Introducción al Procesamiento de Lenguaje Natural

Grupo PLN – InCo

Sintaxis

Sintaxis

- La sintaxis refiere al modo en el que las palabras se ordenan en la oración.

- Tiene que ver con la **estructura** que tienen las oraciones **NO** con su significado.

Sintaxis

- Vamos a ver dos enfoques:
  - gramáticas de constituyentes
  - gramáticas de dependencias

Existen otros:

gramáticas de rasgos (como HPSG),

gramáticas categoriales,

…

Gramática de Constituyentes

Se basan en la noción de sintagma/frase/grupo:

- nombre – GRUPO NOMINAL
- verbo – GRUPO VERBAL
- adjetivo – GRUPO ADJETIVAL
- adverbio – GRUPO ADVERBIAL
- preposición – GRUPO PREPOSICIONAL

Gramática Libre de Contexto

Y en las gramáticas libres de contexto:

Una Gramática Libre de Contexto es una tupla con 4 elementos:

$G = (V,T,P,S)$

- V → conjunto de símbolos variables
- T → conjunto de símbolos terminales
- S ∈ V, símbolo inicial
- P → conjunto de reglas de producción :

  A → α , con α secuencia de símbolos de V ∪ T,
  eventualmente vacía (α = ε)

Ejemplo: S → aSb | ε ;

G = ({S},{a,b},S, {S → aSb | ε })

Ejercicio GLC

Construir una GLC para las siguientes **oraciones**

El niño cocina
Pedro cocina en la cocina
María cocina un guiso de lentejas
El niño cocina un guiso en la cocina

Ejercicio GLC

[El niño]GN [cocina]GV
[Pedro]GN [cocina [en la cocina]GP]GV
[María]GN [cocina [un guiso [de lentejas]GP]GN]GV
[El niño]GN [cocina [un guiso]GN [en la cocina]GP]GV

O –> GN GV
GN –> N | D N | D N GP
GV –> V | V GN | GV GP
GP –> P GN
D – > el | la | un
N –> niño | cocina | guiso | lentejas | Pedro | María
V –> cocina
P –> de | en

Ejercicio GLC

[El niño]GN [cocina]GV
[Pedro]GN [cocina [en la cocina]GP]GV
[María]GN [cocina [un guiso [de lentejas]GP]GN]GV
[El niño]GN [cocina [un guiso]GN [en la cocina]GP]GV

O –> GN GV
GN –> N | D N | D N GP
GV –> V | V GN | GV GP
GP –> P GN
D – > el | la | un
N –> niño | cocina | guiso | lentejas | Pedro | María
V –> cocina
P –> de | en

REGLAS LÉXICAS: normalmente son muy numerosas, deberían introducir todo el léxico

Ejercicio GLC

[El niño]GN [cocina]GV
[Pedro]GN [cocina [en la cocina]GP]GV
[María]GN [cocina [un guiso [de lentejas]GP]GN]GV
[El niño]GN [cocina [un guiso]GN [en la cocina]GP]GV

O –> GN GV
GN –> N | D N | D N GP
GV –> V | V GN | GV GP
GP –> P GN
D – > el | la | un
N –> niño | cocina | guiso | lentejas | Pedro | María
V –> cocina
P –> de | en

podría ser GN GP
pero como análisis lingüístico es más apropiado que el GP modifique solo al N, y no a D N

podría ser V GN GP
pero es común que dentro del GV haya varios GP

Ejercicio GLC

[El niño]GN [cocina]GV
[Pedro]GN [cocina [en la cocina]GP]GV
[María]GN [cocina [un guiso [de lentejas]GP]GN]GV
[El niño]GN [cocina [un guiso]GN [en la cocina]GP]GV

O –> GN GV
GN –> N | D N | D N GP
GV –> V | V GN | GV GP
GP –> P GN
D – > el | la | un
N –> niño | cocina | guiso | lentejas | Pedro | María
V –> cocina
P –> de | en

Esta gramática genera dos análisis para las dos últimas oraciones:
[V [..]GN [..]GP]GV
[V [.. [..]GP]GN]GV

No podemos resolver esta ambigüedad con este tipo de reglas.

Problemas en los análisis con GLC

- Coordinación
- Sobregeneración
- Subcategorización sintáctica
- Subcategorización semántica
- Ambigüedad (por ejemplo, PP attachment)

Problemas en los análisis con GLC

- **Sobregeneración**
- **Subcategorización sintáctica**
- **Subcategorización semántica**

Las reglas generan (o reconocen)
* la guiso * un guiso cocina en la niño

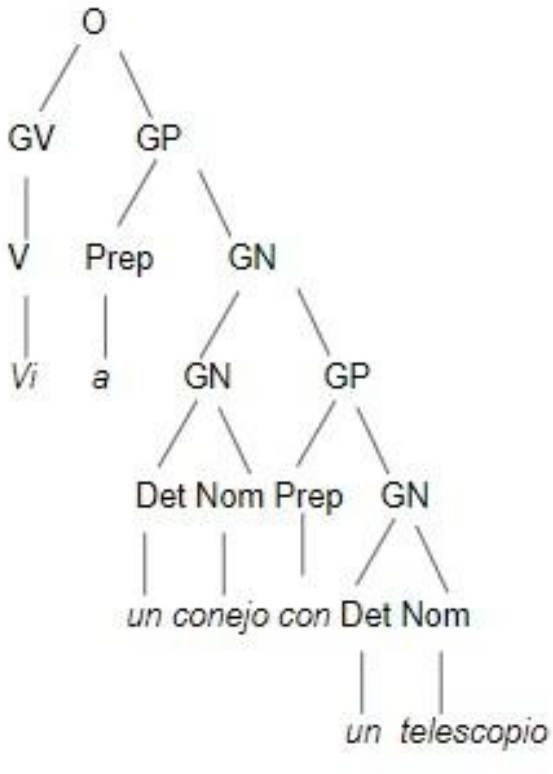
Soluciones:

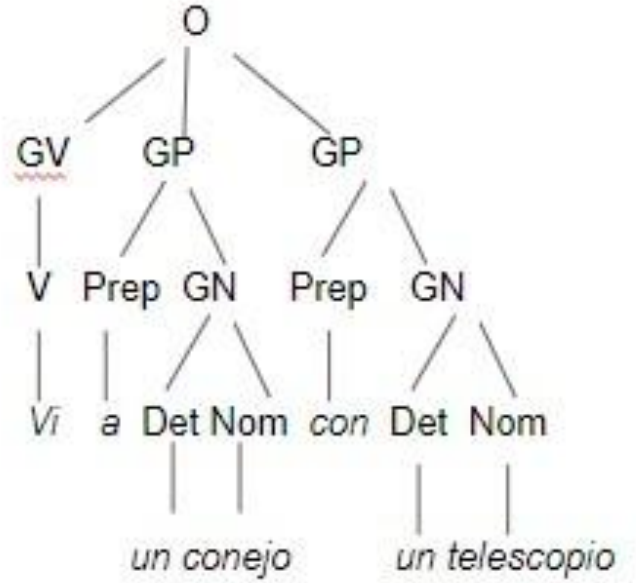
- agregar símbolos no terminales más específicos (NSingMas, NSingFem, …), pero la cantidad de reglas se vuelve inmanejable y se pierden conceptos más generales (como el concepto de nombre).
- gramáticas de rasgos

Problemas en los análisis con GLC

- Ambigüedad (por ejemplo, PP attachment)

Vi a un conejo con un telescopio





Problemas en los análisis con GLC

- Ambigüedad (por ejemplo, PP attachment)
  - Un enfoque para resolverla: GLC probabilísticas
    - se calculan probabilidades de las diferentes estructuras, o sea, de las reglas que las generan, a partir de las ocurrencias en un gran corpus anotado con árboles sintácticos.

Material complementario: anexo C del libro

Ejemplo (adapt. Manning)

S	→	NP VP	1.0	NP	→	NP PP	0.4
VP	→	V NP	0.7	NP	→	*Juan*	0.1
VP	→	VP PP	0.3	NP	→	*María*	0.18
PP	→	P NP	1.0	NP	→	*arroz*	0.04
P	→	*con*	1.0	NP	→	*cuchara*	0.18
V	→	*comió*	1.0	NP	→	*queso*	0.1

Juan comió arroz con cuchara

Penn Treebank

- El *Penn Treebank* es un corpus anotado, ampliamente usado (inglés), mantenido por el LDC (*Linguistic Data Consortium*)
- Es un corpus de árboles lingüísticos
- Cada oración ha sido anotada con su estructura sintáctica (árbol) y algo de información semántica.
- Se trabajó fundamentalmente con el Wall Street Journal: un millón de palabras (unas 40.000 oraciones de ediciones del Wall Street Journal del período 1987-1989)

Ejemplo Penn Treebank

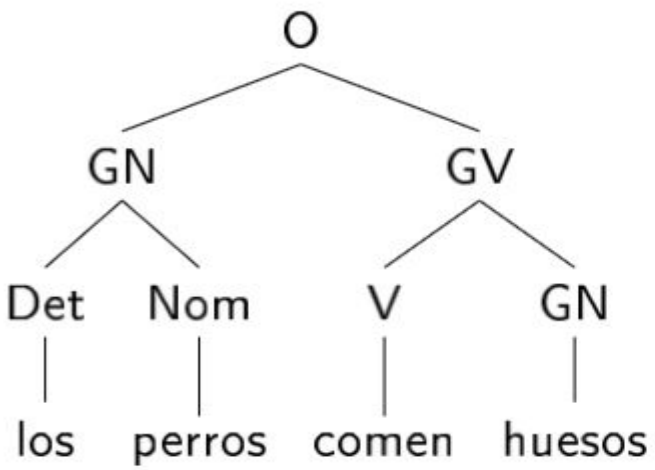
```
( (S
    (NP-SBJ (DT The) (NN move))
    (VP (VBD followed)
      (NP
        (NP (DT a) (NN round))
        (PP (IN of)
          (NP
            (NP (JJ similar) (NNS increases))
            (PP (IN by)
              (NP (JJ other) (NNS lenders)))
            (PP (IN against)
              (NP (NNP Arizona) (JJ real) (NN estate) (NNS loans))))))
      (, ,)
      (S-ADV
        (NP-SBJ (-NONE- *))
        (VP (VBG reflecting)
          (NP
            (NP (DT a) (VBG continuing) (NN decline))
            (PP-LOC (IN in)
              (NP (DT that) (NN market)))))))
    (. .)))
```

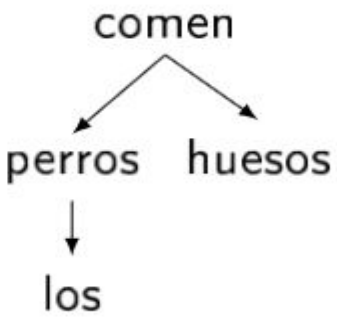
The move followed a round of similar increases by other lenders against Arizona real estate loans, reflecting a continuing decline in that market.

Gramáticas de Dependencias

No se basan en la noción de constituyente.

Ejemplo: *Los perros comen huesos.*





Gramáticas de Dependencias

- Relaciones *biléxicas* (palabra a palabra)
- Asimétricas: una de las palabras gobierna a la otra
- Etiquetadas (en general) con la función sintáctica

En el ejemplo: *Los perros comen huesos*

"*comen*" gobierna a "*perros*" y "*huesos*"
"*perros*" gobierna a "*los*"

Gramáticas de Dependencias

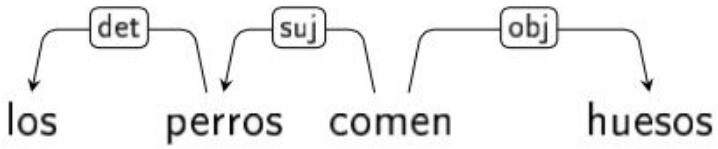
Clausal Argument Relations	Description
NSUBJ	Nominal subject
DOBJ	Direct object
IOBJ	Indirect object
CCOMP	Clausal complement
XCOMP	Open clausal complement
Nominal Modifier Relations	**Description**
NMOD	Nominal modifier
AMOD	Adjectival modifier
NUMMOD	Numeric modifier
APPOS	Appositional modifier
DET	Determiner
CASE	Prepositions, postpositions and other case markers
Other Notable Relations	**Description**
CONJ	Conjunct
CC	Coordinating conjunction

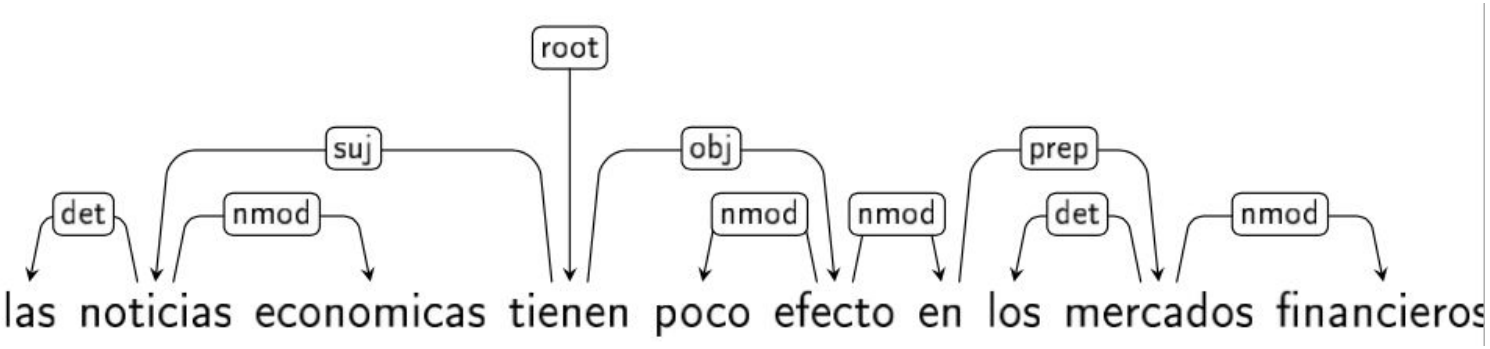
subconjunto de relaciones del Proyecto *Universal Dependencies* (Nivre et al. 2016)

Gramáticas de Dependencias

Ejemplos

Los perros comen huesos.





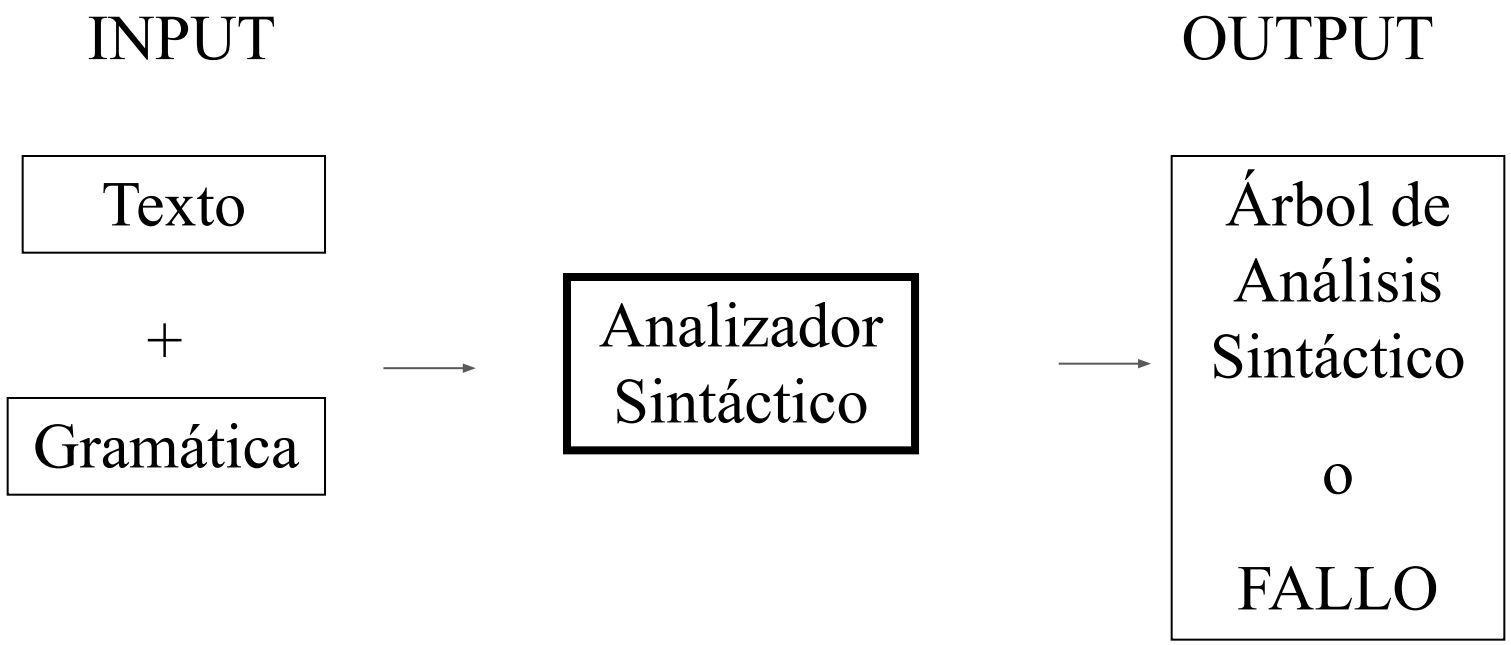
Gramáticas de Dependencias

- Las etiquetas de función suelen estar orientadas a la sintaxis (*suj*, *obj*, *nmod*)
- Se dibuja un:

  1. Grafo conexo, dirigido
  2. Acíclico
  3. Single head: un solo padre por cada nodo y un nodo raíz (sin padre)

  Estas restricciones implican que el grafo es un árbol.
  De hecho, 1 y 3 implican 2

Análisis Sintáctico



Esquema general de un analizador sintáctico

Análisis Sintáctico

Dos grandes clases de análisis sintáctico:

- Ascendentes (bottom-up)

  Para una secuencia de palabras, construye un árbol de análisis sintáctico desde las hojas a la raíz

- Descendentes (top-down)

  Para una secuencia de palabras, construye un árbol de análisis sintáctico desde la raíz a las hojas

Algoritmos de parsing

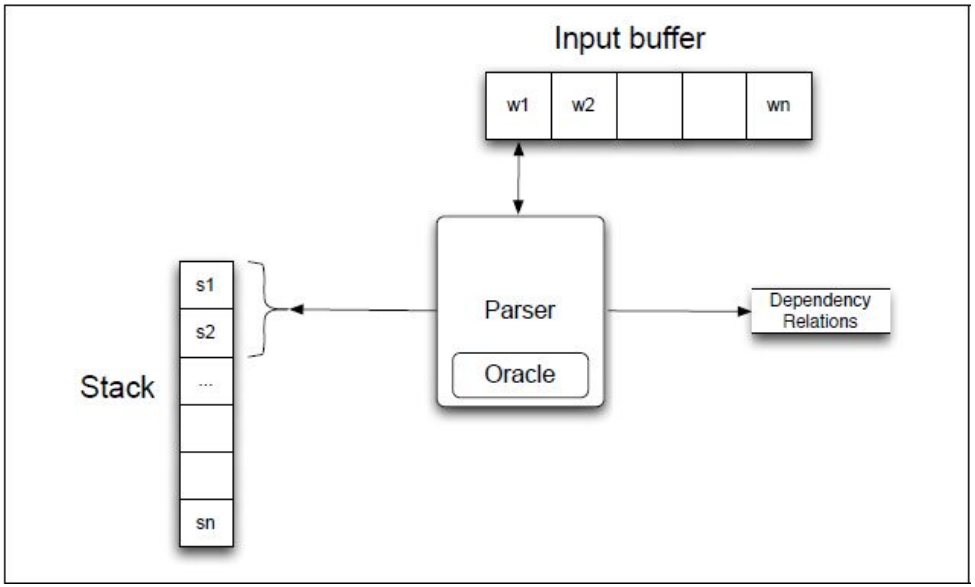
- Cocke-Kasami-Younger (CKY, 1965-1967)
  - Bottom-up. A partir de un par de categorías (lado derecho de las reglas) generamos el lado izquierdo.
  - Exige que la gramática esté en FNC ($A \rightarrow BC$; $A \rightarrow a$) .
  - Se construye una matriz de 2 dimensiones (N x N).
  - A partir de ella se puede construir el árbol sintáctico.
  - Cada celda indica el análisis posible para el span (i,j)
  - En cada celda se mira si es posible que haya un k, con $i<k<j$, tal que una regla $A \rightarrow B\ C$, cumple que B es un posible símbolo para el span (i,k) y C para el span (k,j).
  - En la diagonal se ponen las posibles categrías de las palabras. Se miran las reglas léxicas.

Ejemplo CKY

Analizamos utilizando CKY ***Pedro cocina en la cocina***

(Ejemplo resuelto en clase)

Parsing de dependencias



Algoritmo basado en transiciones

- **LEFTARC:** Agrega una relación entre la palabra superior del stack (head) y la palabra siguiente; esta es eliminada del stack. En general se determina también qué rótulo asignar a la relación.
- **RIGHTARC:** Agrega una relación entre la segunda palabra del stack (head) y la palabra superior del stack; elimina la palabra situada en la parte superior. En general se determina también qué rótulo asignar a la relación.
- **SHIFT:** Elimina la palabra actual del búfer de entrada y le hace push en el stack.

Parsing de dependencias

Algoritmo basado en transiciones

- En cada paso del parsing un **oráculo** define qué transición aplicar.
- Diferentes enfoques para entrenar el oráculo, en general, aprendizaje supervisado en base a un corpus de árboles de dependencias.
- Malt Parser (Nivre et al, 2007):
  - Agrega una cuarta transición (*reduce*).
  - Oráculo: Se agregan atributos a los elementos del stack y del buffer (lemas, pos-tags, etc.) y se miran algunas palabras anteriores o posteriores a la que se está analizando. Algoritmo kNN.
- Actual: enfoques neuronales.

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S. and Marsi, E. (2007) MaltParser: A language-independent system for data-driven dependency parsing. Natural Language Engineering, 13(2), 95-135

Parsing de dependencias

Analizamos utilizando el algoritmo basado en transiciones

Pedro cocina en la cocina

(Ejemplo comentado en clase.)

Referencias

Daniel Jurafsky and James H. Martin. 2024. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models, 3rd edition, draft. Online manuscript released August 20, 2024. https://web.stanford.edu/~jurafsky/slp3.

Capítulos 18 y 19.