

Tratamiento de ciertos pronombres y conjunciones en la transformación de un corpus de constituyentes a un corpus de dependencias

Alexander Gelbukh, Sulema Torres

Centro de Investigación en Computación, Instituto Politécnico Nacional

Av. Juan Dios Batiz s/n col. Zacatenco, 07738, México, DF.

www.gelbukh.com, sulema7@gmail.com

Resumen

En un trabajo previo hemos presentado una técnica heurística para convertir un corpus anotado sintácticamente dentro del formalismo de constituyentes, a un corpus anotado dentro del formalismo de dependencias. El método consiste en la extracción de una gramática libre de contexto del corpus etiquetado, identificación automática del elemento rector en cada regla, y uso de esta información para la construcción del árbol de dependencias. Las reglas presentadas eran “planas”, con sólo un elemento marcado. El presente trabajo generaliza este método, introduciendo la estructura más compleja de subárbol en las reglas. Se demuestra el uso de tales reglas para el tratamiento de los pronombres relativos en el corpus Cast3LB del español. Otra generalización consiste en la operación de duplicación de las reglas, necesaria para el tratamiento de las conjunciones coordinativas. Con la técnica mejorada se identifican correctamente 88% de las dependencias.

Palabras clave: Corpus anotados sintácticamente, constituyentes, dependencias, formalismos gramaticales.

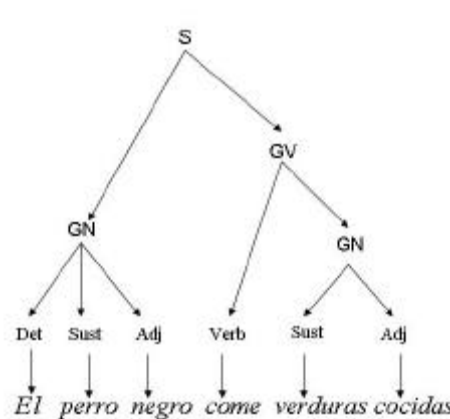
1 Introducción

Existen dos formalismos principales para la representación de la estructura sintáctica: constituyentes y dependencias. Ambos tipos de gramáticas usan árboles para representar la estructura de una frase, sin embargo, el significado de los nodos y sus relaciones es diferente.

En la gramática de constituyentes, los nodos del árbol representan segmentos del texto y sus relaciones están en una relación de inclusión:

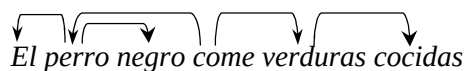
$[[El\ perro\ negro]_{GN} [come\ [verduras\ cocidas]_{GN}]_{GV}]_S$

o en forma de árbol:

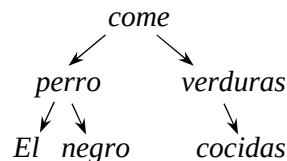


Los nodos están etiquetados: GN es un grupo nominal, GV un grupo verbal, S una oración.

En la gramática de dependencias, los nodos del árbol son palabras, y una dependencia es establecida entre un par de palabras cuando una de las palabras es la gobernante y la otra es subordinada (o dependiente) de la primera.



o en forma de árbol:



Cada palabra (excepto la raíz del árbol) tiene una palabra que la gobierna. Una relación de dependencia entre el gobernante G y el dependiente D significa que la combinación $G D$ (ó $D G$) tiene sentido y hereda las características sintácticas y semánticas de G : $G D$ es una G (y no es una D); se dice que D modifica a G . En el ejemplo, *perro negro* es un tipo de *perro* (y no de *negro*); *perro come verduras* es un tipo de (la situación de) *comer* (y no de *verduras*). Al diferencia del árbol

de constituyentes, en el árbol de dependencias los arcos pueden estar etiquetados: *negro* es atributo modificador de *perro*; *perro* es el sujeto de *comer*.

La representación de dependencias simplifica grandemente algunas tareas en la lexicografía (Bolshakov & Gelbukh 2001, Sidorov 2005), análisis sintáctico (Yuret 1998; Gelbukh 1999; Calvo & Gelbukh 2006), recuperación de información (Koster 2004), análisis semántico (Sowa 1984, Mel'cuk 1988, Sag et al. 2003), etc.

Sin embargo, la mayoría de las herramientas y de recursos existentes se orientan a la representación de constituyentes, para la cual es más simple construir un programa de análisis.

A pesar de diferencias evidentes, ambas representaciones comparten información sobre la estructura sintáctica a tal grado que pueden ser combinadas (Sag et al. 2003). Más aún, salvo ciertos casos específicos (tales como construcciones no-proyectivas), una puede derivarse automáticamente de la otra, obteniendo y agregando información que está presente en la primera representación y ausente en la segunda.

En (Gelbukh et al. 2005) hemos demostrado cómo se puede agregar información faltante a un árbol de constituyentes para convertirlo en un árbol de dependencias. Como trabajo pendiente en aquel artículo se quedaron ciertas deficiencias en el tratamiento de las conjunciones coordinativas y los pronombres relativos. El presente trabajo explica su tratamiento correcto.

En aquél artículo, los patrones usados para la conversión del árbol eran planos (véase la sección 3), con un sólo rector y los demás nodos subordinados a él. La aportación principal del presente trabajo consiste en dos generalizaciones al formalismo para la descripción de las reglas: a) se introduce la operación de multiplicación del patrón—véase la sección 4— y b) se introducen los patrones con su propia estructura, los cuales generan subárboles más complejos en el árbol resultante—véase la sección 5—.

El artículo se organiza como sigue. La sección 2 explica brevemente el corpus usado. La sección 3 presenta una breve descripción del procedimiento de transformación. La sección 4 explica el tratamiento que le dimos a las conjunciones coordinativas en el proceso de transformación. La sección 5 discutimos acerca del tratamiento que creemos correcto para el caso de los pronombres relativos. La sección 6 presenta los resultados experimentales, y la sección 7 da las conclusiones.

2 El corpus en español Cast3LB

Cast3LB es un corpus de cien mil palabras (aproximadamente 3,500 oraciones) creado a partir de dos corpus: CLiCTALP (75,000 palabras), un corpus balanceado y anotado morfológicamente que contiene

un lenguaje literario, periodístico, científico, etc., y el corpus de la agencia de noticias española EFE (25,000 palabras) correspondiente al año 2000.

El corpus se ha anotado con varios tipos de anotaciones: sintácticas, pragmáticas, los sentidos de palabras, etc. (Navarro et al. 2003). Por ejemplo, las anotaciones de los sentidos de palabras se asignan a todas las palabras con significado léxico, es decir, sustantivos, verbos y adjetivos. Estas anotaciones se usaron para llevar a cabo varios proyectos, tales como el aprendizaje automático para la desambiguación de sentidos de palabras (Izquierdo-Bevia et al. 2006). En el presente trabajo, sólo usamos las anotaciones sintácticas, y de éstas, omitimos ciertos detalles, como se explica a continuación.

3 Procedimiento de transformación

La transformación del corpus de constituyentes a uno de dependencias según (Gelbukh et al. 2005) consiste de los siguientes pasos:

1. Extracción de reglas de gramática de constituyentes del corpus Cast3LB.
2. Etiquetamiento de rectores: Uso de heurísticas para marcar el rector de la regla extraída.
3. Uso de la información de rectores para determinar cuáles de ellos subirán en el árbol.

Brevemente presentamos cada uno de ellos.

La extracción de la gramática del corpus Cast3LB consiste en los siguientes dos pasos:

Simplificación de constituyentes. Todas las etiquetas utilizadas por el corpus se componen de dos partes. La primera especifica la categoría gramatical, por ejemplo, sustantivo, verbo, grupo nominal, grupo verbal, oración, etc. Esta parte de la etiqueta es la más importante para la extracción de la gramática.

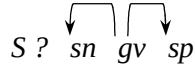
La segunda parte especifica datos adicionales como pueden ser género y número para frases nominales, o el tipo de la oración subordinada. Para simplificar el corpus y reducir el número de reglas gramaticales eliminamos la segunda parte de la etiqueta (datos adicionales), ya que al hacerlo no se afecta la estructura del corpus, por lo tanto la transformación tampoco.

Además, para reducir el número de patrones de la gramática resultante, simplificamos el corpus eliminando todos los signos de puntuación (puntos, comas, etc.).

Extracción de reglas. Para extraer las reglas de la gramática, se recorre cada árbol del corpus en profundidad, los nodos con más de un hijo se consideran como la parte izquierda de una regla, y sus hijos se consideran la parte derecha de la regla. Por ejemplo, para el nodo raíz en la figura 1, se genera la regla:

$S ? \quad sn \quad gv \quad sp.$

Etiquetamiento de rectores. En las reglas extraídas, marcamos automáticamente el rector de cada regla utilizando reglas de heurística. El rector de cada regla se marca con el símbolo @, por ejemplo: $S ? \quad sn \quad @gv \quad sp$. Como se verá más adelante, @ marca la raíz en un subárbol plano (de dos niveles) resultante de tal regla:



Las heurísticas utilizadas para marcar el rector de cada regla se pueden ver en (Gelbukh *et al.* 2005).

Uso de los rectores para la transformación. El algoritmo de transformación usa recursivamente la información de las reglas de gramática marcadas con rectores para determinar cuáles componentes subirán en el árbol. Esto significa desconectar el nodo rector de sus nodos hermanos y colocarlo en la posición del nodo padre. El algoritmo se detalla a continuación:

Recorriendo el árbol en profundidad, para cada nodo n :

- └ Determinar entre sus hijos el nodo rector, r , según las reglas de la gramática.
- └ Subir r a la posición del nodo padre n .

4 Conjunciones coordinativas

No existe consenso en los formalismos de dependencias de cómo manejar conjunciones coordinativas. Nosotros hicimos lo siguiente.

Cuando se encuentra una conjunción coordinativa (y, u o), se duplica (o multiplica) la oración de forma tal que cada ejemplar contendrá la relación sintáctica de sólo un elemento de la conjunción. Por ejemplo, la frase *nombres para niños y niñas* se interpreta como *nombres para niños* y *nombres para niñas*, creando así dos árboles diferentes. La figura 1 (a) presenta el árbol de constituyentes de la oración “*Los policías y otros numerosos agentes de seguridad nacionales y extranjeros velarán por la seguridad de los líderes*”. El algoritmo duplica el árbol al encontrar la primera conjunción (y) marcada como rector, obteniendo los árboles mostrados en la figura 1 (b) y (c).

Para el tratamiento de tales casos, generalizamos el formalismo de nuestras reglas, introduciendo una notación especial que causa la duplicación de la oración, por ejemplo:

$sn? \quad [sn] \quad coord \quad [sn].$

5 Pronombres relativos

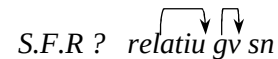
El algoritmo de transformación de (Gelbukh *et al.* 2005) no da un tratamiento adecuado para el caso de los pronombres relativos, pero es importante hacerlo para mejorar los resultados del corpus de dependencias extraído a partir del corpus de constituyentes Cast3LB.

Los pronombres relativos son palabras (*que*, *cuyo*, *quien*, etc.) que se refieren a otra anterior, antecedente, e introducen una oración finita relativa de la que forman parte. En la figura 2 se muestra una oración con el pronombre relativo *que*, el cual se refiere a su palabra anterior *reforma* e introduce la oración finita relativa “*que permita un sistema ágil*” de la que forma parte.

El algoritmo original encontraría el patrón $S.F.R ? \quad relatiu \quad gv \quad sn$, marcaría como rector el pronombre relativo quedando de la siguiente forma $S.F.R ? \quad @relatiu \quad gv \quad sn$, y lo subiría en el árbol, generando la estructura mostrada en la figura 3 (a). Sin embargo, la estructura correcta es como la mostrada en la figura 3 (b).

Para el tratamiento correcto de este caso, no sólo se tiene que subir el elemento *relatiu*, sino de entre los demás elementos se tiene que elegir un rector secundario, subordinarlo al rector principal, y subordinar los demás elementos a éste. Sin embargo, la estructura “plana” –de exactamente dos niveles– de las reglas originales, obliga al algoritmo de transformación a subordinar todos los hermanos del nodo rector a éste último cuando éste sube en el árbol, lo que en el caso en cuestión da resultado incorrecto.

Por lo tanto, se generaliza la notación de las reglas con subárboles más profundos:



Para implementar las heurísticas que automáticamente marcan la estructura en las reglas, introducimos otra marca @@ y la aplicación

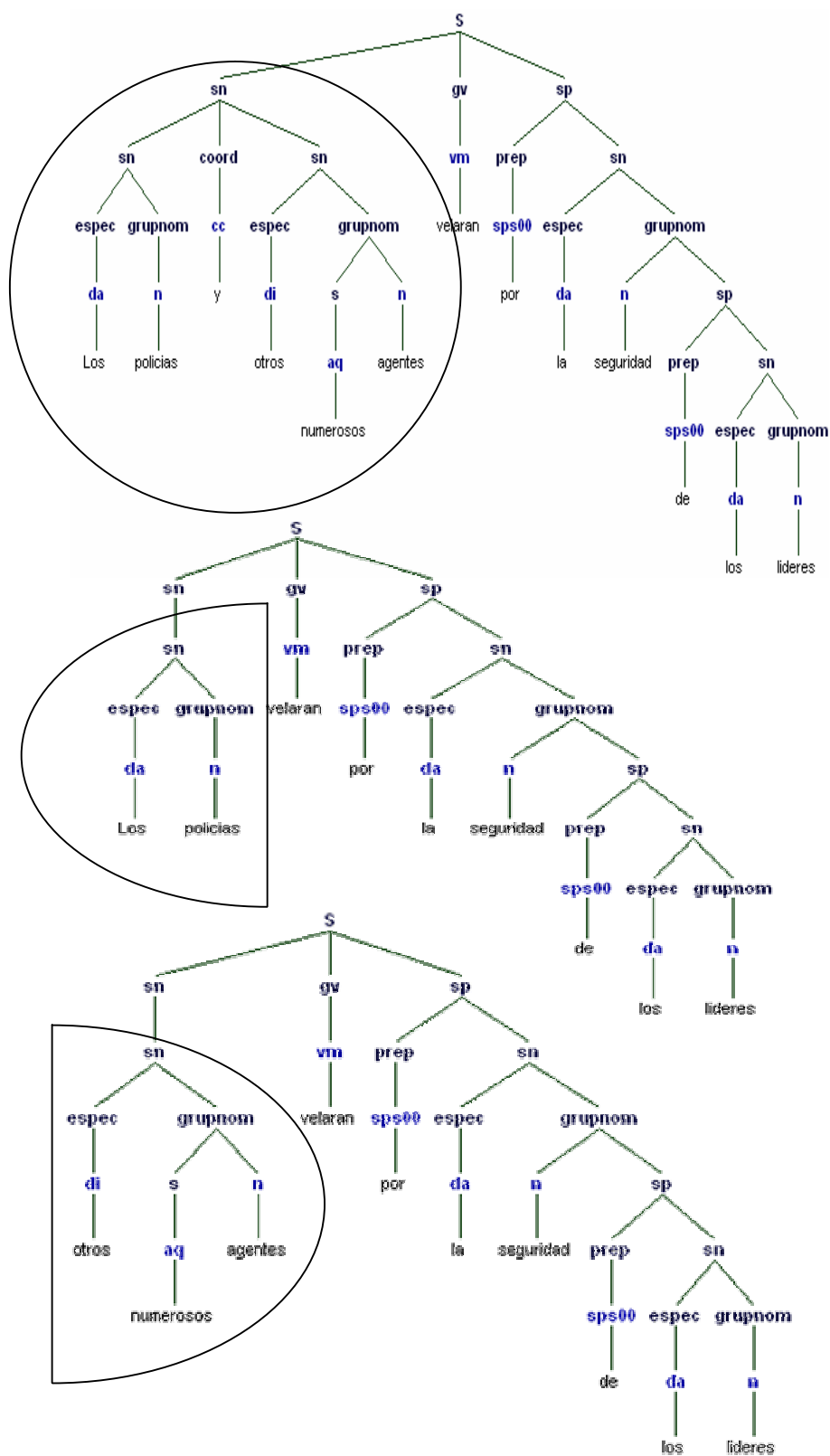


Figura 1. El árbol original de la oración “Los policías y otros numerosos agentes velarán por la seguridad de los líderes” y dos árboles resultantes de la conjunción.

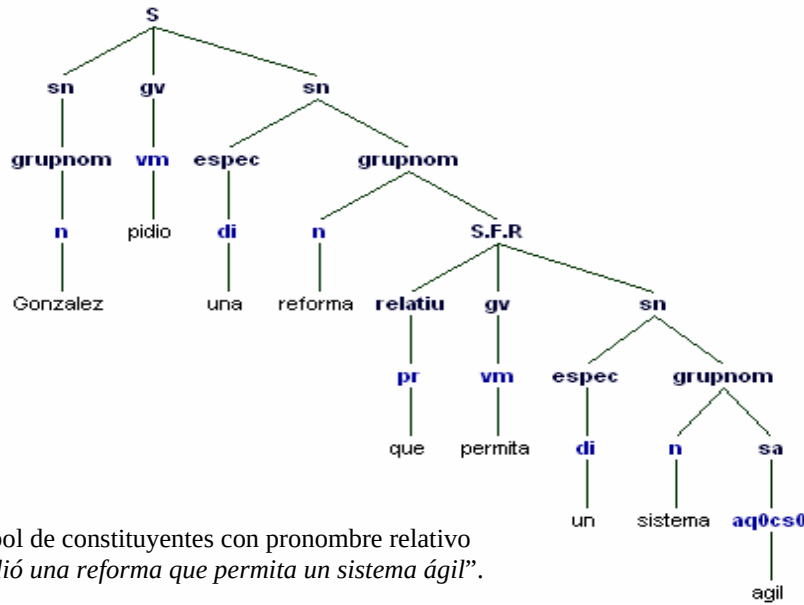


Figura 2. Árbol de constituyentes con pronombre relativo “González pidió una reforma que permita un sistema ágil”.

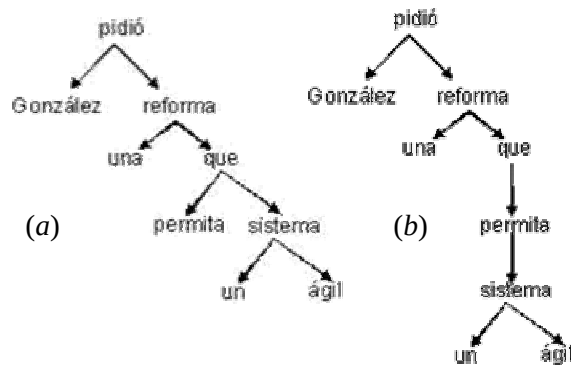


Figura 3. Árbol de dependencias equivocado (a) y correcto (b) para “González pidió una reforma que permita un sistema ágil”.

recursiva de las heurísticas. Al conjunto de las heurísticas se agregó una más que marca pronombres relativos con esta marca: @@relatiu. El proceso de la aplicación de esta heurística especial se queda como sigue:

- Heurísticas con esta marca tienen prioridad sobre otras. Se aplica para marcar el pronombre: S.F.R ? @@relatiu gv sn.
- Se genera una regla con el elemento marcado removido: S.F.R ? gv sn.
- Se aplican las demás heurísticas en orden normal, para determinar el rector de tal patrón: S.F.R ? @gv sn.
- Se regresa el elemento con la marca especial: S.F.R ? @@relatiu @gv sn.

La regla resultante se interpreta para generar el subárbol mostrado más arriba: cuando la regla conten-

ga el rector @@, el algoritmo lo coloca en la posición de su nodo padre; el rector @ como un nodo hijo del rector @@, y los nodos no marcados los subordina al rector @.

6 Resultados experimentales

Seguimos el esquema de evaluación propuesto por (Briscoe et al. 2002), que sugiere evaluar la precisión con base en relaciones gramaticales entre cabezas lematizadas.

Para obtener los valores de las métricas de evaluación *precision* y *recall*, se extraen colocaciones de los árboles de dependencias resultantes de nuestro método y se compara con colocaciones extraídas manualmente del mismo corpus Cast3LB.

Consideramos la colocación como una relación entre un nodo padre con un nodo hijo y su tipo de relación. Por ejemplo, las colocaciones extraídas de la frase “Los policías y otros numerosos agentes velarán por la seguridad de los líderes” son:

policías DET los
agentes DET otros
agentes ADJ numerosos
seguridad de líderes
velarán SUST agentes
velarán SUST policías
velarán por seguridad

Para la evaluación, seleccionamos aleatoriamente 35 oraciones del corpus y se convirtieron manualmente a árboles de dependencias, de los cuales se extraen 419 colocaciones. Aplicamos nuestro algoritmo a esas mismas oraciones para obtener los árboles de depen-

dencias y extraemos las colocaciones automáticamente.

El sistema extrae automáticamente 417 colocaciones de los 35 árboles seleccionados aleatoriamente. De las 419 colocaciones extraídas manualmente, el sistema extrae 370 (88%).

De esta forma, la calidad de los árboles de dependencias extraídos del corpus Cast3LB se puede evaluar de la siguiente manera:

Métrica de evaluación	Valor
Precisión	0.883
Recall	0.887

7 Conclusiones

La representación de dependencias de estructuras sintácticas tiene importantes ventajas en ciertas aplicaciones. Sin embargo, la mayoría de las herramientas y recursos léxicos existentes están orientados al formalismo de constituyentes.

Hemos presentado anteriormente una técnica para convertir, automática o semiautomáticamente, un corpus representado en el formalismo de constituyentes en uno representado en el formalismo de dependencias, proponiendo heurísticas para la conversión. En este trabajo, introducimos dos generalizaciones del formalismo de reglas de tal conversión:

- Duplicación o multiplicación de las reglas, necesaria para el tratamiento de las conjunciones coordinativas, y
- La estructura compleja de subárbol en las reglas, la cual necesaria para especificar el tratamiento de los pronombres relativos.

Agradecimientos. El trabajo fue realizado con soporte parcial del Gobierno de México (SNI, CONACyT, SIP-IPN, PIFI-IPN). Los autores agradecen al revisor anónimo por sus valiosos comentarios.

8 Referencias

Bolshakov, Igor A., Alexander Gelbukh. A Large Database of Collocations and Semantic References: Interlingual

Applications. *International J. of Translation*, V.13, No.1–2, 2001, pp. 167–187.

Briscoe, Ted, John Carroll, Jonathan Graham and Ann Copestake. 2002. Relational evaluation schemes. In: *Proceedings of the Beyond PARSEVAL Workshop at the 3rd International Conf. on Language Resources and Evaluation*, Las Palmas, Gran Canaria, 4–8.

Calvo, H., A. Gelbukh. 2006. DILUCT: An Open-Source Spanish Dependency Parser Based on Rules, Heuristics, and Selectional Preferences. *NLDB 2006, Lecture Notes in Computer Science*, 3999, Springer, 164–175.

Gelbukh, A. Syntactic disambiguation with weighted extended subcategorization frames. *Proc. PACLING-99*, 1999, pp. 244–249.

Gelbukh, A., H. Calvo, S. Torres. 2005. Transforming a Constituency Treebank into a Dependency Treebank. *Procesamiento del Lenguaje Natural* No. 35, España.

Izquierdo-Bevia, R., L. Moreno-Monteagudo, B. Navarro, A. Suarez. 2006. Spanish All-Words Semantic Class Disambiguation Using Cast3LB Corpus. In: A. Gelbukh & C. A. Reyes-Garcia (eds). *MICAI 2006: Advances in Artificial Intelligence. Lecture Notes in Artificial Intelligence*, Springer, a publicarse.

Koster, C. H. A. 2004. Head/Modifier Frames for Information Retrieval. In: A. Gelbukh (ed.), *Computational Linguistics and Intelligent Text Processing, CICLing-2004*, Lecture Notes in Computer Science, N 2945, Springer, pp 420–432.

Mel’cuk, Igor A. *Dependency Syntax: Theory and Practice*. State University Press of New York, 1988.

Navarro, Borja, Montserrat Civit, M. Antonia Martí, R. Marcos, B. Fernández. Syntactic, Semantic and Pragmatic Annotation in Cast3LB. *Shallow Processing of Large Corpora (SProLaC)*, a Workshop on Corpus Linguistics, Lancaster, UK, 2003.

Sag, Ivan, Tom Wasow, and Emily M. Bender. *Syntactic Theory. A Formal Introduction* (2nd Edition). CSLI Publications, Stanford, CA, 2003.

Sidorov, Grigori. Linguist’s Intelligent Workplace for Spanish Language. In: *Advances in artificial intelligence and computer science*, special issue of *J. Research on computing science*, 14, 2005, pp. 173–178.

Sowa, John F. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley Publishing Co., Reading, MA, 1984.

Yuret, Deniz. *Discovery of Linguistic Relations Using Lexical Attraction*, PhD thesis, MIT, 1998.