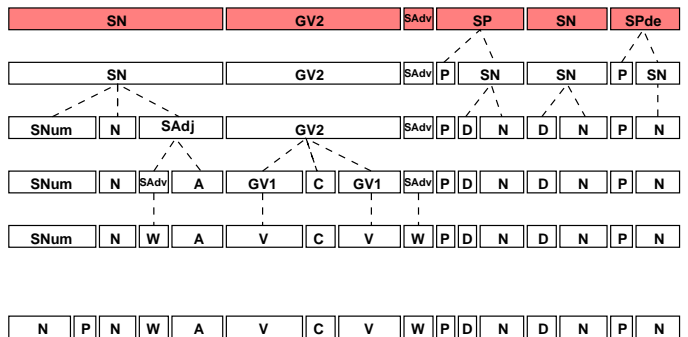
Nivel 4: Frases Preposicionales



docena de niño muy alegre tener que aprender hoy en el colegio un lección de historia

docenas de niños muy alegres han tenido que aprender hoy en el colegio una lección de historia

Resultado final del análisis



docena de niño muy alegre tener que aprender hoy en el colegio una lección de historia
 docenas de niños muy alegres han tenido que aprender hoy en el colegio una lección de historia

Mediante Reglas Aprendidas Automáticamente

- Similar al *etiquetador de Brill* pero para IOB tagging:
 - Etiqueta inicial: en base a la etiqueta morfosintáctica (*part-of-speech/PoS tag*) de la palabra
 - Se le asigna el *chunk tag* (I,O,B) más frecuente para esa categoría
 - Reglas de transformación: en base a la forma, etiqueta morfosintáctica y *chunk tag* actuales de la palabra y sus contiguas

Mediante Reglas Aprendidas Automáticamente (cont.)

- Necesidad de un **corpus de entrenamiento**:
 - Textos con las frases de interés previamente delimitadas y etiquetadas
 - Problema: muy costoso de crear
 - Solución: reutilizar *treebanks* ya existentes
 - Se toma un árbol sintáctico del *treebank*
 - Se identifican sus frases/grupos básicos (NP, VP, PP, ...) no recursivos
 - Se [re]anotan convenientemente

Ejemplo de Reglas Aprendidas Automáticamente

W_0, W_{-1}, W_1 : palabras actual, a la izquierda y a la derecha, respectivamente

P_0, P_{-1}, P_1 : ídem para las etiquetas morfosintácticas

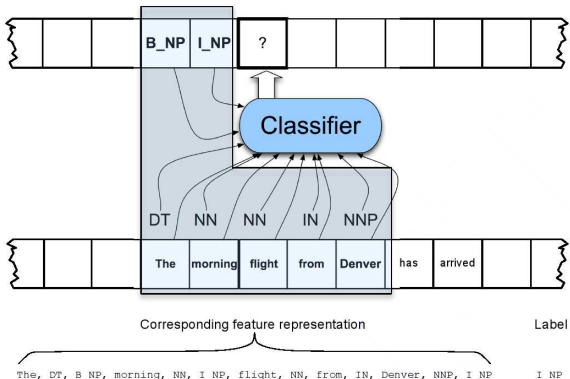
T_0, T_{-1}, T_1 : ídem para las *chunk tags*

Pasada	Anterior	Contexto	Nueva
1	I	$T_1=O, P_0=ADJ$	O
2	-	$T_{-2}=I, P_{-1}=I, P_0=DET$	B
		...	

- 1 Una *chunk tag* I pasa a ser O cuando la etiqueta morfosintáctica de la palabra actual es un adjetivo (ADJ) y la siguiente palabra tiene un *chunk tag* O.
- 2 Asignamos una *chunk tag* B a la palabra actual si los *chunk tag* de las dos palabras anteriores son I y la etiqueta morfosintáctica de la palabra actual es un determinante (DET)

Mediante Clasificadores Secuenciales

- P.ej. etiquetación estocástica basada en modelos de Markov
- Aproximaciones posibles (ya introducidas):
 - Etiquetación de palabras
 - Etiquetación de separaciones entre palabras (i.e. parentización)
- **Etiquetación de palabras:** etiquetar cada palabra en base a la forma, etiqueta morfosintáctica y *chunk tag* de ella misma y sus contiguas



Mediante Clasificadores Secuenciales (cont.)

- **Etiquetación de separaciones:** determinar la secuencia de *gap tags* $G = g_2, g_3 \dots g_n$ óptima en función de las etiquetas morfosintácticas $T = t_1, t_2 \dots t_n$ y las formas $W = w_1, w_2 \dots w_n$ de las palabras que separan:

- i.e. maximizar $P(G) = \prod_{i=2}^n P(g_i | w_{i-1}, t_{i-1}, w_i, t_i)$

Referencias

- [Abney, 1997] Abney, S. (1997). Partial Parsing via Finite-State Cascades. In *Natural Language Engineering*, 2(4), 337–244.
- [Jurafsky & Martin, 2009] Jurafsky, D. & Martin, J.H. (2009). Chapter 13: Syntactic Parsing. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (2nd ed.)*. Pearson–Prentice Hall.
- [Nugues, 2006] Nugues, P.M. (2006). *An Introduction to Language Processing with Perl and Prolog*. Springer-Verlag.
- [Vilares et al., 2008] Vilares, J., Alonso, M.A. & Vilares, M. (2008). Extraction of Complex Index Terms in Non-English IR: A Shallow Parsing Based Approach. *Information Processing & Management*, 44(4), 1517–1537.