# Aprendizaje de gramáticas probabilísticas a partir de árboles sintácticos*

**Jose L. Verdú-Mas**
Departamento de Lenguajes y Sistemas Informáticos.
Universidad de Alicante, Spain
verdu@dlsi.ua.es

**Resumen:** En este artículo se analizan varios tipos de gramáticas independientes del contexto probabilísticas obtenidas a partir de corpus etiquetados sintácticamente (*treebanks*). Estas gramáticas se utilizan para la desambiguación léxica y sintáctica de frases procedentes del lenguaje natural. Los modelos que aquí se estudian son los siguientes: (1) uno que simplemente extrae las reglas contenidas en el corpus y cuenta el número de ocurrencias de cada una; (2) un modelo que además almacena información acerca de la categoría sintáctica del nodo padre, y (3) un modelo que extrae y estima las probabilidades de las reglas almacenando información acerca de la categoría sintáctica de los hijos. Este último permite análisis sintácticos más eficientes, disminuye considerablemente la perplejidad de los conjuntos de tests y supone formalmente una generalización del concepto de $n$-gramas al caso de árboles.
**Palabras clave:** gramáticas probabilísticas de contexto libre, análisis sintáctico, *treebanks*.

**Abstract:** In this paper, we compare three different approaches to build a probabilistic context-free grammar for natural language parsing from a tree bank corpus: (1) a model that simply extracts the rules contained in the corpus and counts the number of occurrences of each rule; (2) a model that also stores information about the parent node's category, and (3) a model that estimates the probabilities according to a generalized $k$-gram scheme for trees with $k = 3$. The last model allows for faster parsing, decreases considerably the perplexity of test samples and may be seen as a generalization of the classic $n$-gram models to the case of trees.
**Keywords:** stochastic context-free grammar, parses, treebanks.

## 1. *Introducción*

Las gramáticas independientes del contexto son una forma habitualmente utilizada de representar la estructura sintáctica de las oraciones. Muchas tareas dedicadas al procesamiento del lenguaje natural (PLN) necesitan de dicha estructura para poder interpretar cada frase. Sin embargo, el problema de la ambigüedad estructural es muy común (sobre todo en frases con más de 15 palabras) y dificulta enormemente la labor. Algunos autores (p.e. (2)) establecen que la mayoría de ambigüedades sintácticas se pueden resolver sin utilizar información semántica alguna; esto es, sólo seleccionando el análisis sintáctico más probable de entre todos los candidatos. Esto establece las bases de una familia de técnicas que utilizan las probabilidades para decidir cual es el análisis sintáctico que mejor se adapta a cada frase.

Las probabilidades de cada estructura se estiman a partir de corpus de frases analizadas sintácticamente. El *Penn Treebank* (2) es un ejemplo de este tipo de corpus. La técnica más conocida que construye gramáticas probabilísticas de contexto libre (GPCL) a partir de *treebanks* es la denominada *tree bank grammars* (2). En este tipo de gramáticas, las reglas se extraen directamente de los árboles de análisis sintácticos; cada constituyente del árbol de dos niveles es una regla. Las probabilidades se estiman contando el número de veces que cada producción aparece. Este es el esquema más sencillo y no está libre de problemas. Mejores resultados se obtienen utilizando el modelo parent annotation (2) donde cada nodo del árbol almacena cierta información contextual; en este caso, la categoría sintáctica del nodo padre.

* Trabajo financiado por el proyecto de la CICyT número TIC2000-1599 y el proyecto de la Generaliat Valencia número CTIDIB/2002/173 .

Todo esto ya lo estableció Charniak (2) cuando observó que las *treebank grammars* sobregeneralizaban en exceso y que se necesitaban mecanismos, como el mencionado, para relajar la independencia de la GPCL.

Con este espíritu se ha introducido una generalización de los clásicos $n$-gramas aplicados a árboles. Las GPCL obtenidas consisten en reglas que incluyen información contextual acerca de cuando la regla puede ser aplicada. Por analogía con el modelo de Johnson (2) se le podría llamar al nuevo modelo *modelo de anotación de los hijos.*

## 2. Una generalización de los $n$-gramas

Los $n$-gramas son modelos estocásticos para la generación de secuencias $s_1, s_2, ...$ basados en probabilidades condicionales de forma que:

1. Dado un modelo $M$, la probabilidad $P(s_1 s_2 \dots s_t|M)$ de una secuencia se define como un producto de probabilidades condicionales

$$P(s_1 s_2 \dots s_t|M) = \\ = p_M(s_1)p_M(s_2|s_1)\cdots p_M(s_t|s_1 s_2 \dots s_{t-1}) \quad (1)$$

2. Las probabilidades $p_M$ se limitan a algún contexto inmediatamente anterior, en particular, a las últimas $n-1$ palabras

$$p_M(s_t|s_1 \dots s_{t-1}) = p_M(s_t|s_{t-n+1} \dots s_{t-1}) \quad (2)$$

De esta manera, los parámetros a determinar en un modelo de $n$-gramas son las probabilidades $p_M(s_n|s_1 \dots s_{n-1}) \forall s_1, \dots, s_n \in \Sigma$. El número de dichos parámetros crece exponencialmente con $n$ de manera que sólo los casos $n = 2$ (*bigramas*) y $n = 3$ (*trigramas*) se utilizan en la práctica.

Nótese que en este tipo de modelos, la probabilidad de generar en tiempo $t$ la palabra $s_t$ queda definida como una función de la subsecuencia de longitud $n-1$ que inmediatamente precede a dicha palabra. Si lo que se pretende es extenderlo a árboles, ya no está tan claro que contexto debería considerarse. Los $n$-gramas aplicados a secuencias se analizan siguiendo un orden natural: de izquierda a derecha. En el caso de árboles cabría pensar en dos formas de analizarlos:

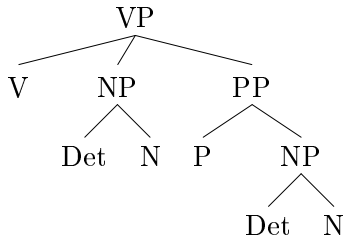


Figura 1: Un ejemplo de árbol de análisis sintático de profundidad 3.

ascendente (o *bottom-up*) o descendente (o *top-down*). La clase de lenguajes que reconocen los autómatas de árboles ascendentes es mayor (2) y, por tanto, permiten descripciones sintácticas más ricas.

Así, en nuestro modelo, la probabilidad de expansión de un nodo vendrá dada en función del subárbol de longitud $k-2$ [1] que dicho nodo genera[2], esto es, cada *estado* almacena un subárbol de profundidad $k-2$. De esta manera:

- Las probabilidades son estimadas deacuerdo a un modelo de $k$-gramas generalizado al caso de árboles.
- Las reglas gramaticales incluyen información acerca del contexto donde son aplicadas.

En el caso particular de $k = 2$ sólo se tiene en cuenta la etiqueta del nodo (análogo a los *bigramas* para secuencias). El modelo coincide con la simple extracción de reglas contenidas en el corpus, las llamadas *treebank grammars* de Charniak, (2), entre otros. Por ejemplo, para el árbol de la figura 1, se obtiene las siguientes reglas:

$$\begin{array}{lcl} \text{VP} & \rightarrow & \text{V NP PP} \\ \text{NP} & \rightarrow & \text{Det N} \\ \text{PP} & \rightarrow & \text{P NP} \end{array}$$

Sin embargo, en nuestro modelo de $k$-gramas y tomando $k = 3$, los símbolos no terminales $X_{Z_1,\dots,Z_m}$ quedan definidos por:

- la etiqueta del nodo $X$,

[1] Mientras que en el caso de secuencias, el tamaño de la ventana se denota por $n$, para el caso de árboles hemos preferido, como suele ser habitual, utilizar la variable $k$.

[2] Notar que en nuestra notación un árbol con un único nodo tiene profundidad 0. Sin embargo, una secuencia con una única palabra tiene longitud 1.

| frases procesadas | $k=3$ | PA | $k=2$ |
|---|---|---|---|
| 10 000 | 6 529 | 263 | 70 |
| 20 000 | 9 958 | 293 | 70 |
| 30 000 | 12 548 | 321 | 70 |
| 40 000 | 14 703 | 339 | 70 |
| 50 000 | 16 842 | 354 | 70 |

Cuadro 1: Tamaño del conjunto de símbolos gramaticales en función del número de frases procesadas

- el número $m$ de descendientes del nodo $X$ (si los tiene) y
- el orden de las etiquetas de los descendientes $Z_1, ..., Z_m$ (si los tiene)

De esta manera, para el mismo árbol se obtienen las siguientes reglas gramaticales:

$$\begin{aligned} \mathrm{VP}_{\mathrm{V,NP,PP}} &\rightarrow \mathrm{V\ NP}_{\mathrm{Det,N}}\ \mathrm{PP}_{\mathrm{P,NP}} \\ \mathrm{NP}_{\mathrm{Det,N}} &\rightarrow \mathrm{Det\ N} \\ \mathrm{PP}_{\mathrm{P,NP}} &\rightarrow \mathrm{P\ NP}_{\mathrm{Det,N}} \end{aligned}$$

Nótese que el proceso es equivalente a la realización de un re-etiquetado en el árbol de análisis sintáctico antes de extraer las reglas según un modelo $k = 2$.

Finalmente, en el modelo *parent-annottated* (PA) descrito en (2) los símbolos no terminales quedan definidos a partir de las etiquetas del propio nodo y la del nodo padre:

$$\begin{aligned} {}^{\mathrm{S}}\mathrm{VP} &\rightarrow \mathrm{V}\ {}^{\mathrm{VP}}\mathrm{NP}\ {}^{\mathrm{VP}}\mathrm{PP} \\ {}^{\mathrm{VP}}\mathrm{NP} &\rightarrow \mathrm{Det\ N} \\ {}^{\mathrm{VP}}\mathrm{PP} &\rightarrow \mathrm{P}\ {}^{\mathrm{PP}}\mathrm{NP} \\ {}^{\mathrm{PP}}\mathrm{NP} &\rightarrow \mathrm{Det\ N} \end{aligned}$$

Es evidente que los modelos $k = 3$ y PA incorporan información contextual que no está presente en el modelo $k = 2$ y, por lo tanto, un número mayor de reglas se van a obtener. Sin embargo, en la práctica el tamaño del conjunto de reglas siempre es moderado. Ahora bien, aunque la generalización permite valores de $k$ superiores a 3, el gran número de posibles reglas obtenidas obligaría a disponer de un corpus extremadamente grande para obtener un modelo estadísticamente aceptable.

La gráfica de la figura 2 muestra como crece el conjunto de reglas en función del número de frases procesadas. Como cabía esperar, se observa que el número de reglas obtenidas aumenta conforme más información se codifica en la etiqueta del nodo. Por otra parte, no parece alcanzar un límite incluso después de analizar todo el corpus. Similares conclusiones se pueden extraer del cuadro 1 dónde se aprecia cómo varía el tamaño del conjunto de símbolos según se procesan frases del corpus (experimentos realizados según las condiciones expresadas en la sección 3.1).

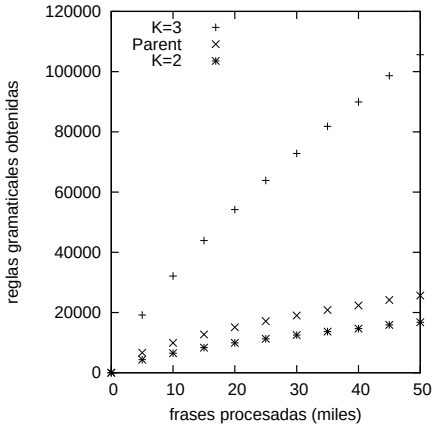


Figura 2: Número de reglas inferidas en función del número de frases procesadas

## 3. *Resultados experimentales*

### 3.1. Condiciones generales

Se han realizado experimentos para valorar la capacidad de desambiguación de estos modelos de $k$-gramas comparándolos con las *treebank grammars* y el modelo *parent-annotation* (PA) (2). En concreto, hemos analizado la habilidad de cada gramática para seleccionar el árbol sintáctico que mejor se adapta a cada oración, esto es, la habilidad para resolver el problema de la ambigüedad estructural. También hemos estudiado la perplejidad como un indicador de la bondad de cada método. El corpus utilizado, tanto para entrenamiento como para test, ha sido la porción *Wall Street Journal* del Penn Treebank (PTB), versión 3, con algunas modificaciones básicamente estructurales:

- En cada oración se ha insertado un nuevo nodo raíz (con etiqueta ROOT, como en (2)) por encima del que ya contenía. Éste será el símbolo inicial de la gramática.

- De las etiquetas de los nodos se ha eliminado todo aquello que no aporta información sintáctica de interés (prefijos y sufijos). Así, por ejemplo, la etiqueta NP-SBJ queda reducida a NP.
- Los constituyentes (subárboles) vacíos se han eliminado por no aportar tampoco información de interés. (etiquetas -NONE- del PTB que producen reglas del tipo ($X \rightarrow \lambda$)
- Se han eliminado las ramas de los árboles que producen reglas del tipo ($X \rightarrow X$). Para evitar la obtención, en algunas frases, de un número infinito de análisis posibles.

En los experimentos, el conjunto de entrenamiento estuvo formado por todos los árboles (41,532) que hay en las secciones 02 a la 22 del mencionado corpus (80 % del PTB). Esto da un total de 600,000 subárboles o constituyentes. El conjunto de test estaba compuesto de todas las frases de la sección 23 con un número de palabras no superior a 40.

### 3.2. Resultados relativos a la desambiguación estructural

Cada una de las frases del conjunto de test se analizó utilizando el algoritmo de Cocke-Younger-Kasami extendido a cualquier gramática probabilística independiente del contexto (2). El análisis sintáctico más probable se comparó con el correspondiente que figuraba en el *treebank* utilizando la métrica de evaluación PARSEVAL (2; 2, p. 432), la cual da cierto crédito a los análisis parcialmente incorrectos estableciendo, entre otras, estas dos medidas:

- *labeled precision* ($P$) Es la fracción de constituyentes etiquetados de forma correcta en el análisis más probable sobre el total de constituyentes obtenidos en dicho análisis.
- *labeled recall* ($R$) Es la fracción de constituyentes etiquetados de forma correcta en el análisis más probable sobre el total de constituyentes en el análisis del *treebank*.

Los modelos evaluados han sido los siguientes:

| MODELO | $R$ | $P$ | $f_{R=100\,\%}$ |
|---|---|---|---|
| $k$=2 | 70.7 % | 76.1 % | 10.4 % |
| $k$=3 | 79.6 % | 74.3 % | 19.9 % |
| PARENT | 80.0 % | 81.9 % | 18.5 % |
| AMBOS | 80.5 % | 74.5 % | 22.7 % |

| MODELO | EXACTAS | COBERTURA | $t$ |
|---|---|---|---|
| $k$=2 | 10.0 % | 100 % | 57 |
| $k$=3 | 13.4 % | 94.6 % | 7 |
| PARENT | 16.3 % | 100 % | 340 |
| AMBOS | 15.5 % | 79.6 % | 4 |

Cuadro 2: Resultados del análisis para cada modelo:labelled recall $R$, labelled precision $P$, fracción de frases con total labelled recall $f_{R=100\,\%}$, fracción de análisis idénticos al PTB, cobertura o fracción de frases analizadas, y tiempo medio de respuesta fpor frase en segundos.

- Una *treebank grammar* estándar, sin ningún tipo de anotación ($k$=2), con probabilidades para 15,140 reglas.
- Una gramática con anotación únicamente de la categoría sintáctica de los nodos hijos ($k$=3), con probabilidades para 92,830 reglas.
- Una gramática del tipo *parent-annotated* (PA), con probabilidades para 23,020 reglas.
- Una gramática con anotación tanto del padre como de los hijos (AMBOS), con probabilidades para 112,610 reglas.

Como ya se ha dicho, el número de reglas obtenidas crece conforme más información se codifica en cada nodo, sin embargo, este incremento no es extremo. Por otra parte, al disminuir la capacidad de sobregeneralización, algunas frases del test no se pueden analizar, esto es, la gramática las rechaza.

Los resultados del cuadro 2 muestran que:

- Los modelos con anotación de la categoría sintáctica de los nodos padre o hijos son capaces de analizar sintácticamente mejor que aquellos sin dicha anotación.
- En términos generales, los modelos con anotación de hijos se comportan de forma ligeramente peor que los de anotación del padre. Esto puede ser debido

| Modelo | $t$ (s) |
|---|---|
| $k = 3$ | 5.45 |
| PA | 229.17 |
| $k = 2$ | 45.77 |
| AMBOS | 3.60 |

Cuadro 3: Tiempo medio de análisis para frases de 25 palabras

| MODELO | Número de posibles análisis sintácticos |
|---|---|
| $k = 3$ | 68 |
| PA | 642 152 |
| $k = 2$ | 662 422 |

Cuadro 4: Número de análisis posibles que, para cada modelo, tiene la secuencia de etiquetas sintácticas: DT NN VB JJ.

a la gran cantidad de parámetros a estimar que se obtienen de los primeros comparándolo con los segundos. Los modelos de $k$-gramas con $k \geq 3$ producen reglas gramaticales muy precisas que sólo se pueden aplicar en un contexto determinado. Esto que a priori es una ventaja, se convierte en un inconveniente cuando el *treebank* del que disponemos tiene en promedio sólo 6 subárboles por regla.

- Los modelos $k = 3$ y AMBOS no son capaces de analizar todo el test. En otras palabras, rechazan algunas de las frases contenidas en él.
- El tiempo medio para analizar una frase es, con mucha diferencia, inferior si se utilizan los modelos $k = 3$ y AMBOS (donde también interviene $k = 3$). Esto es razonable teniendo en cuenta que en dichos modelos el número de posibles análisis sintácticos se reduce drásticamente.

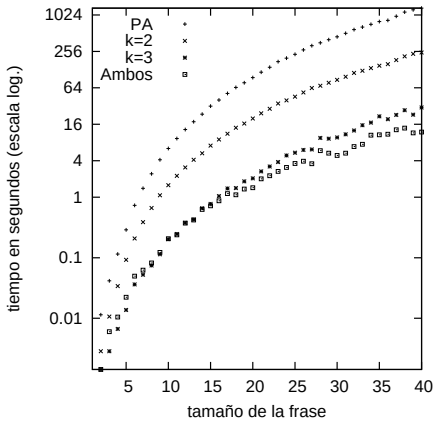


Figura 3: Tiempo medio de análisis de la frase para cada modelo.

La gráfica de la figura 3 muestra, en escala logarítmica y para cada modelo, como varía el tiempo medio de análisis de cada frase en función su tamaño (número de palabras). Se observa que efectivamente los modelos $k = 3$ y AMBOS permiten analizar de manera mucho más rápida. El cuadro 3 muestra el tiempo medio de análisis, en segundos, de las frases con 25 palabras (tamaño medio de las frases en el PTB).

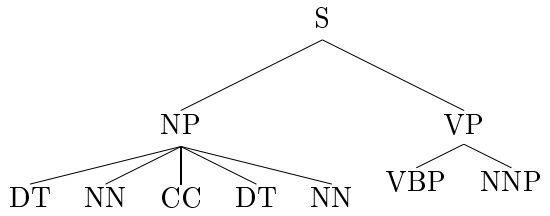


Figura 4: Un ejemplo de árbol de análisis sintatico obtenido con el modelo *parent annotation*. Coincide con el encontrado en el PTB.

Es evidente que las diferencias se deben a la ambigüedad intrínseca de cada modelo. El cuadro 4 muestra el número de análisis posibles que, con diferentes gramáticas, tiene la secuencia de etiquetas sintácticas: DT NN VB JJ. Se observa que, efectivamente, la ambigüedad del modelo $k = 3$ es muy inferior al resto.

Por otra parte, los modelos con anotación de los hijos producen árboles de análisis sintácticos mucho más estructurados y refinados que los obtenidos por los que no utilizan anotación alguna o utilizan anotación del padre, los cuales tienden a utilizar reglas que producen árboles muy planos o poco profundos. Sirvan como ejemplo las figuras 4 y 5. Muestran dos análisis sintácticos producidos, respectivamente, por un modelo con anotación del padre y otro con anotación de los hijos. Además, el primero de ellos es el que el PTB establece como *correcto*.

El ejemplo también es útil para poner de relieve las deficiencias que subyacen en las estructuras de los árboles del PTB. Son, como

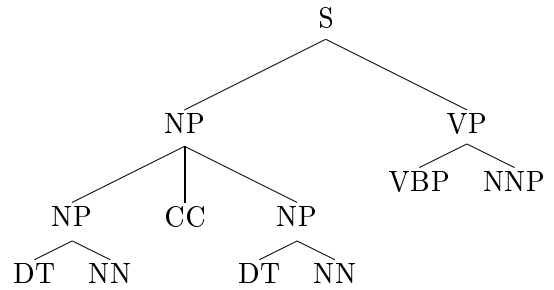


Figura 5: Un ejemplo de árbol de análisis sintático obtenido con el modelo $k = 3$

| Combinación lineal | $P_{\min}$ | $\lambda_{\min}$ |
|---|---|---|
| $k = 2$ y $k = 3$ | 90.8 | 0.25 |
| $k = 2$ y Parent | 108.7 | 0.6 |
| $k = 2$ y Ambos | 94 | 0.3 |
| $k = 3$ y Parent | 88 | 0.65 |

Cuadro 5: Valores del parámetro $\lambda_{\min}$ que minimiza la perplejidad del test para cada combinación lineal estudiada. La perplejidad mínima se obtuvo con una mezcla del modelo $k = 3$ y *parent annotation*. Todos los modelos tienen cobertura total.

se ha dicho, excesivamente planas en algunos casos como para ser sintáctica o semánticamente útiles. Esto perjudica de manera muy especial a la notación de los hijos, al menos si como métrica de evaluación se emplean cuestionadas por algunos autores (ver, p.e. (2)) PARSEVAL que asignan, en este ejemplo, 100 % a las medidas *labelled recall* y *labelled precision* del árbol de la figura 4 y, respectivamente, 100 % y 60 % a las mismas medidas del árbol de la figura 5 el cual no debería considerarse peor que el encontrado en el PTB.

### 3.3. Resultados relativos a la perplejidad

También se ha utilizado la perplejidad de un conjunto de test $S = \{w_1, ..., w_{|S|}\}$ como indicador de la bondad de cada modelo, $P = \frac{1}{|S|} \sum_{l=1}^{|S|} \log_2 p(w_l|M)$, donde $p(w_l|M)$ es la suma de las probabilidades de todos los análisis posibles de la frase $w_l$. La perplejidad mide la falta de habilidad del modelo para predecir nuevos eventos, así, mejor será el modelo si tiene una menor perplejidad. Ahora bien, puesto que algunos modelos asignan probabilidad nula a ciertas frases y, por tanto, una perplejidad infinita, se han estudiado ciertas combinaciones lineales de modelos $M_i$ y $M_j$ con $p(w_l|M_i, M_j) = \lambda p(w_l|M_i) + (1-\lambda) p(w_l|M_j)$ que garantizaban una cobertura total. El parametro $\lambda \in [0, 1]$ elegido es el que minimizaba la perplejidad.

Los mejores resultados se obtuvieron con una mezcla de los modelos $k = 3$ y *parent annotation* con una mayor presencia (65 %) del primero. Las medidas *labelled recall* y *labelled precision* fueron, respectivamente, 82.1 % y 81 %. El porcentaje de frases con total *labelled recall* $f_{R=100\,\%}$ alcanzó el 22.2 %, similar al mejor resultado del cuadro 2 pero cubriendo todo el conjunto de test.

Los valores de la perplejidad $P_{\min}$ y el correspondiente valor de $\lambda$ con el que se obtuvieron se muestran en el cuadro 5.

## 4. Conclusiones

En este artículo se ha introducido un nuevo modelo de gramática probabilística de contexto libre en la que sus variables se especializan almacenando información del contexto en el que aparecen. Esto es, tienen en cuenta el árbol que generan hasta un cierto nivel. En particular, se han estudiado modelos en los que los nodos guardan la categoría sintáctica de los hijos. Se han comparado con aquellos que no almacenan información adicional alguna (*treebank grammars*) y aquellos que almacenan en cada nodo la categoría sintáctica del nodo padre (*parent annotated models*).

Por otra parte, los modelos introducidos suponen una generalización de los clásicos $n$-gramas aplicados a árboles. Los experimentos han mostrado que:

- La habilidad de los modelos $k = 3$ y PA para resolver el problema de la ambigüedad estructural es similar.
- El análisis sintáctico de las frases resulta, con diferencia, mucho más rápido si se utilizan gramáticas con anotación de los hijos ($k = 3$). Esto es debido a que en estos modelos el número de posibles análisis se reduce drásticamente. Sin embargo, no es este el caso del modelo PA donde por el contrario, la ambigüedad aumenta.
- Las gramáticas del tipo $k = 3$ y superiores serían más efectivas si se dispusiera de un corpus lo suficientemente grande como para poder estimar de forma correcta todos sus parámetros.

- Las gramáticas con anotación de los descendientes tienden a dar análisis sintácticos mucho más estructurados y refinados que las otras.
- La perplejidad de los conjuntos de test se reduce siempre que la anotación de los hijos esté presente en el modelo.

Actualmente se están desarrollando nuevas técnicas de suavizado que consisten en la incorporación de nuevas reglas gramaticales para permitir el uso del *back off* desde valores superiores de $k$ hasta $k = 2$ (con cobertura total). Al mismo tiempo, permitirán recuperar el valor de $k$ original (notar que cuanto mayor es $k$ más rápido es el análisis sintáctico). En otras palabras, se trata de un *back off* a nivel de regla gramatical con capacidad de recuperación. Por otra parte y con el objetivo de que el análisis sea lo más eficiente posible, se está desarrollando un nuevo algoritmo de análisis sintáctico orientado a este tipo de gramáticas con el suavizado mencionado.

### *Bibliografía*


L. Frazier and K. Rayner. Making and correcting errors during sentence comprehension: Eye movements in the analysis of structurally ambiguous sentences. *Cognitive Psychology*, 14:178–210, 1982.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19:313–330, 1993.

Maurice Nivat and Andreas Podelski. Minimal ascending and descending tree automata. *SIAM Journal on Computing*, 26(1):39–58, 1997.

Eugene Charniak. Treebank grammars. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1031–1036. AAAI Press/MIT Press, 1996.

Mark Johnson. PCFG models of linguistic tree representations. *Computational Linguistics*, 24(4):613–632, 1998.

J.-C. Chappelier and M. Rajman. A generalized CYK algorithm for parsing stochastic CFG. In *Actes de TAPD'98*, pages 133–137, 1998.

Ezra Black, Steven Abney, Dan Flickinger, Claudia Gdaniec, Ralph Grishman, Philip Harrison, Donald Hindle, Robert Ingria, Frederick Jelinek, Judith Klavans, Mark Liberman, Mitch Marcus, Salim Roukos, Beatrice Santorini, and Tomek Strzalkowski. A procedure for quantitatively comparing the syntatic coverage of english grammars. In *Proc. Speech and Natural Language Workshop 1991*, pages 306–311, San Mateo, CA, 1991. Morgan Kauffmann.

Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing.* MIT Press, 1999.

John Carroll, Ted Briscoe, and Antonio Sanfilippo. Parser evaluation: A survey and a new proposal. In *Proceedings of the International Conference on Language REsources and Evaluation*, pages 447–454, Granada, Spain, 1998.