Podado y lexicalización de reglas gramaticales y su aplicación al análisis sintáctico parcial*

Empar Bisbal, Antonio Molina, Lidia Moreno
Universitat Politècnica de València
Camí de Vera s/n, València, Spain
{ebisbal, amolina, lmoreno}@dsic.upv.es

Resumen: En este artículo se presenta un mecanismo de adquisición automática de reglas a partir de corpora anotados sintácticamente. Esta aproximación se basa en una extensión del algoritmo propuesto por Claire Cardie y David Pierce (Cardie y Pierce, 1998). Se muestra cómo mejora el modelo aplicando técnicas de lexicalización (Molina, 2004).
Palabras clave: Técnicas basadas en corpus, Análisis sintáctico superficial

Abstract: In this paper we present a mechanism for the automatic acquisition of rules from syntactic anotated corpora. This aproach is based on an extension of the Claire Cardie and David Pierce algorithm (Cardie y Pierce, 1998). The improvement of the model using lexicalization techniques (Molina, 2004) is shown.
Keywords: Corpus-based techniques, Shallow parsing

1. Introducción

Actualmente, uno de los objetivos principales del procesamiento del lenguaje natural (PLN) es el análisis de *textos no restringidos*. Este tipo de análisis es necesario en multitud de aplicaciones como la traducción automática, la comprensión de textos transcritos, la extracción de información o la búsqueda de respuestas, entre otras. Las prestaciones de estas aplicaciones pueden mejorarse si se incorpora un módulo de PLN que realice el análisis sintáctico de los textos manejados por la aplicación. Sin embargo, la realización del análisis completo de estos textos presenta problemas bien conocidos, como son: la definición de una gramática con la suficiente cobertura del lenguaje, la resolución de la ambigüedad estructural, el coste de definición de la gramática por expertos lingüistas o el coste computacional de un análisis completo que puede ser prohibitivo en algunas aplicaciones. Por otro lado, no todas las aplicaciones requieren un análisis completo de la oración. Por ejemplo, en extracción de información o en búsqueda de respuestas, la identificación de sintagmas nominales o verbales sencillos y algunas relaciones entre éstos puede mejorar las prestaciones de dichas aplicaciones.

En este sentido el análisis sintáctico parcial o superficial se ha mostrado como una técnica útil tanto en dominios del lenguaje hablado como escrito, y como una alternativa al análisis completo en aplicaciones que necesitan analizar grandes cantidades de texto y que, por razones computacionales o de robustez, no puedan procesarse con un analizador completo. El análisis parcial (Abney, 1997) consiste en dividir un texto en segmentos no solapados que se corresponden con constituyentes sintácticos no recursivos [1] (o *chunks*), con el objetivo de extraer información sintáctica de manera fiable, robusta y eficiente.

Al igual que otras tareas de desambiguación en PLN, el análisis sintáctico parcial, puede abordarse desde dos perspectivas diferentes: mediante técnicas deductivas basadas en el conocimiento y mediante técnicas inductivas o de aprendizaje automático basadas en corpus.

Las técnicas deductivas definen un conjunto de reglas gramaticales que representan la estructura sintáctica de la lengua objeto de análisis mediante algún formalismo gramatical y aplican algún método de análisis que permite procesar textos no restringidos de manera robusta. La mayoría de estos trabajos se basan en técnicas de estados finitos, definiendo los constituyentes sintácticos me-

* Este trabajo ha sido subvencionado por los proyectos PROFIT 3LB (FIT-150500-2002-244) y CICYT R2D2 (TIC2003-07158-C04-03)

[1] Un constituyente es *no recursivo* si no contiene en su interior ese mismo constituyente

diante patrones o expresiones regulares. De esta manera se han desarrollado analizadores parciales para diversas lenguas como el inglés (Abney, 1996), el francés (Aït-Mokhtar y Chanod, 1997) o el castellano (Gala, 1999).

Las técnicas inductivas construyen de manera automática una representación del conocimiento sintáctico a partir de ejemplos anotados con información sintáctica (*treebanks*). Entre las aproximaciones inductivas al análisis parcial más relevantes se encuentran métodos de aprendizaje basado en ejemplos (Veenstra y Van den Bosch, 2000), métodos estadísticos (Koeling, 2000) (Molina y Pla, 2002), aprendizaje de reglas (Déjean, 2000), soporte vectorial (Kudo y Matsumoto, 2001), redes neuronales (Carreras y Màrquez, 2003) o métodos combinados (Tjong Kim Sang, 2002). El éxito de estos métodos radica fundamentalmente en la determinación de las características relevantes para el proceso de desambiguación. La mayoría consideran como características relevantes las palabras, las etiquetas morfosintácticas y las etiquetas de *chunk*, en un contexto de dos o tres posiciones alrededor de la palabra foco.

En este trabajo se presenta una generalización de la técnica de aprendizaje propuesta por Claire Cardie y David Pierce (Cardie y Pierce, 1998) que consiste en la aplicación de un algoritmo iterativo que permite el aprendizaje automático de reglas correspondientes a sintagmas nominales básicos (NPs) a partir de un corpus anotado sintácticamente. Este trabajo extiende dicha técnica en dos sentidos: por un lado, se generaliza el algoritmo para permitir la identificación de otros constituyentes no recursivos, además de los NPs. Por otro lado, se introducen una técnica de lexicalización de la gramática que permite mejorar las prestaciones del analizador. El analizador propuesto se ha evaluado sobre la tarea compartida, para el inglés, desarrollada en la edición de 2000 de la conferencia CoNLL, que se ha convertido en el punto de referencia para comparar distintas aproximaciones al análisis parcial.

En la sección 2 de este artículo se describen las características de la tarea de análisis parcial abordada y del corpus utilizado. En la sección 3 se describe la técnica de aprendizaje propuesta por Cardie y Pierce. La generalización del algoritmo para identificar distintos constituyentes se presenta en la sección 4 resultados experimentales se muestran en la sección 5. En la sección 6 se describe la técnica de lexicalización. Finalmente, se exponen la conclusiones y líneas de trabajo futuras.

2. Descripción de la tarea

En CoNLL00 se planteó como tarea compartida el análisis superficial o *chunking*. El objetivo de esta tarea era la obtención de métodos de aprendizaje automático que, tras una fase de entrenamiento, pudieran reconocer la segmentación en *chunks* de los datos de test lo mejor posible. Por ejemplo, la oración *Rockwell said the agreement calls for it to supply 200 additional so-called shipsets for the planes* se podría dividir de la siguiente manera:

[NP Rockwell/NNP] [VP said/VBD] [NP the/DT agreement/NN] [VP calls/VBZ] [SBAR for/IN] [NP it/PRP] [VP to/TO supply/VB] [NP 200/CD additional/JJ so-called/JJ shipsets/NNS] [PP for/IN] [NP the/DT planes/NNS]

Figura 1: Ejemplo de análisis superficial.

Los etiquetadores se evaluaban mediante el indicador $F_{\beta=1}$, que combina precisión (P) y cobertura (C), aplicado a todos los chunks:

$$F_{\beta=1} = \frac{2*precision*cobertura}{cobertura+precision}$$

Se emplearon para entrenamiento y test unas particiones del corpus Wall Street Journal (WSJ) que han sido ampliamente utilizadas para *NP chunking*: las secciones 15-18 para entrenamiento (211.727 tokens) y la sección 20 para test (47.377 tokens). La anotación de los datos se llevó a cabo mediante el script *chunklink*[2], escrito por Sabine Buchholz, de la Universidad de Tilburg.

Los datos se presentaron en un formato de tres columnas. En la primera columna estaba la palabra actual, en la segunda su etiqueta morfosintáctica (PoS - *Part of Speech*) y en la tercera su etiqueta de chunk siguiendo el formato IOB2 [3]. En el cuadro 1 se muestran los chunks existentes en el corpus WSJ junto con su frecuencia de aparición en las particiones que forman el corpus de entrenamiento.

[2]Disponible en http://ilk.kub.nl/~sabine/chunklink

[3]Este esquema de anotación, propuesto por Ramshaw y Marcus en 1995 (Ramshaw y Marcus, 1995), etiqueta la primera palabra dentro de un sintagma básico X como B-X, y las siguientes palabras del sintagma como I-X. Si una palabra no pertenece a ningún sintagma se le asigna la etiqueta O.

Tipo de constituyente	Cantidad
Noun Phrase (NP)	55,081 (51 %)
Verb Phrase (VP)	21,467 (20 %)
Prepositional Phrase (PP)	21,281 (20 %)
Adverb Phrase (ADVP)	4,227 (4 %)
Subordinated clause (SBAR)	2,207 (2 %)
Adjective Phrase (ADJP)	2,060 (2 %)
Particles (PRT)	556 (1 %)
Conjunction Phrase (CONJP)	56 (0 %)
Interjection (INTJ)	31 (0 %)
List marker (LST)	10 (0 %)
Unlike Coordinated Phrase (UCP)	2 (0 %)

Cuadro 1: Cantidad de cada *chunk* en el corpus de entrenamiento (particiones 15 a 18 del WSJ).

En este trabajo se han empleado los mismos corpus de entrenamiento y test para que los resultados obtenidos sean comparables con otras aproximaciones.

3. Algoritmo de extracción de reglas

Claire Cardie y David Pierce plantearon en (Cardie y Pierce, 1998) un algoritmo para la detección de sintagmas nominales no recursivos basado en reglas. Estas reglas son extraídas de forma automática a partir de corpora anotados morfológicamente, mediante el proceso iterativo de análisis, evaluación y poda que se describe a continuación:

1. Obtención de la gramática inicial. Mediante un recorrido secuencial de una parte del corpus (denominada corpus de entrenamiento), se extrae una regla para cada NP, formada por sus etiquetas PoS. Al finalizar, se eliminan las reglas duplicadas.

2. Obtención de los sintagmas de referencia. Se extraen del corpus de entrenamiento todos los NPs anotados. Estos constituyentes sirven más adelante como referencia para la evaluación de las reglas.

3. Análisis del corpus de poda. Se analiza, mediante la heurística de la regla más larga, un corpus diferente al de entrenamiento, al que denomina *corpus de poda*, y se anotan tanto los sintagmas, como las reglas que los detectan.

4. Evaluación de las reglas. Se puntúa cada regla en función de su utilidad en la identificación de sintagmas nominales. El beneficio de una regla r viene dado por la fórmula $B_r = C_r - E_r$, siendo C_r el número de NPs identificados correctamente por r y E_r el número de errores de los que es responsable. Se considera que r es responsable de un error si es la primera regla en detectar parte de un sintagma.

5. Poda. Se proponen dos métodos: (a) podar las reglas cuya puntuación no supere un determinado umbral y (b) podar las N peores reglas.

6. Volver al paso 3.

El proceso termina cuando todas las reglas reciben una puntuación mayor o igual que el umbral en el caso (a) o bien cuando comienza a decrementarse la precisión del conjunto de reglas en el (b).

4. Desarrollo de la propuesta extendida

En las siguientes secciones se expone detalladamente cómo se ha extendido este algoritmo para que, siguiendo los mismos pasos, se puedan extraer las reglas correspondientes, no sólo a los NPs sino a todos los constituyentes presentes en el WSJ y los resultados que se ha obtenido con esta extensión.

4.1. Obtención de la gramática inicial

Para la obtención de la gramática inicial, se recorre de forma secuencial el corpus de entrenamiento, y se interpretan los códigos IOB2 para extraer las reglas que dan lugar a cada sintagma.

Las reglas obtenidas a partir de la oración de la figura 1 serían:

NP → NNP
VP → VBD
NP → DT NN
VP → VBZ
SBAR → IN
NP → PRP
VP → TO VB
NP → CD JJ JJ NNS
PP → IN
NP → DT NNS

4.2. Obtención de los sintagmas de referencia

En esta fase se toma el corpus de poda y se extraen todos los sintagmas anotados. La evaluación posterior de las reglas se realizará comparando los sintagmas extraídos por el analizador con los de referencia, como se explica en el punto 4.4.

4.3. Análisis del corpus de poda

Para esta fase, se ha implementado un analizador basado en la heurística de la regla más larga.

1. $i = 0$
2. Inicializa el conjunto de reglas disponibles a todas las reglas de la gramática.
3. Mientras el conjunto de reglas disponibles no esté vacío:
   - 3.1. Toma i siguientes palabras (a partir de la actual) y sus PoS.
   - 3.2. Busca una regla que coincida **exactamente** con la secuencia de etiquetas.
   - 3.3. Elimina del conjunto de reglas disponibles todas aquéllas que no comiencen por la secuencia de PoS.
   - 3.4. $i = i + 1$
4. Si se ha encontrado alguna regla que se haya emparejado, se anota el sintagma correspondiente, se guarda en otro fichero la regla que lo detectó y se vuelve al paso 1 adelantando i palabras.
5. En caso contrario, se vuelve al paso 1 con la siguiente palabra.

En la figura 2 se muestra el análisis de la primera oración de la partición 20 del WSJ realizado en la primera iteración del algoritmo.

[ADJP Rockwell] [ADJP said] [NP the agreement] [VP calls] [SBAR for] [NP it] [VP to supply] [NP 200 additional so-called shipsets] [SBAR for] [NP the planes]

Figura 2: Análisis de una oración tras la primera iteración del algoritmo

4.4. Evaluación y Poda

La evaluación de las reglas se realiza de la siguiente manera: si un sintagma ha sido anotado correctamente, se le suma 1 a la puntuación de la regla responsable; si es incorrecto, y la regla ha sido la primera en errar la anotación, se le resta 1; pero si, aunque la regla no haya anotado bien, ésta no ha sido la primera en fallar, no se modifica su puntuación, puesto que se supone que ha fallado en su anotación como consecuencia de un error anterior.

Por ejemplo, supongamos el fragmento presentado en la figura 1 etiquetado, tras la primera iteración, como se muestra en la figura 2.

La regla NP → CD JJ JJ NNS, que detecta el NP, sumaría 1 a su puntuación porque lo hace bien; a la de la regla SBAR → DT IN se le restaría 1 porque ha anotado un SBAR incorrecto y a la puntuación de NP → DT NNS se le sumaría 1 puesto que anota correctamente (ver sintagmas de referencia en la figura 1).

La poda ha sido implementada mediante un *umbral de poda*, es decir, en cada iteración se eliminan todas aquellas reglas cuya puntuación no supere un determinado valor (caso (a)). Las reglas no evaluadas se mantienen hasta el final y se eliminan tras la última iteración. De esta manera se reduce drásticamente el número de reglas de la gramática final sin que las prestaciones resulten afectadas de forma significativa como se muestra en la parte experimental (sección 5).

4.5. Condición de parada

El proceso debía finalizar cuando no se podara ninguna regla de una iteración a la siguiente, pero al realizar los primeros experimentos, se pudo observar que en la penúltima iteración los resultados eran siempre sensiblemente mejores que los de la última. Este hecho hizo que se planteara una nueva condición de parada: finalizar el proceso cuando entre dos iteraciones consecutivas no se eliminaran más de m reglas. La nueva condición propuesta se podría ver como una extensión directa de la planteada por Cardie, que sería el equivalente a hacer $m = 0$.

5. Resultados Experimentales

Se han empleado los corpus de entrenamiento y test comentados en la sección 2 (las secciones 15-18 y 20 del WSJ, respectivamente). Como corpus de poda se ha utilizado la partición 10.

Los resultados sobre el test se han comparado con la anotación supervisada disponible

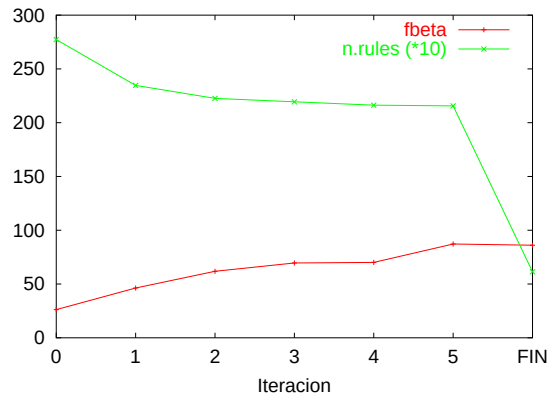

Figura 3: Evolución de $F_{\beta=1}$ y del número de reglas a lo largo de la ejecución del algoritmo con poda 0

en el WSJ en base al $F_{\beta=1}$ del conjunto de chunks, empleando para ello el comparador **conlleval**[4] que se utilizó en la competición CoNLL00.

Se han probado diferentes umbrales de poda y valores de m, obteniendo los mejores resultados con poda 0 y $m = 10$. El análisis ha sido realizado sobre el corpus de test con las gramáticas obtenidas en cada iteración, para comprobar si la poda de reglas suponía una pérdida de precisión o de cobertura y se ha añadido una iteración más, a la que hemos llamado *FIN* que es la resultante de eliminar las reglas no evaluadas de la gramática obtenida. La figura 3 muestra la evolución del número de reglas de la gramática resultante y del parámetro $F_{\beta=1}$ a medida que avanza el proceso. Es esta gráfica se puede observar cómo va disminuyendo el número de reglas de la gramática a medida que avanza el algoritmo y cómo se va incrementando el $F_{\beta=1}$ debido principalmente a que se eliminan reglas que probablemente habían sido extraídas a partir de secuencias poco frecuentes o mal etiquetadas. El descenso en la curva del número de reglas que se produce en la última etapa (FIN) se debe a que se eliminan de la gramática las reglas no evaluadas. Como se ha comentado anteriormente y se puede observar en la gráfica, esta decisión no afecta significativamente al $F_{\beta=1}$.

6. *Lexicalización*

En esta aproximación la relación entre las palabras de una oración no puede ser captu-

[4]Disponible en http://cnts.uia.ac.be/conll2000/

rada, ya que este modelo representa únicamente las relaciones existentes entre las categorías morfosintácticas. Esto significa que ciertas relaciones relevantes entre palabras, o entre palabras y etiquetas, no son modelizadas. Mediante técnicas de lexicalización se puede incorporar esta información.

Debido a que una lexicalización total del modelo incrementaría excesivamente el número de reglas resultante, se ha optado por lexicalizar con grupos de palabras que a priori puedan ser relevantes para la tarea. Para ello, se han escogido los criterios de lexicalización determinados por (Molina, 2004) para la lexicalización de modelos de Markov. Estos conjuntos de palabras también se determinaron sobre los mismos conjuntos de entrenamiento y de prueba:

Criterio WCC. Palabras que aparecen etiquetadas en el corpus WSJ con categorías gramaticales cerradas: CC, DT, EX, IN, MD, POS, PDT, PP$, PRP, RP, TO, WDT, WP, WP$ (un total de 154 palabras). Se prueba con estas palabras porque representan un conjunto de palabras bastante común y que pueden aparecer en cualquier corpus.

Criterio WCH. Palabras pertenecientes a determinados *chunks* y para las cuales existen casos de ambigüedad que no se pueden resolver si solamente se considera la etiqueta morfosintáctica. Se ha probado con palabras pertenecientes a los *chunks* SBAR, PP y VP (257 palabras).

Criterio WHF. Palabras que superan un cierto umbral de frecuencia en el conjunto de entrenamiento, en concreto, palabras cuya frecuencia es superior a 200, lo que supone un total de 88 palabras. La mayoría de estas palabras también se usan habitualmente en cualquier corpus.

Criterio WTE. Palabras difíciles de clasificar correctamente, si se considera únicamente la etiqueta morfosintáctica asociada. En concreto, se escogen aquellas cuya frecuencia de error de etiquetado ha sido mayor que 2 sobre el corpus de prueba (solamente 38 palabras).

Como se puede observar en el cuadro 2, el menor $F_{\beta=1}$ se produce cuando no se aplica lexicalización, por contra, el conjunto de reglas final es también el menor. Este conjunto

Experimento	$F_{\beta=1}$	N. Reglas
Sin Lexicalización	86.08	614
Lexicalización WCC	87.50	932
Lexicalización WHF	85.42	1092
Lexicalización WTE	87.91	797
Lexicalización WCH	86.88	1211

Cuadro 2: Resultados obtenidos con la partición 20 del WSJ en la iteración 0

de reglas sintácticas se incrementa con la lexicalización aunque no depende del número de palabras lexicalizadas. Hay que destacar que este valor tampoco influye en las prestaciones obtenidas puesto que el resultado con mayor $F_{\beta=1}$ coincide con el criterio WTE, que consta de sólo 38 palabras.

Los sistemas de aprendizaje que han obtenido los mejores resultados sobre la tarea CoNLL00 alcanzan un $F_{\beta=1}$ entre 92-94. La aproximación aqui presentada queda lejos de esos resultados. Si la comparamos únicamente con las aproximaciones basadas en reglas ((Déjean, 2000), (Johansson, 2000), (Vilain y Day, 2000)), como se puede observar en el cuadro 3, se mejoran ligeramente los resultados de algunas de ellas.

	P	**C**	$F_{\beta=1}$
Baseline	72.58 %	82.14 %	77.07
Déjean	91.87 %	92.31 %	92.09
LexWTE	**86.00 %**	**89.90 %**	**87.91**
Johansson	86.24 %	88.25 %	87.23
Vilain	88.82 %	82.91 %	85.76

Cuadro 3: Resultados de CoNLL00

7. Conclusiones y líneas futuras

En este trabajo se ha presentado un mecanismo de adquisición automática de reglas a partir de corpora anotados sintácticamente. La aproximación desarrollada permite la detección de cualquier sintagma no recursivo. Para tratar de aumentar las prestaciones se ha experimentado aplicando técnicas de lexicalización, con las que se han obtenido mejores resultados.

Actualmente se están estudiando dos nuevas vías de ampliación: por un lado, la incorporación de información estadística (n-gramas de categorías, frecuencia de reglas, frecuencia de chunks, etc.) y por otro, la posibilidad de reconocer sintagmas recursivos realizando un análisis incremental.

Además, se aplicará este proceso a otras lenguas como el español y el catalán a partir de los corpora anotados Cast3LB y Cat3LB (Aduriz et al., 2003).

Bibliografía

Abney, S. 1996. Partial Parsing via Finite-State Cascades. En *Proceedings of the ESSLLI'96 Robust Parsing Workshop*, Prague, Czech Republic.

Abney, Steven. 1997. *Part-of-Speech Tagging and Partial Parsing.* S. Young and G. Bloothooft (eds.) Corpus-Based Methods in Language and Speech Processing. An ELSNET book. Kluwer Academic Publishers, Dordrecht.

Aduriz, I., A. Ageno, B. Arrieta, J.M. Arriola, E. Bisbal, N. Castell, M. Civit, A. Díaz de Ilarraza, B. Fernández, K. Gojenola, R. Halkoum, R. Marcos, L. Márquez, M.A. Martí, P. Martínez-Barco, y A. Molina. 2003. 3lb: Construcción de una base de datos de árboles sintáctico-semánticos. *Revista para el Procesamiento del Lenguaje Natural (ISSN: 1135-5948)*, páginas 297–298, Septiembre.

Aït-Mokhtar, S. y J.P. Chanod. 1997. Incremental Finite-State Parsing. En *Proceedings of the 5th Conference on Applied Natural Language Processing*, Washington D.C., USA.

Cardie, C. y D. Pierce. 1998. Error-Driven Prunning of Treebank Grammars for Base Noun Phrase Identification. En *Proceedings of the joint 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics, COLING-ACL*, páginas 218–224, Montréal, Canada, August. http://xxx.lanl.gov/ps/cmp–lg/9808015.

Carreras, Xavier y Luís Màrquez. 2003. Phrase recognition by filtering and ranking with perceptrons. En *Proceedings of the 4th RANLP Conference*, Borovets, Bulgaria, September.

Déjean, Hervé. 2000. Learning Syntactic Structures with XML. En *Proceedings of CoNLL-2000 and LLL-2000*, Lisbon, Portugal, September.

Gala, Núria. 1999. Using the Incremental Finite-State Architecture to create a Spanish Shallow Parser. *Procesamiento del Lenguaje Natural*, 25:75–82, September. Also in Proceedings of the 14th Conferencia de la Sociedad Española para el Procesamiento del Lenguaje Natural.

Johansson, Christer. 2000. A Context Sensitive Maximum Likelihood Approach to Chunking. En *Proceedings of CoNLL-2000 and LLL-2000*, Lisbon, Portugal, September.

Koeling, Rob. 2000. Chunking with Maximum Entropy Models. En *Proceedings of CoNLL-2000 and LLL-2000*, Lisbon, Portugal, September.

Kudo, Taku y Yuji Matsumoto. 2001. Chunking with Support Vector Machines. En *Proceedings of NAACL 2001*, Pittsburgh, USA. Morgan Kaufman Publishers. http://cactus.aist-nara.ac.jp/~taku-ku/publications/naacl2001.ps.

Molina, A. 2004. *Desambiguación en procesamiento del lenguaje natural mediante técnicas de aprendizaje automático.* Tesis doctoral en informática, Departamento de Sistemas Informáticos y Computación, Universitat Politècnica de València.

Molina, Antonio y Ferran Pla. 2002. Shallow Parsing using Specialized HMMs. *Journal of Machine Learning Research*, 2:595–613.

Ramshaw, L. y M. Marcus. 1995. Text Chunking Using Transformation-Based Learning. En *Proceedings of third Workshop on Very Large Corpora*, páginas 82–94, June.

Tjong Kim Sang, Erik F. 2002. Memory-based shallow parsing. *Journal of Machine Learning Research*, 2:559–594.

Veenstra, Jorn y Antal Van den Bosch. 2000. Single-Classifier Memory-Based Phrase Chunking. En *Proceedings of CoNLL-2000 and LLL-2000*, Lisbon, Portugal, September.

Vilain, Marc y David Day. 2000. Phrase Parsing with Rule Sequence Processors: an Application to the Shared CoNLL Task. En *Proceedings of CoNLL-2000 and LLL-2000*, Lisbon, Portugal, September.