Introducción a Natural Language Processing

Análisis de Dependencias

Ronald Cárdenas Acosta

Octubre, 2016

Outline

Análisis de Dependencias

- Objetivo: Obtener el árbol de dependencias gramaticales más probable.
- No depende de gramáticas.
- El árbol de dependencias puede o no tener anotaciones en sus aristas (sujeto, objeto directo,etc).
- El enfoque de análisis pasa de un estructuras sintácticas y gramáticas generativas a gramáticas lexicas y transformativas.

Árboles de Dependencias

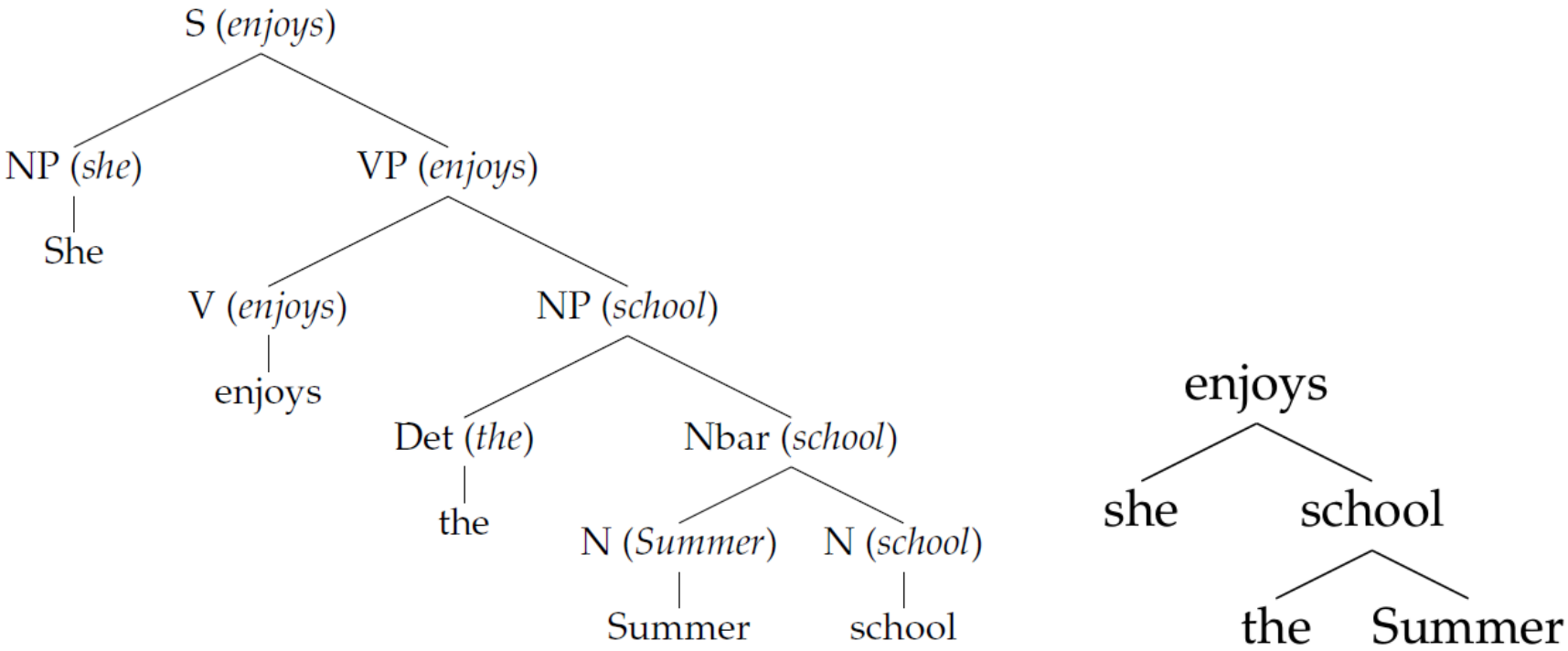


Figure: Árbol de constituyentes lexicalizado puede ser reducido a un árbol de dependencias

Árboles de Dependencias

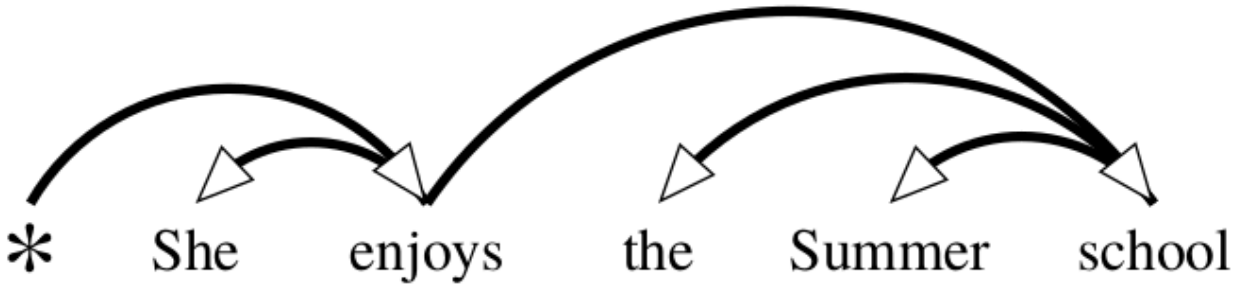


Figure: Representación alternativa. Nodo adicional agregado como raíz.

Análisis Proyectivo y No Proyectivo

Todo árbol de depencias debe presentar las siguientes características:

- Cada palabra (excepto la raiz adicional) tiene un único padre.
- La raíz adicional no tiene padre.
- No tiene ciclos.
- La raíz adicional tiene un único hijo.

Análisis Proyectivo y No Proyectivo

Análisis Proyectivo

Un árbol de dependecias será proyectivo si además sus arcos son *proyectivos*:

Para un arco $< u, v >$, todas las palabras entre u y v son descendientes de u.

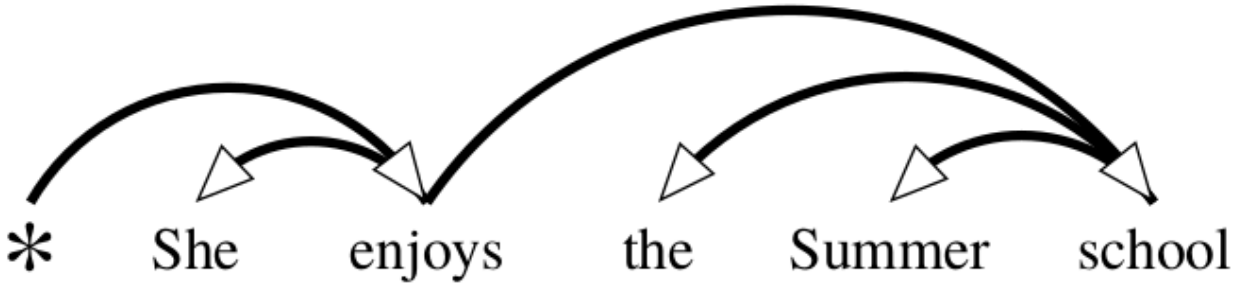


Figure: Árbol proyectivo.

Análisis Proyectivo y No Proyectivo

Análisis No Proyectivo

Un árbol de dependecias será no proyectivo si sus arcos no cumplen la condición proyectiva.

Todo lenguaje presenta relaciones no proyectivas, en especial los lenguajes de gramática de libre orden.

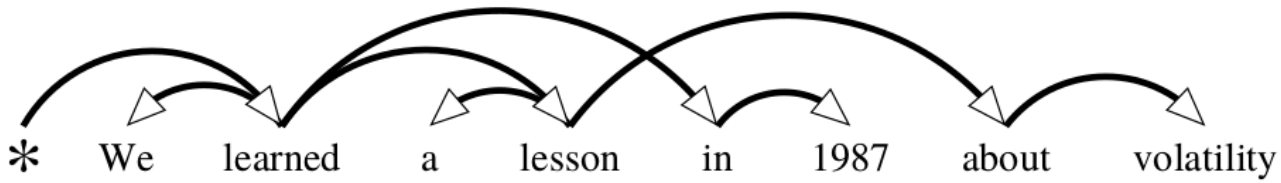


Figure: Árbol no proyectivo.

Projective Dependency parsing

- Nos centraremos en modelos basados en arcos.
- Se asigna un *score* $s_\theta(u, v)$ a cada posible arco que conecte un par de palabras. El *score* del árbol será:

$$score_\theta(t) = \sum_{<u,v>\in t} s_\theta(u, v)$$

- Para inferir el árbol óptimo, se tiene:

$$\hat{t} = \arg\max_t score_\theta(t)$$

Algoritmo de Eisner

- Tiene complejidad en tiempo de $O(N^3)$, donde N es la longitud de la oración.
- Esta relacionado con el algoritmo CKY, en que procesa de una manera *bottom-up*.
- Usa la noción de spans completos e incompletos.

Algoritmo de Eisner

Algorithm 13 Eisner's algorithm for first-order projective dependency parsing

input: Arc scores $s_\theta(h,m)$, for $h \in \{0,\ldots,N\}$, $m \in \{1,\ldots,N\}$, and $h \neq m$, associated with a sentence $s = w_1 \ldots w_N$.
{Initialization}
for $i = 0$ **to** N **do**
  {Initialize incomplete spans.}
  $\text{incomplete}[i,i,\leftarrow] := 0.0$
  $\text{incomplete}[i,i,\rightarrow] := 0.0$

  {Initialize complete spans.}
  $\text{complete}[i,i,\leftarrow] := 0.0$
  $\text{complete}[i,i,\rightarrow] := 0.0$
end for

{Induction}
for $k = 1$ **to** N **do** {k is length of span}
  for $s = 1$ **to** $N - k$ **do** {s is start of span}
    Set $t := s + k$ {t is end of span}

    {First, create incomplete spans.}
    $\text{incomplete}[s,t,\leftarrow] := \max_{s \leq r < t}(\text{complete}[s][r][\rightarrow] + \text{complete}[r+1][t][\leftarrow] + s_\theta(t,s))$
    $\text{incomplete}[s,t,\rightarrow] := \max_{s \leq r < t}(\text{complete}[s][r][\rightarrow] + \text{complete}[r+1][t][\leftarrow] + s_\theta(s,t))$

    {Then, create complete spans.}
    $\text{complete}[s,t,\leftarrow] := \max_{s \leq r < t}(\text{complete}[s][r][\leftarrow] + \text{incomplete}[r][t][\leftarrow]$
    $\text{complete}[s,t,\rightarrow] := \max_{s < r \leq t}(\text{incomplete}[s][r][\rightarrow] + \text{complete}[r][t][\rightarrow])$
  end for
end for

{Termination}
Backtrack to obtain the actual tree, whose score is $\text{complete}[0,N,\rightarrow]$.

Algoritmo de Eisner: Formación de spans

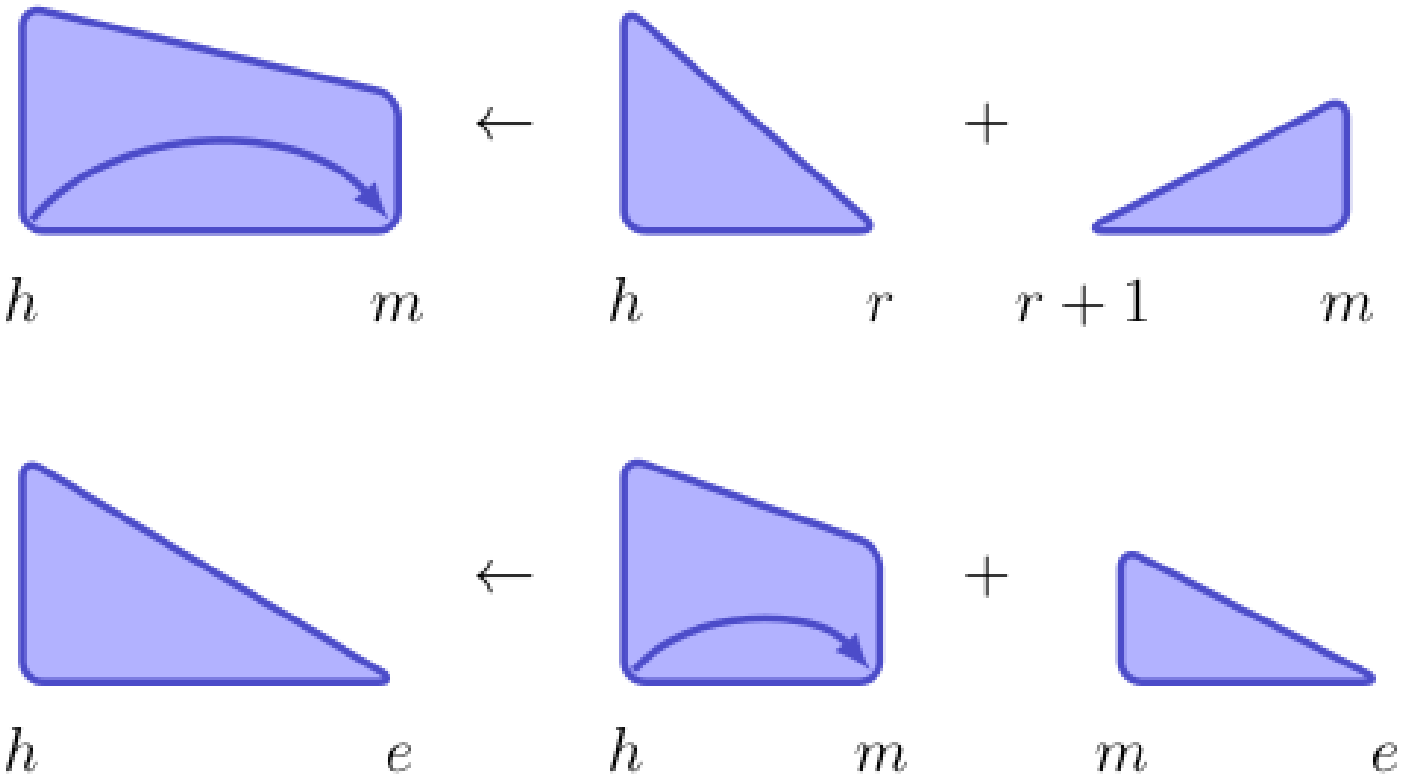


Figure: Fila de arriba: formación de un span incompleto. Fila de abajo: formación de un span completo.

Algoritmo de Eisner: Formación de spans

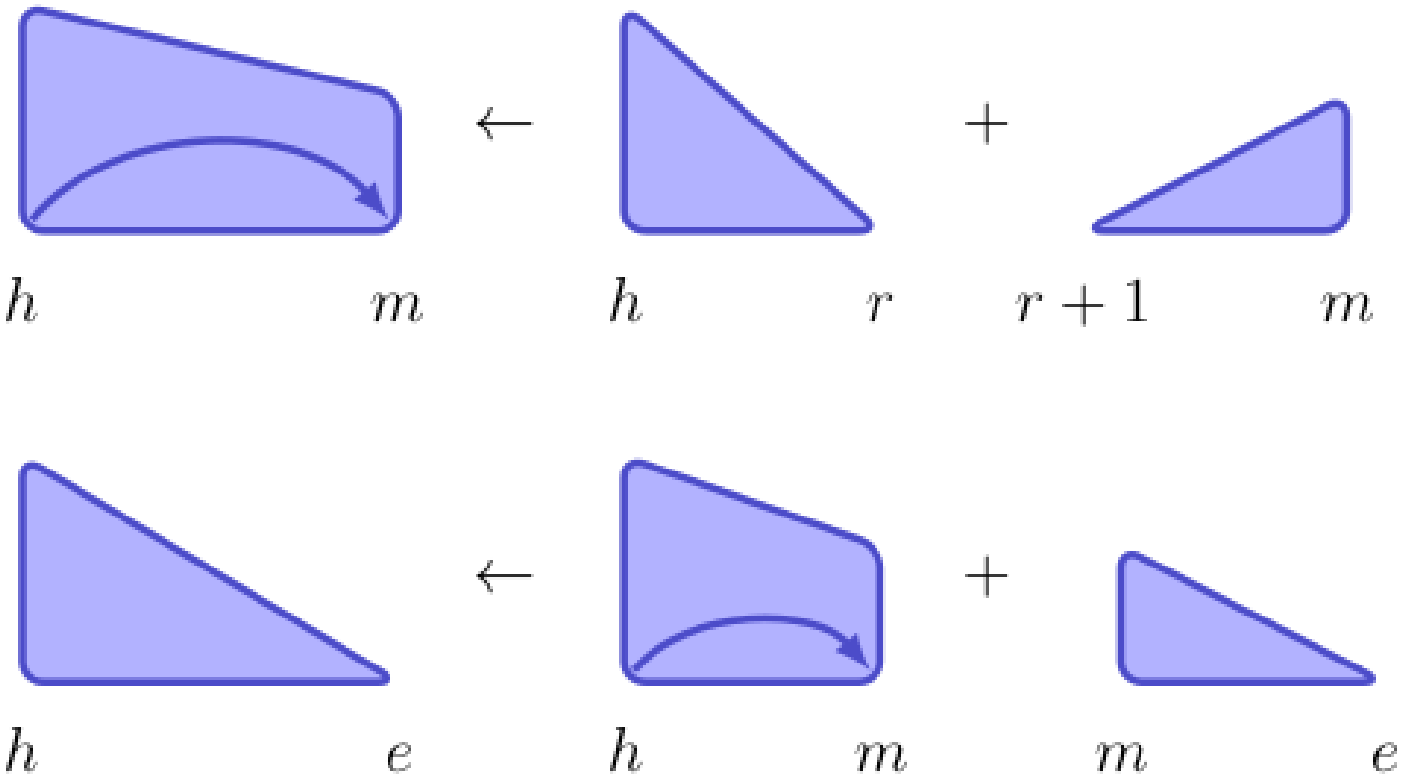
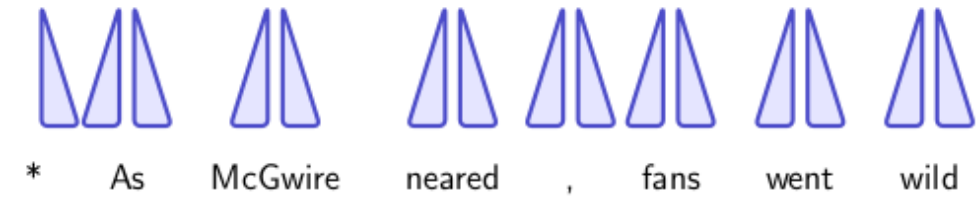
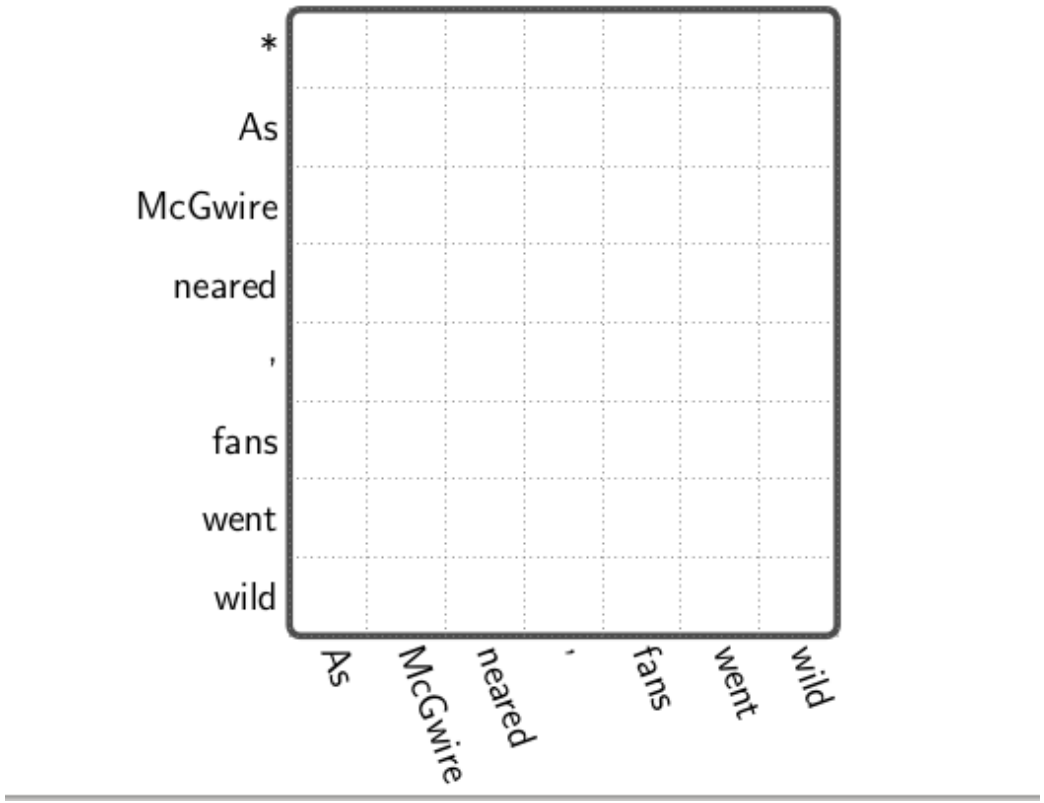


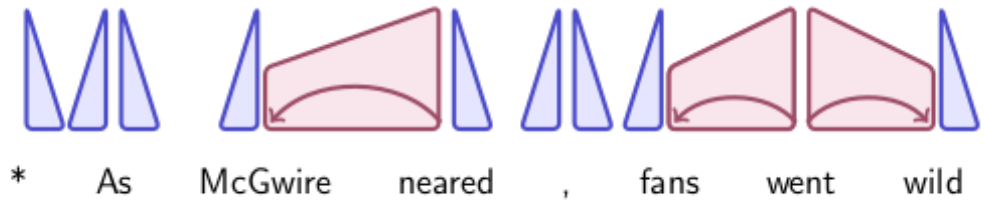
Figure: Fila de arriba: formación de un span incompleto. Fila de abajo: formación de un span completo.

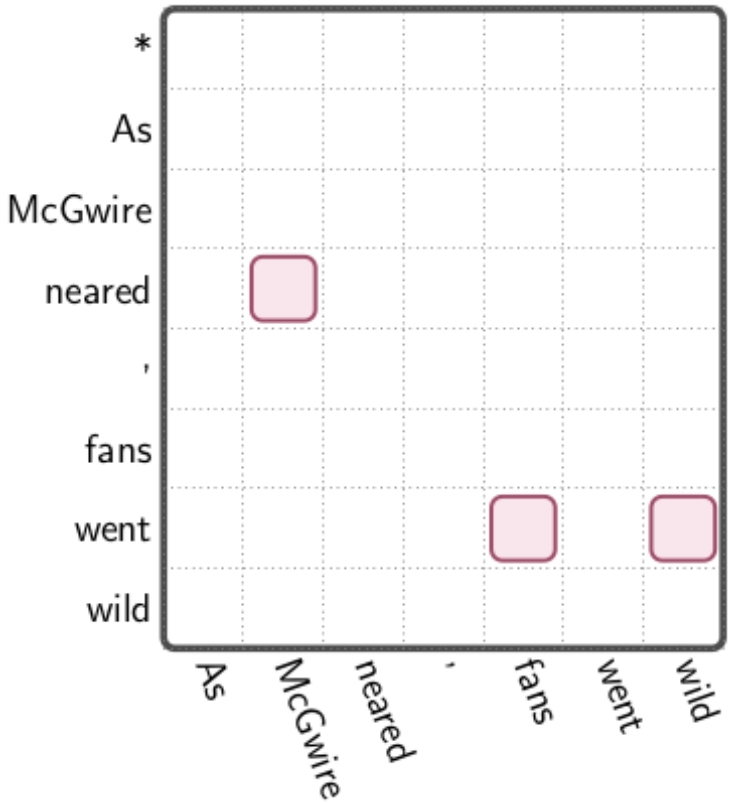
Algoritmo de Eisner: Ejemplo



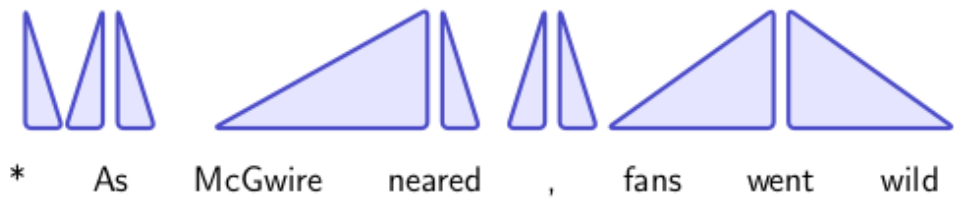


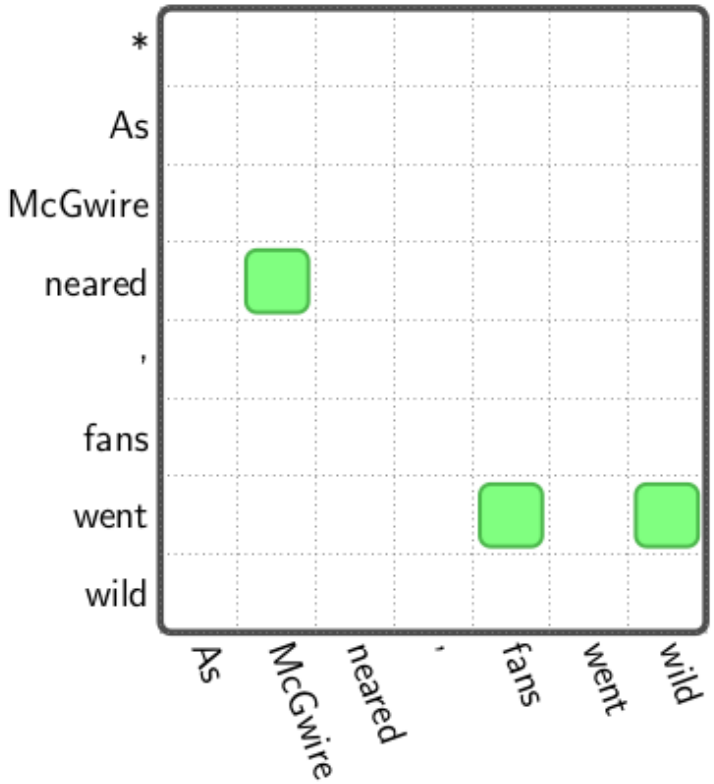
Algoritmo de Eisner: Ejemplo



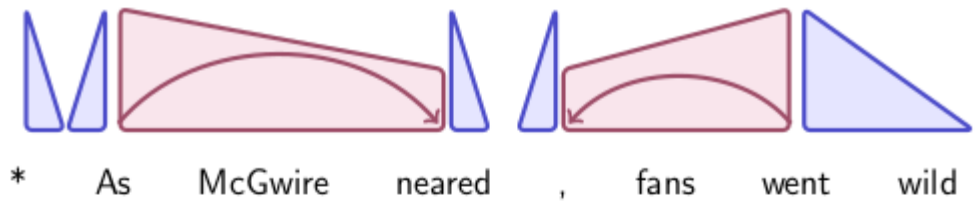


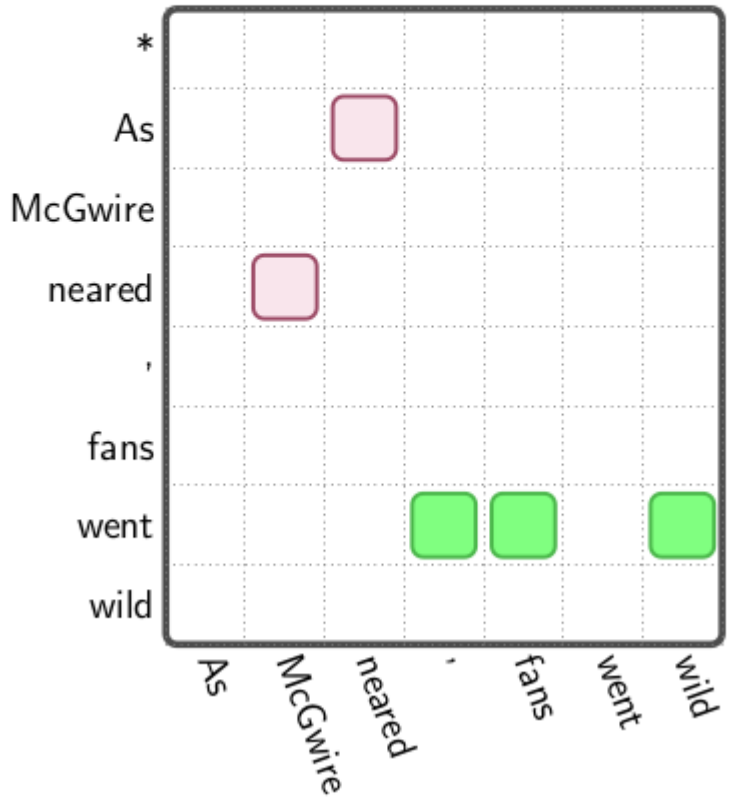
Algoritmo de Eisner: Ejemplo



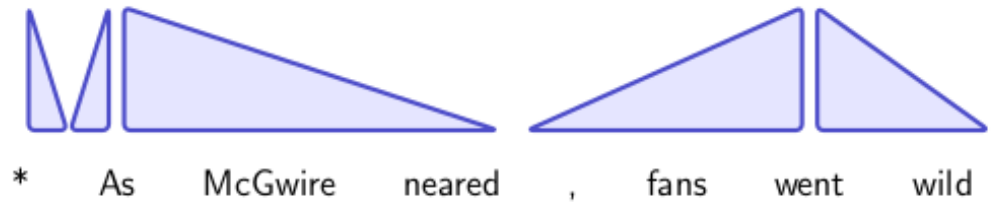


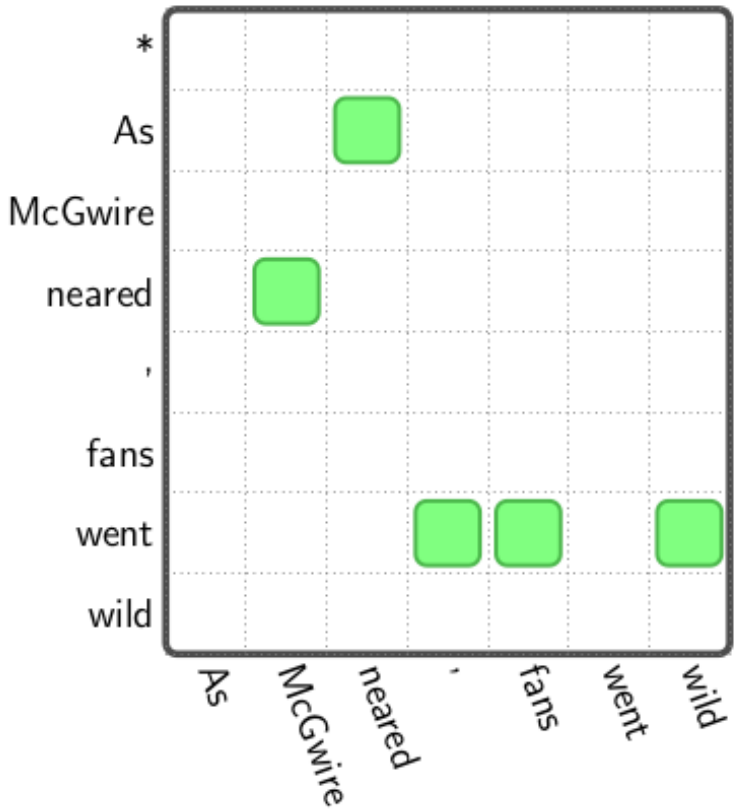
Algoritmo de Eisner: Ejemplo



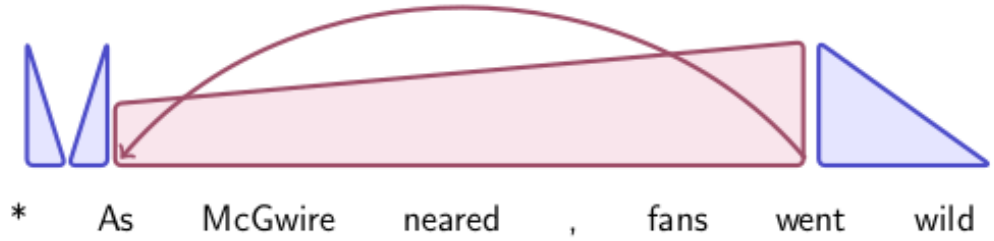


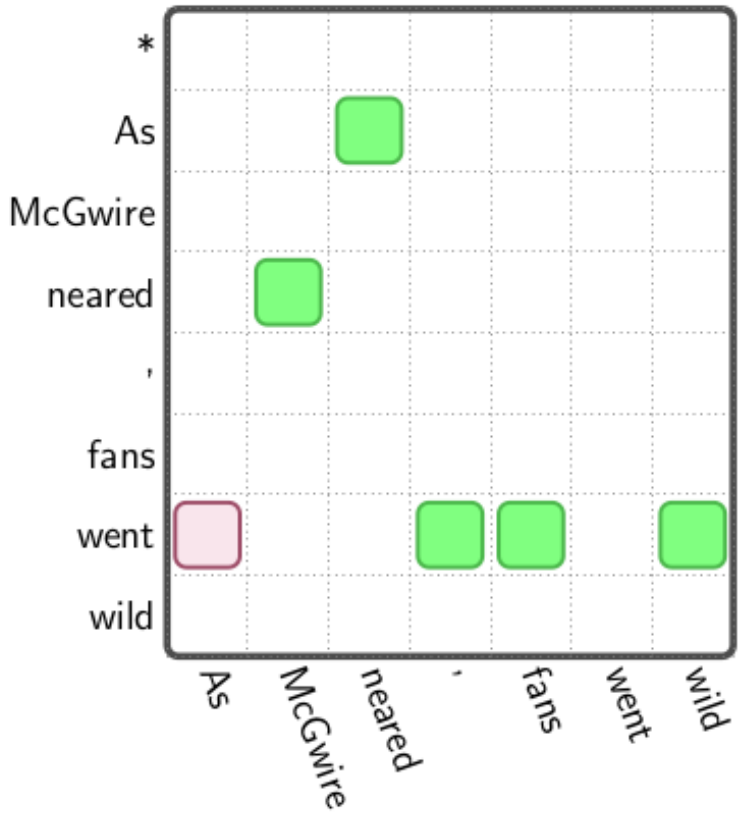
Algoritmo de Eisner: Ejemplo



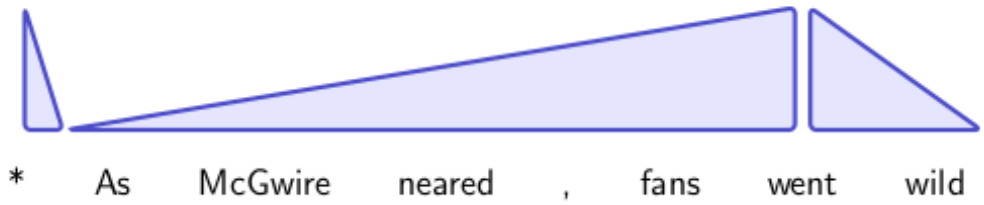


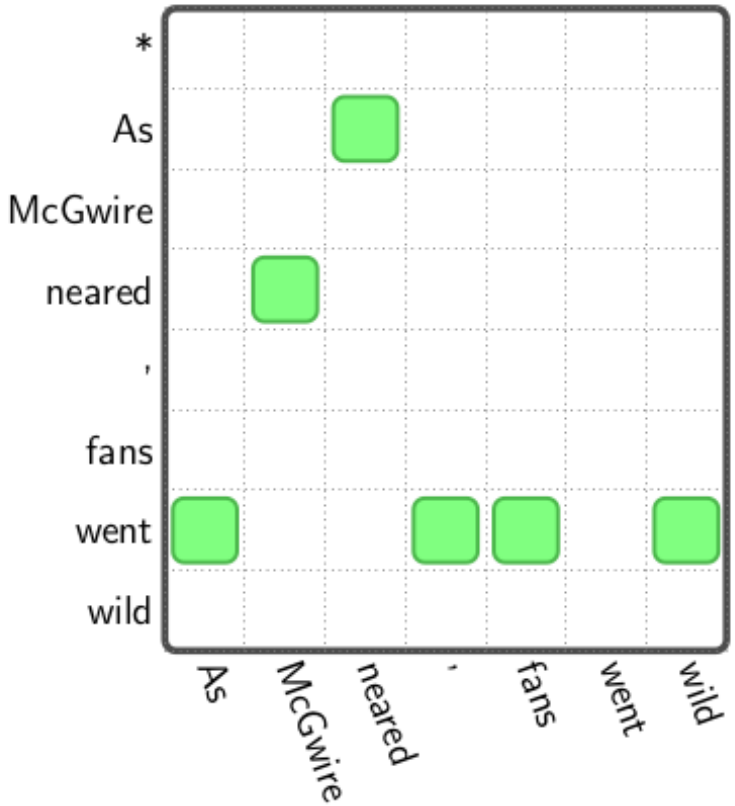
Algoritmo de Eisner: Ejemplo



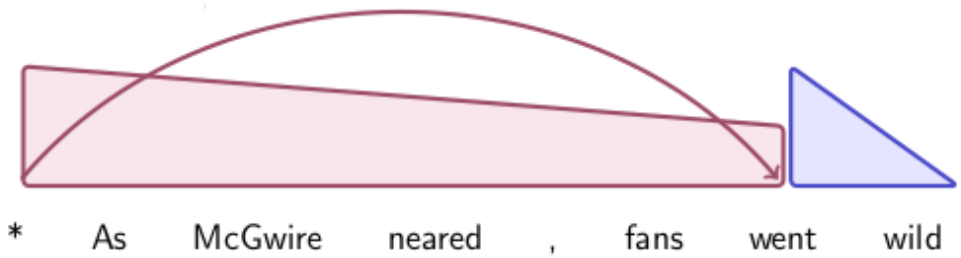


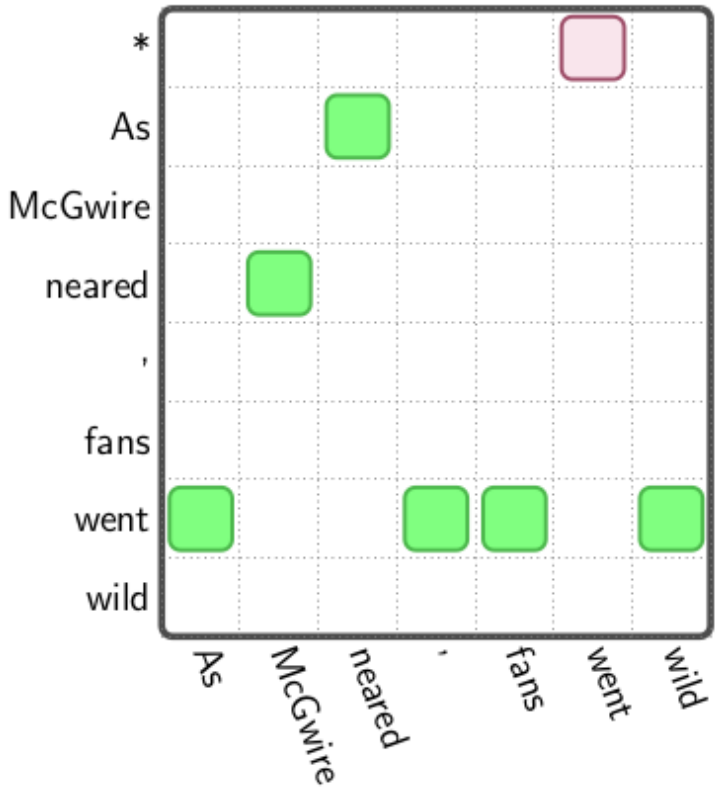
Algoritmo de Eisner: Ejemplo





Algoritmo de Eisner: Ejemplo





Algoritmo de Eisner: Ejemplo