Introducción a Natural Language Processing

Análisis Sintáctico

Ronald Cárdenas Acosta

Setiembre, 2016

Outline

Análisis de constituyentes

- Una forma de modelar un lenguaje natural es aprendiendo su sintáxis o gramática.
- **Parsing**: análisis sintáctico de un oración. Puede ser basado en constituyentes o en dependencias.
- El Análisis de constituyentes o *Phrase-based Parsing* genera un árbol de constituyentes de manera recursiva.

Gramática No Contextual (Context Free Grammars)

- Noam Chomsky propuso una forma de generación de lenguaje a través de reglas.
- La Gramática No Contextual (CFG) fue diseñada para generar lenguaje sintácticamente válido recursivamente.
- En un inicio, los CFGs fueron modelados por máquinas finitas de estado (*finite state machines*)

Gramática No Contextual (Context Free Grammars)

Un CFG se define como: $G = \langle \mathcal{N}, \mathcal{T}, \mathcal{R}, S \rangle$, donde:

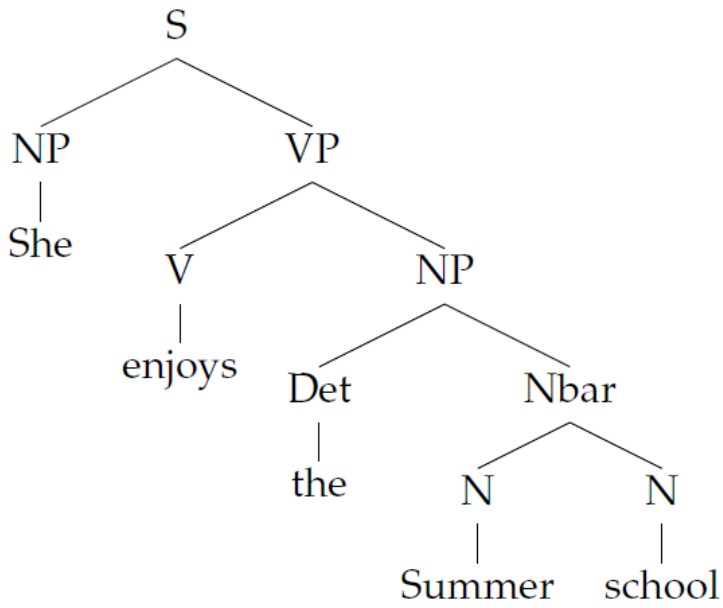
- $\mathcal{N}$: conjunto finito de símbolos no terminales. Representan cada tipo de constituyente.
  Notación: letra mayúscula (A,B,...)
- $\mathcal{T}$: conjunto finito de símbolos terminales, es decir, las palabras de la oración.
  Notación: letra minúscula (a,b,...)
- $\mathcal{R}$: conjunto de reglas de producción que relacionan $\mathcal{N}$ con $(\mathcal{N} \cup \mathcal{T})$
- S: *símbolo inicial*, usado para representar a la oración entera. Debe ser un elemento de $\mathcal{N}$

Se dice que G está en **forma normal Chomsky (Chomsky normal form CNF)** si las reglas en $\mathcal{R}$ son solo binarias ($A \rightarrow BC$) o unarias ($A \rightarrow a$)

CFGs: ejemplo

Considere el siguiente CFG en Chomsky Normal Form, junto a un Parse tree generado del mismo:

```
S --> NP VP
NP --> Det N
NP --> She
VP --> V NP
V --> enjoys
Det --> the
Nbar --> N N
N --> Summer
N --> school
```



Ambigüedad

The man sees the boy in the park with a telescope.

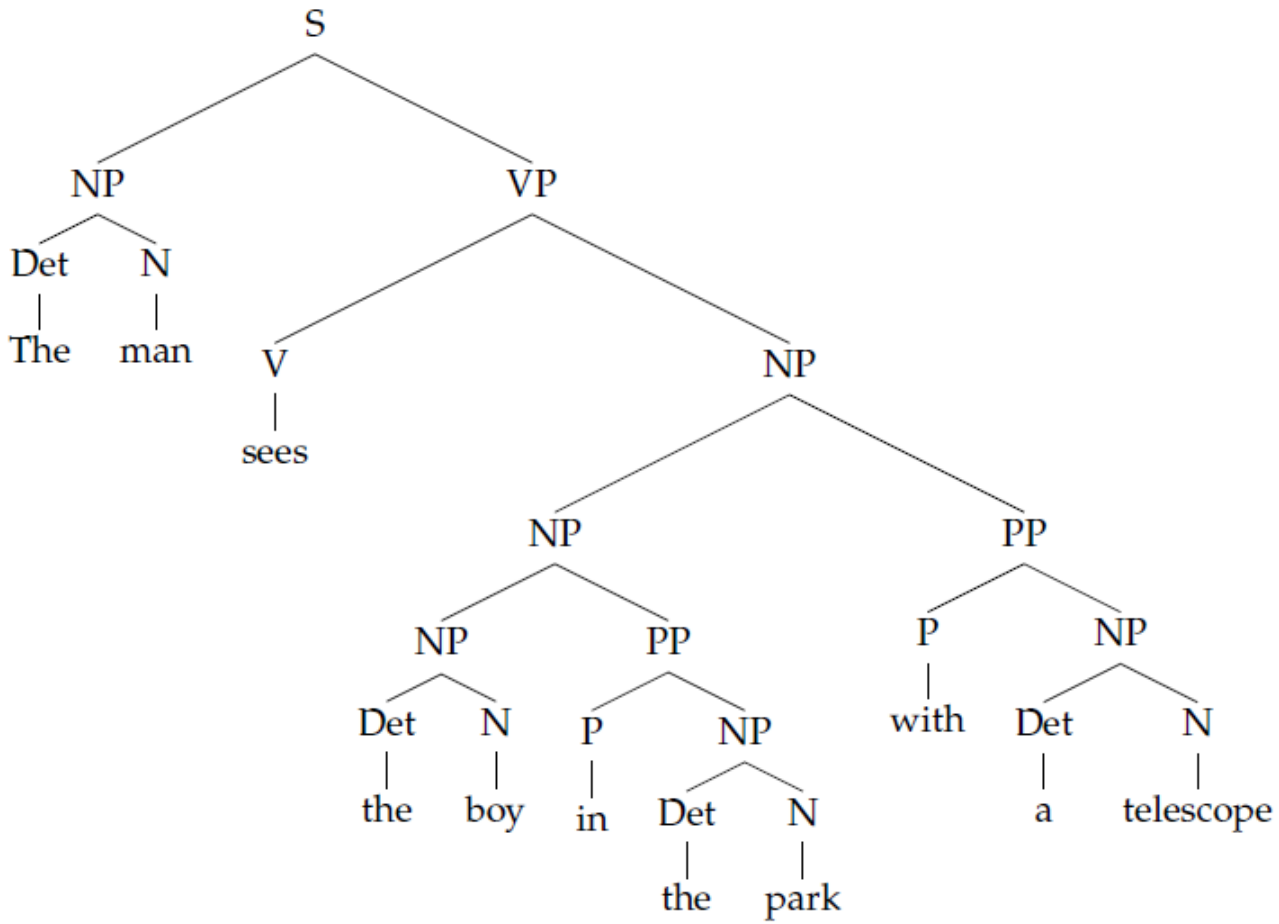


Figure: Interpretación 1:El niño esta en el parque y tiene un telescopio.

Ambigüedad

The man sees the boy in the park with a telescope.

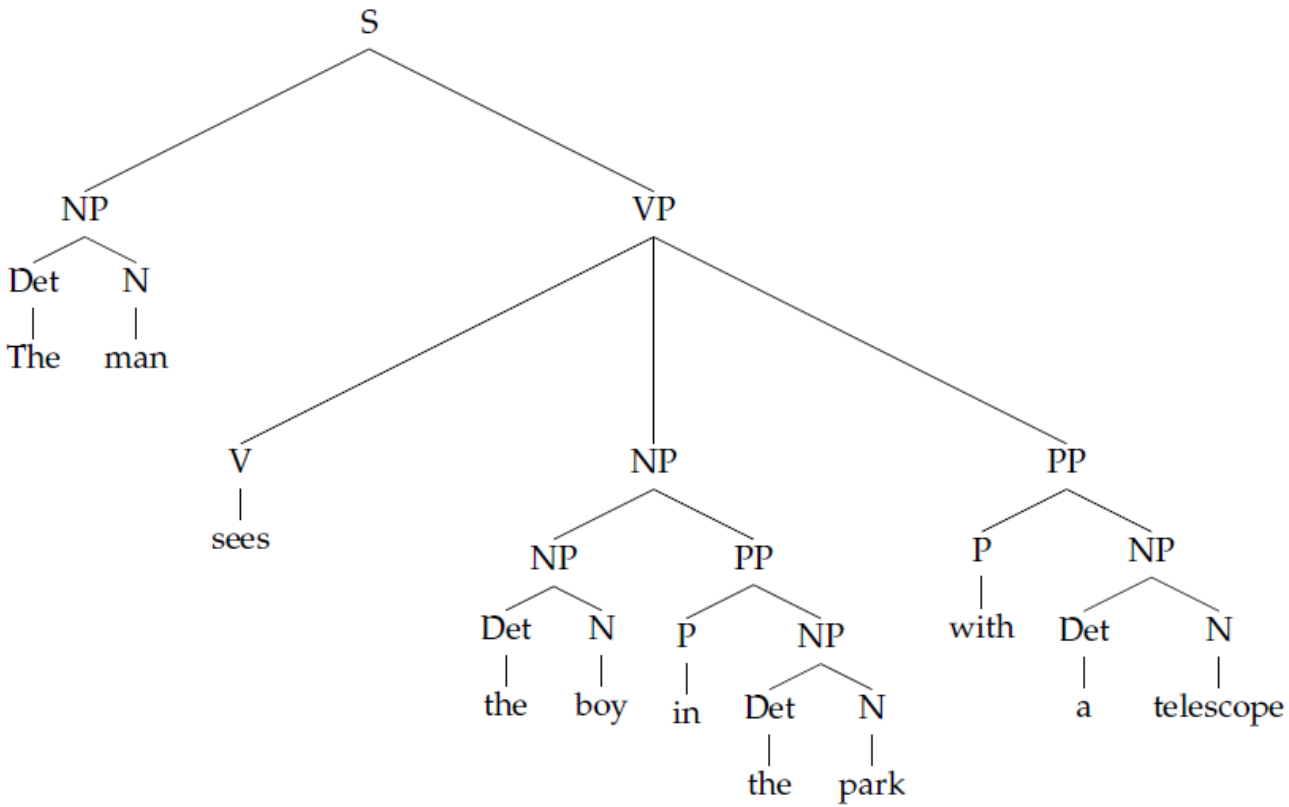


Figure: Interpretación 2:El niño esta en el parque, y el hombre lo ve usando un telescopio.

Ambigüedad

The man sees the boy in the park with a telescope.

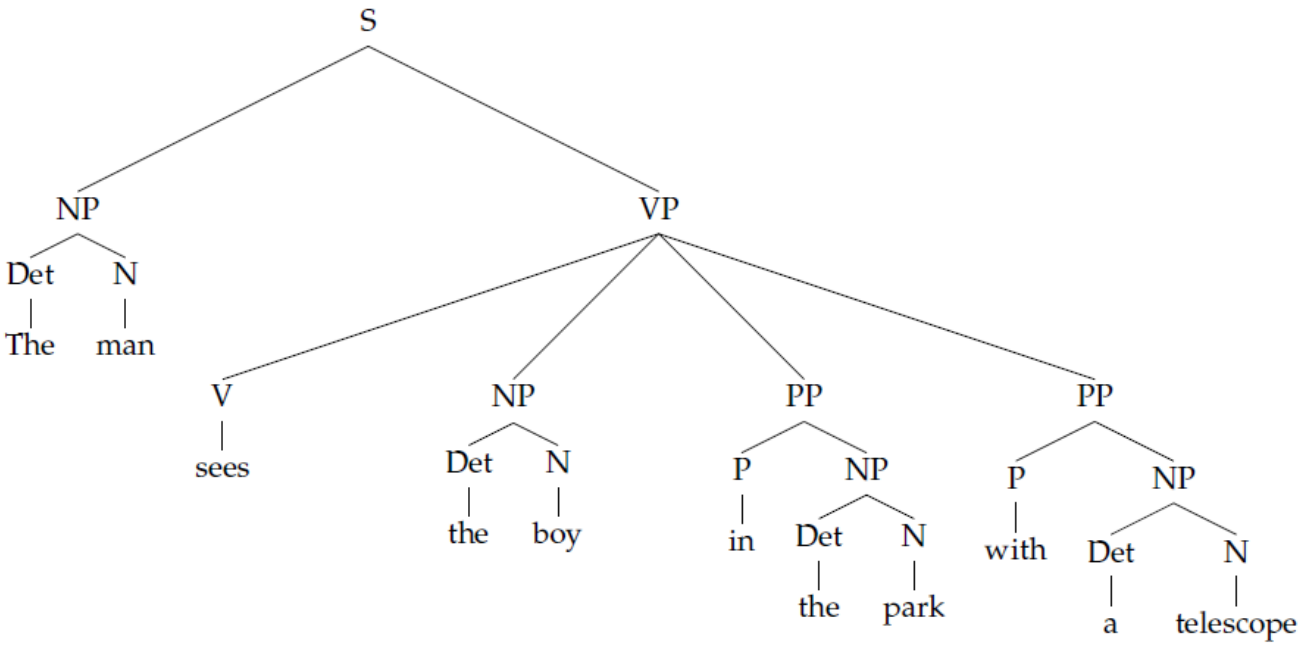


Figure: Interpretación 3:El hombre esta en el parque y tiene un telescopio con el cual ve a un niño en algún lugar.

Probabilistic Context Free Grammar (PCFG)

Un PCFG está definido por $G = \langle \mathcal{N}, \mathcal{T}, \mathcal{R}, S, \theta \rangle$, donde $\langle \mathcal{N}, \mathcal{T}, \mathcal{R}, S \rangle$ es un CFG y θ es un vector de parámetros, uno por cada regla en $\mathcal{R}$. Asumiendo que el CFG está en Chomsky normal form, se tiene:

- $\theta_{Z \to XY} = P_\theta(XY|Z)$
- $\theta_{Z \to w} = P_\theta(w|Z)$

Para que estas prob. condicionales estén bien definidas, se debe cumplir que:

$$\sum_{X,Y \in \mathcal{N}} \theta_{Z \to XY} + \sum_{w \in \mathcal{T}} \theta_{Z \to w} = 1$$

PCFGs: Inferencia

Sea s una oración y t un parse tree derivado de s. Para cada regla $r \in \mathcal{R}$, sea $n_r(t, s)$ el número de veces que la regla r aparece en t. La probabilidad conjunta de s y t esta definida como:

$$P(t, s) = \prod_{r \in \mathcal{R}} \theta_r^{n_r(t,s)}$$

Para la oración: She enjoys the Summer school

$$\begin{aligned} P(t,s) \quad = \quad & P(\texttt{NP VP}|\texttt{S}) \times P(\texttt{She}|\texttt{NP}) \times P(\texttt{V NP}|\texttt{VP}) \times P(\texttt{enjoys}|\texttt{V}) \\ & \times P(\texttt{Det Nbar}|\texttt{NP}) \times P(\texttt{the}|\texttt{Det}) \times P(\texttt{N N}|\texttt{Nbar}) \\ & \times P(\texttt{Summer}|\texttt{N}) \times P(\texttt{school}|\texttt{N}). \end{aligned}$$

PCFGs: Inferencia

- Se debe escoger el árbol óptimo tal que:

$$\hat{t} = \arg\max_{t} P(t|s)$$

- La estimación de parámetros se da por *Maximum Likelihood Estimate*:

$$P(X, Y|Z) = \frac{count(Z \rightarrow XY)}{count(Z \rightarrow **)}$$

- El número de árboles posibles crece exponencialmente con la longitud de s
- Maximización directa es insoluble.

Algoritmo CKY

- Generalización del algoritmo de Viterbi para parsing, conocido como el algoritmo *Inside-Outside*.
- CKY: Cocke-Kasami-Younger
- Dada una gramática en CNF con $|\mathcal{R}|$ reglas de producción, la complejidad de ejecución para analizar una oración de longitud N es $O(N^3)|\mathcal{R}|$
- Asigna una probabilidad $\delta(i, j, Z)$ a cada posible sub-árbol con regla $Z \in \mathcal{R}$

Algoritmo CKY

Algorithm 12 CKY algorithm

input: probabilistic CFG G_θ in CNF, and sentence $s = w_1 \dots w_N$ (each w_i is a word)

{We'll fill the CKY table bottom-up with partial probabilities δ and backtrack pointers ψ. This is similar to the Viterbi algorithm but it works for parse trees rather than sequences.}

{Initialization}
for $i = 1$ **to** N **do**
  for each production rule $r \in \mathcal{R}$ of the form $Z \to w_i$ **do**
    $\delta(i, i, Z) = \theta_{Z \to w_i}$
  end for
end for

{Induction}
for $i = 2$ **to** N **do** {i is length of span}
  for $j = 1$ **to** $N - i + 1$ **do** {j is start of span}
    for each non-terminal $Z \in \mathcal{N}$ **do**
      Set partial probability:

$$\delta(j, j+i-1, Z) = \max_{\substack{X,Y \\ j<k<j+i}} \delta(j, k-1, X) \times \delta(k, j+i-2, Y) \times \theta_{Z \to XY}$$

      Store backpointer:

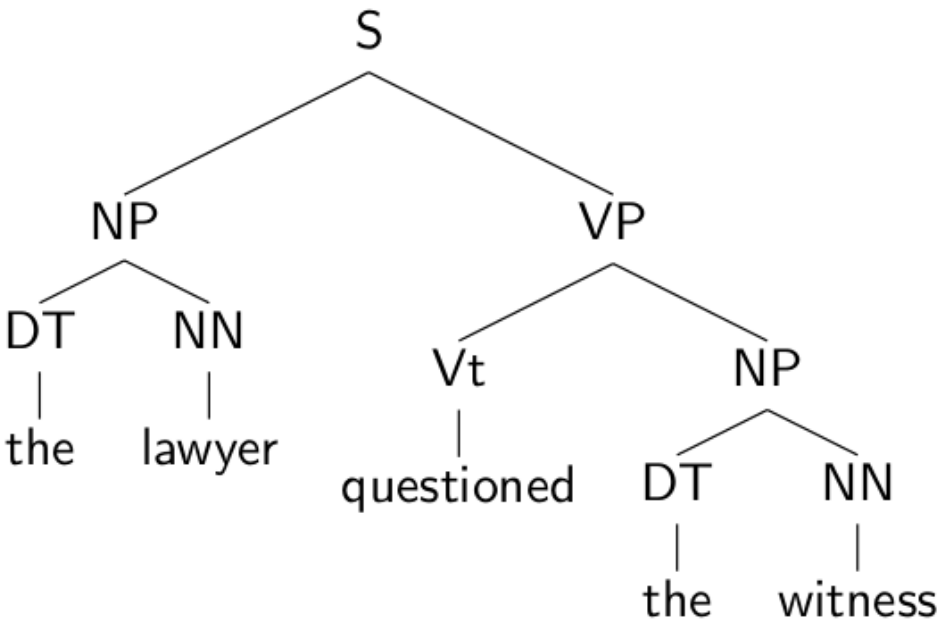
$$\psi(j, j+i-1, Z) = \arg\max_{\substack{X,Y \\ j<k<j+i}} \delta(j, k-1, X) \times \delta(k, j+i-2, Y) \times \theta_{Z \to XY}$$

    end for
  end for
end for

{Termination}
$P(s, \hat{t}) = \delta(1, N, S)$
Backtrack through ψ to obtain most likely parse tree $\hat{t}$

Evaluación de Parsers

La evaluación se hace muestra por muestra.
Se representa cada constituyente como la terna $(Z, inicio, final)$



Label	Start Point	End Point
NP	1	2
NP	4	5
VP	3	5
S	1	5

Evaluación de Parsers: Precision y Recall

Label	Start Point	End Point
NP	1	2
NP	4	5
NP	4	8
PP	6	8
NP	7	8
VP	3	8
S	1	8

Label	Start Point	End Point
NP	1	2
NP	4	5
PP	6	8
NP	7	8
VP	3	8
S	1	8

- G: num. de const. en "gold standard" o muestra de entrenamiento
- P: num. de const. en arbol predecido
- C: num. de const. correctos

$$Precision = P/G, Recall = C/G$$

Parsers refinados

Un número de mejores han dado lugar a parsers más potentes:

- **Lexicalización**:
  Consiste en anotar cada nodo fraseal con el *item lexico* (palabra) que gobierna tal frase.
- **Modelos Discriminativos**:
  Permite considerar features no locales. El vector de features se descompone de acuerdo a los símbolos no terminales y a las reglas de producción.
- **Variables latentes**:
  Consiste en anotar cada nodo fraseal con una variable latente u oculta. Variando la cantidad de variables ocultas se controla la granularidad del parser.

Parsers refinados

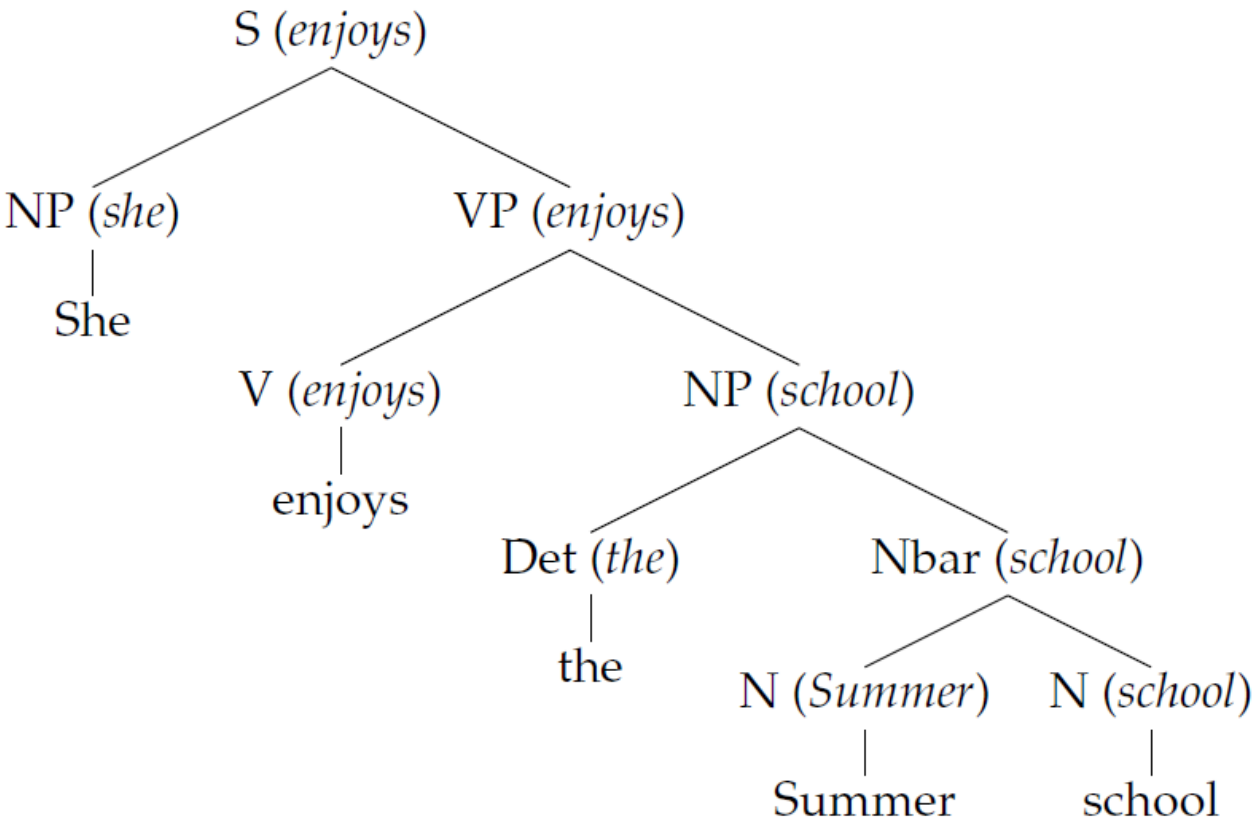


Figure: Lexicalized tree. Salida de Parsers lexicalizados

Parsers refinados

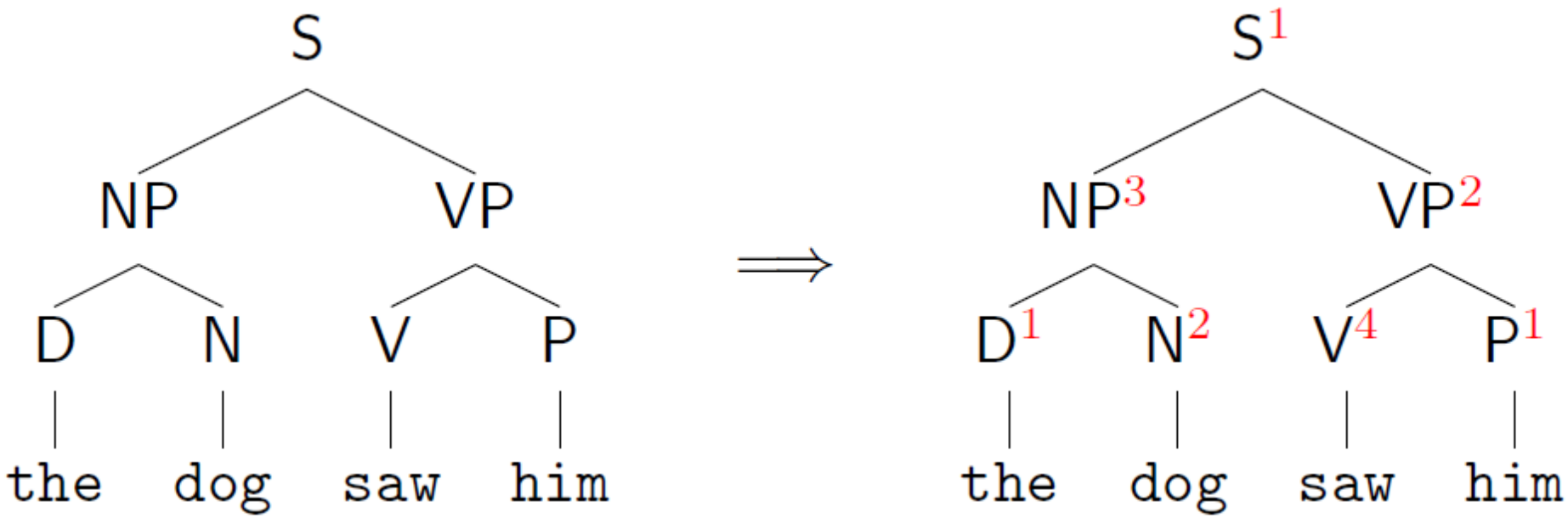


Figure: Latent PCFG tree. Salida de Parsers que usan variables latentes

Parsing: Estado del arte

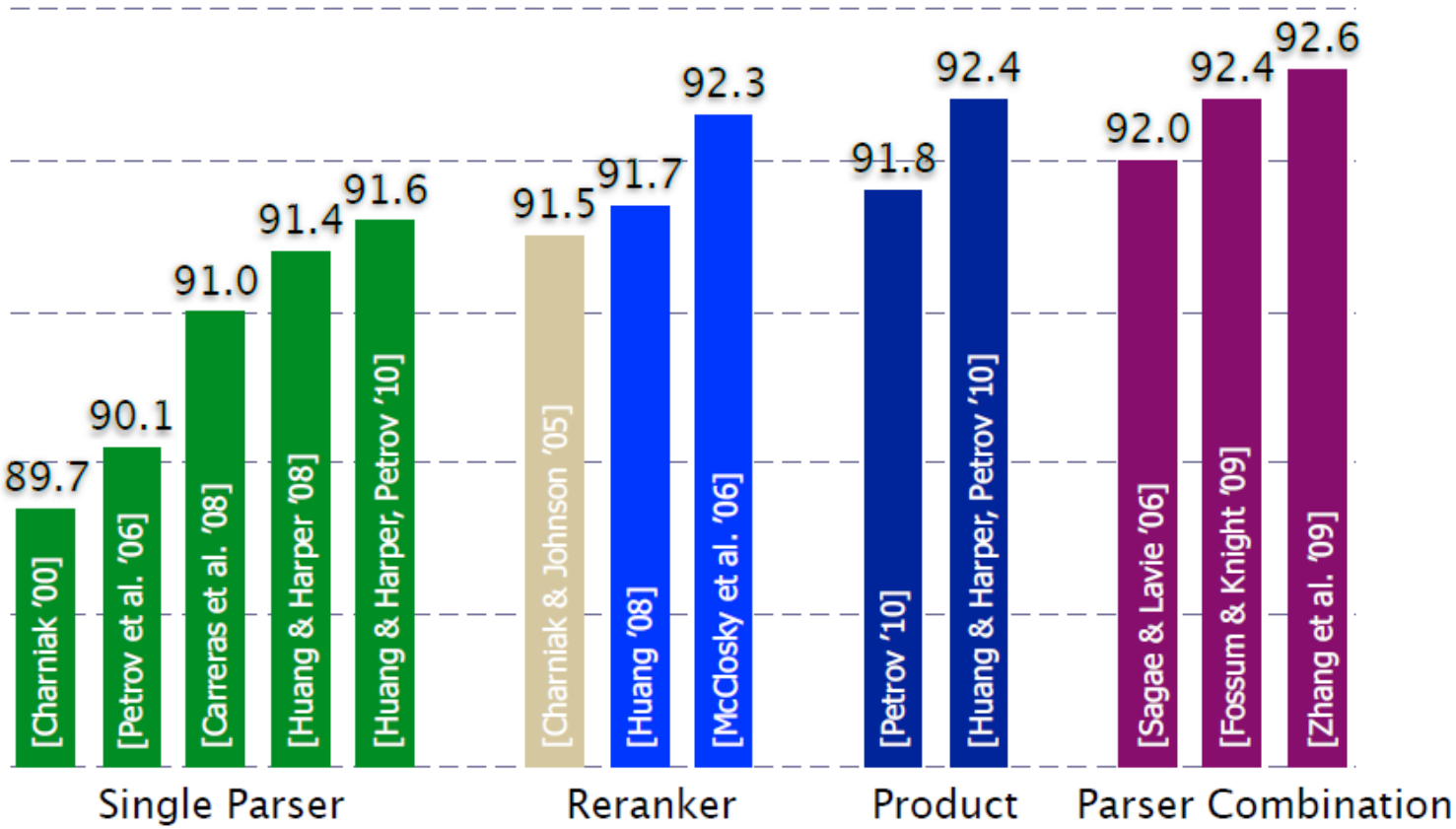


Figure: Parsing en Inglés

Parsing: Estado del arte

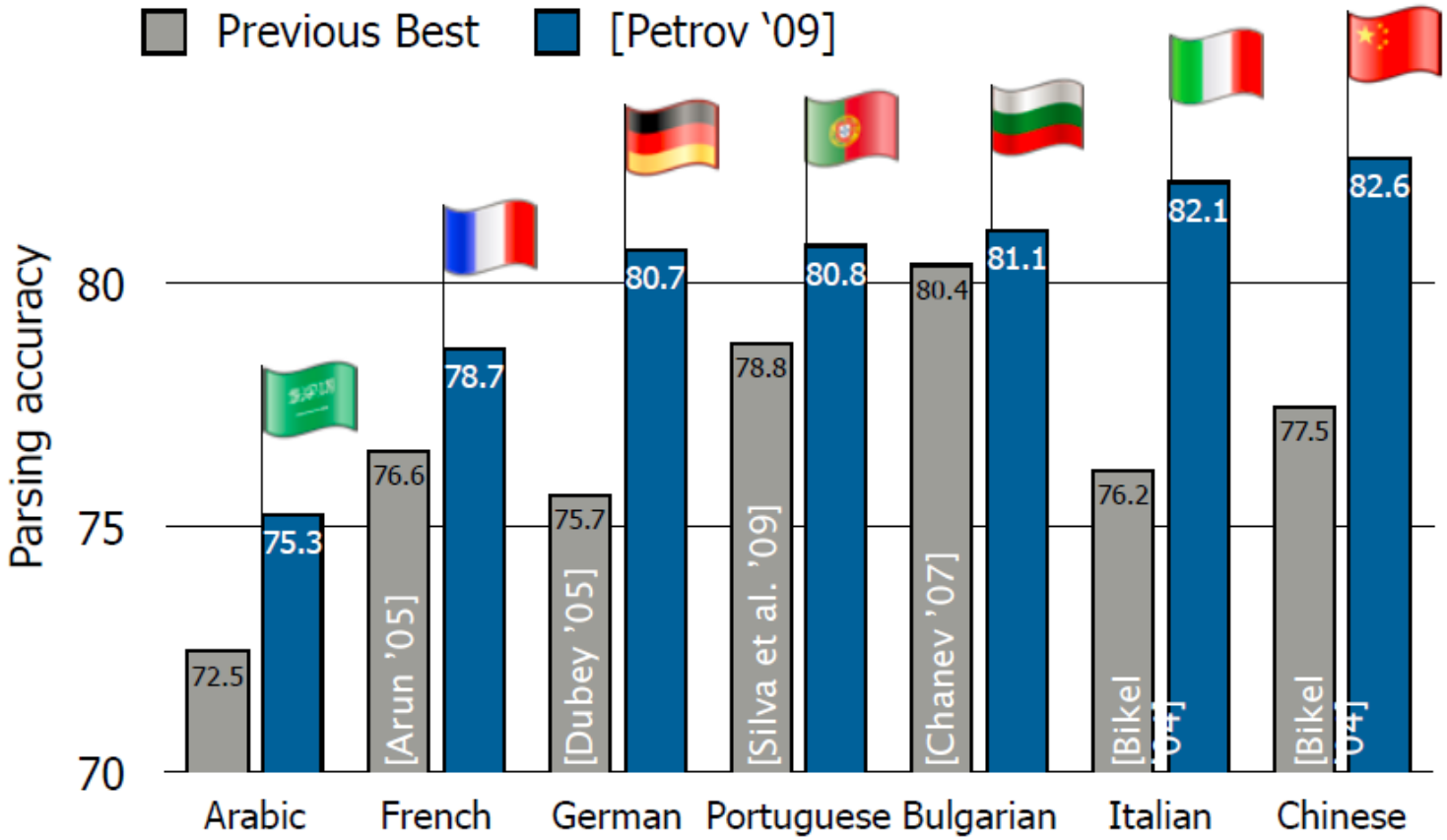


Figure: Parsing para otros lenguajes.