RISI **1**(1), 19-26 (2004)
Rev. investig. sist. inform.
Facultad de Ingeniería de Sistemas e Informática
Universidad Nacional Mayor de San Marcos
ISSN: 1815-0268 (impreso)

UN ANALIZADOR SINTÁCTICO EFICIENTE PARA GRAMÁTICAS DEL ESPAÑOL

Nora La Serna*

RESUMEN

El trabajo que se presenta en este artículo se enmarca en el área de la Lingüística Computacional, concretamente en el proceso del análisis sintáctico para el tratamiento de la lengua española. Básicamente, se describe el desarrollo e implementación de un analizador sintáctico eficiente que utiliza un prototipo de gramática computacional del español. La base del sistema es el algoritmo *Left-corner* o esquina de la izquierda, y adicionalmente se ha definido el formalismo de unificación para Gramática de Cláusulas Definidas (DCG) que el sistema utiliza. La eficiencia del analizador se debe principalmente a la implementación del algoritmo para gramáticas de unificación y a la creación del módulo en donde se realiza la compilación de las reglas. El proceso del análisis sintáctico se ha realizado con oraciones que han sido tomadas del corpus del proyecto MULTEXT; dichas oraciones contienen una amplia variedad de estructuras sintácticas del español. Finalmente, se discute la evaluación del sistema a partir de los resultados obtenidos, y se proyectan trabajos futuros por desarrollar.

Palabras clave: Inteligencia Artificial, Lingüística Computacional, Gramáticas del Español, Analizadores Sintácticos, Corpus.

AN EFFICIENT SYNTACTIC ANALYZER FOR SPANISH GRAMMARS

ABSTRACT

The research that is presented in this article is developed in the Computational Linguistics Area, specifically in the syntactic parsing processing for the treatment of the Spanish language. Basically, the development and implementation of an efficient syntactic parser is described, which utilizes a prototype of computational grammar for the Spanish. The base of the system is the Left-corner algorithm; and additionally, the Unification Formalism for Definite Clause Grammar (DCG) that the system utilizes has been defined. The efficiency of the parser owes mainly to the implementation of the algorithm for unification grammars, and to the creation of the module where the compilation of the rules is carried out. The process of the syntactic parsing has been achieved with sentences that have been taken from the corpus of the project MULTEXT, those sentences contain an extensive variety of the Spanish syntactic structures. Finally, the evaluation of the system is argued from the obtained results, and future jobs are projected for developing.

Keywords: Artificial Intelligence, Computational Linguistics, spanish grammar, syntactic parser, corpus.

1. INTRODUCCIÓN

En análisis sintáctico, el enfoque de las investigaciones básicamente están dedicadas a la construcción de formalismos gramaticales de diferentes estilos y características y a su procesamiento eficiente mediante la optimización de algoritmos de análisis. El algoritmo de análisis es el método utilizado para decidir si una cadena de símbolos es una frase de un lenguaje dado, determinando su estructura sintáctica de acuerdo a la gramática.

En este trabajo se presenta el desarrollo e implementación de un analizador sintáctico eficien-

* Docente de la Facultad de Ingeniería de Sistemas e Informática, Universidad Nacional Mayor de San Marcos, Lima-Perú. E-mail: nlasernap@unmsm.edu.pe

te. El analizador básicamente contiene tres módulos: 1) el algoritmo *Left-corner* o esquina de la izquierda para gramáticas de unificación, 2) el algoritmo de unificación de las estructuras de rasgos, y 3) el módulo de compilación de las reglas. A partir del proceso del análisis sintáctico de una muestra importante de oraciones del español, presentamos la evaluación del sistema implementado; la muestra fue seleccionada desde el corpus del proyecto MULTEXT (Multilingual Text Tools and corpora). El corpus mencionado es un recurso importante utilizado. Contiene una gran variedad de oraciones con diferentes estructuras sintácticas y la codificación estandarizada de las etiquetas que albergan información lingüística importante.

Los formalismos gramaticales permiten escribir y tratar computacionalmente las gramáticas de una lengua. Los formalismos de unificación expresan en forma estructurada la información sintáctica de una gramática. En este trabajo utilizamos un tipo de estos formalismos, el que es una versión simplificada de la Gramática de Cláusulas Definidas (DCG, Definited Clause Grammar; Pereira & Warren 1980 y 1983). Adicionalmente, la gramática computacional para el español que se emplea (N. La Serna 2001) describe las construcciones sintácticas básicas de la lengua.

La organización del resto del artículo es la siguiente: en la sección 2 se bosqueja la construcción de la gramática del español y su cobertura actual, en la sección 3 se describe el formalismo de unificación utilizado, en la sección 4 se explica el proceso del análisis sintáctico, su desempeño y trabajos relacionados, y finalmente, en la sección 5 se presentan las conclusiones del trabajo realizado y las tareas para su desarrollo posterior.

II. PROTOTIPO DE GRAMÁTICA DEL ESPAÑOL

2.1 Metodología de construcción

La definición de la gramática computacional para el español se hizo en forma manual y se utilizó un corpus etiquetado como fuente de información lingüística. Las reglas de la gramática se construyeron en base a la observación de las estructuras lingüísticas que aparecen en las frases y oraciones del corpus, teniendo en cuenta la teoría lingüística de la gramática. Para cada categoría sintáctica y léxica de las reglas y del léxico se codificaron sus atributos y valores variables, utilizando el formato que se presenta en la sección 3, y considerando en cada regla la concordancia de los atributos o rasgos de las categorías madre (en la parte izquierda de las reglas) y hermanas (en la parte derecha de las reglas). La información lingüística fue tomada desde la definición de las etiquetas del corpus.

Por ejemplo, en la seccion 3 mostramos un fragmento de la gramática, cuyas reglas de estructura de frase se pueden obtener a partir de la oración *El(tdms) número(ncms) de(sp) personas(ncfp) que(pr3xxx) han(vaip3px) aprobado(vmpxxsm) concursos(ncmp) y(cc) figuran(vmip3px) en(sp) listas(ncfp) de(sp) reserva(ncfs).* Las etiquetas del corpus proporcionan la categoría léxica y los atributos que podrían contener las categorías en general; por ejemplo, en la oración anterior la etiqueta *tdms* de la palabra *El* indica que es artículo, de tipo definido, género masculino y número singular.

El corpus utilizado en este trabajo es parte del corpora desarrollado en el proyecto MULTEXT (Multilingual Text Tools and corpora)[1], el que contiene datos de las preguntas y respuestas escritas realizadas por los miembros del Parlamento Europeo. Los datos versan sobre una amplia variedad de tópicos y son publicadas en el Diario Oficial de la Comunidad Europea. Un total de 200 000 palabras para el español están gramaticalmente etiquetadas, y además se encuentran manualmente verificadas. Las etiquetas representan la información morfosintáctica de cada palabra en el corpus.

2.2 Cobertura de la gramática

La versión actual de la gramática contiene una amplia variedad de construcciones sintácticas básicas. Podemos citar las siguientes:

- Oraciones de tipos declarativos e interrogativos; afirmativas y negativas.
- Oraciones simples y compuestas; compuestas coordinadas y subordinadas.
- Estructuras coordinadas de las siguientes categorías: frases verbales, nominales, adverbiales y adjetivas.
- Tratamiento de diferentes tipos de complementos.
- Estructuras para varias clases de verbos: con complementos de objeto directo e indirecto; complementos obligatorios y opcionales.

[1] Corpora Multext es un conjunto paralelo de datos escritos en cinco lenguas: inglés, español, francés, alemán e italiano.

- Construcciones verbales auxiliares, pasivas y reflexivas.

En la figura N.º 1, se muestra el número de componentes de la gramática; se han considerado 16 categorías, el número de atributos para todas las categorías léxicas que se presenta en Multext es 37, el número de reglas definidas es 127. En la creación del léxico, se han colocado las etiquetas definidas en el corpus en lugar de las palabras que pertenecen a la lengua; de esta manera hemos definido 132 unidades o entradas en el vocabulario, que representan a todas las palabras que puedan aparecer en el corpus. En la construcción del prototipo se han utilizado solamente 162 oraciones del corpus, que representan un 20% del total; en la siguiente etapa del trabajo se plantea continuar y alcanzar una gramática real y de amplia cobertura. La longitud promedio de las oraciones en el corpus es de 28 palabras.

	Categorías	Atributos	Reglas
Sintácticas	6	43	127
Léxicas	10	37	132

Figura 1. Componentes de la gramática.

2.3. El Formalismo de Unificación

La representación formal de las gramáticas han ido evolucionando, desde los modelos más sencillos en los que las categorías gramaticales eran simples etiquetas (aunque éstas están vigentes actualmente) hasta los modelos más modernos en los que la información lingüística se codifica en formas de estructuras relacionadas, dando lugar a las conocidas gramáticas de unificación. El formalismo gramatical que se ha definido para este trabajo es una versión simplificada de la Gramática de Cláusulas Definidas (DCG), el que permite escribir la gramática en una forma natural. En la figura 2 y 3 presentamos el formato de las reglas de la gramática y entradas del léxico respectivamente; los componentes *categoría* y *atributo (atr)* toman valores atómicos, mientras que *valor (val)* puede ser atómico u otra lista de atributos y valores. Los subíndices n, m, j y k pertenecen al conjunto de los números enteros y deben ser finitos.

etiqueta:

categoría ([atr_1 val_1 , atr_2 val_2, ..., atr_n val_n])

Figura N.º 3. Formato del léxico.

En la Figura N.º 4 presentamos un fragmento de la gramática para ilustrar este formalismo. Las líneas que empiezan con el caracter *%* muestran la regla de estructura de frase de la regla que se presenta en la siguiente línea. Por ejemplo, la regla 4 define grupos verbales coordinados, como en la frase *han aprobado concursos y figuran en listas de reserva.*

Observamos además, en la regla 4, las listas de atributos y sus valores variables que corresponden a las categorías de la regla; los atributos que se muestran para las frases verbales en esa regla son *estado (stat), modo (mood), tiempo (tense), persona (per), número (num) y clítico* (*clit*). Podemos resaltar que los valores de los atributos del primer *grupo verbal* (vp) son diferentes que aquéllos en el segundo grupo, como podemos comprobarlo en la frase. Concretamente *aprobado* es un verbo en *modo* (*mood*) participio, que además precedido por el verbo auxiliar *han* constituyen una forma compuesta verbal, mientras que *figuran* es un verbo en modo indicativo.

Similarmente, en la figura 5 presentamos un fragmento del léxico donde se observan las etiquetas del corpus, sus correspondientes categorías léxicas y atributos definidos para cada entrada.

% *r1: S —> np vp*

s([gen G, stat SV, mood MV, tense T, per P, num N]) —> np([gen G, num N]), vp([stat SV, mood MV, tense T, per P, num N2, clit CL]).

% *r2: np —> art n*

np([gen G, num N]) —> art([type TR, gen G, num N]), n([type TN, gen G, num N]).

% r3: np —> np pp

np([gen G, num N]) —> np([gen G, num N]), pp([]).

categoría1 ([$atr1_1$ $val1_1$,..., $atr1_n$ $val1_n$]) —> categoría2 ([$atr2_1$ $val2_1$,..., $atr2_m$ $val2_m$]) ,..., categoríaj ([$atrj_1$ $valj_1$,..., $atrj_k$ $valj_k$])

Figura N.º 2. Formato de las reglas de la gramática.

```
% r4: vp —> vp conj vp

vp([stat SV, mood MV, tense T, per P, num N, clit CL]) —> vp([stat SV, mood MV, tense T, per P, num N, clit CL]), conj([type TC]), vp([stat SV2, mood MV2, tense T2, per P2, num N2, clit CL2]).

% r5: vp —> va vp

vp([stat SV, mood MV, tense T, per P, num N, clit CL]) —> va([stat SV, mood MV, tense T, per P, num N, clit CL]), vp([stat SV2, mood MV2, tense T2, per P2, num N2, clit CL2]).
```

Figura N.º 4. Ejemplo de reglas de la gramática.

```
tdms: art([type  def, gen m, num s]).

ncms: n([type com, gen m, num s]).

vaip3px: va([stat aux, mood ind, tense pre, per 3, num p]).

vmpxxsm: v([stat main, mood part, num s, gen m]).

vmip3px: v([stat main, mood ind, tense pre, per 3, num p]).
```

Figura N.º 5. Fragmento del léxico definido.

Las reglas de la gramática para el español, escritos en el formalismo presentado, son compilados con la finalidad de obtener mejor eficiencia en el proceso del análisis. El formato compilado de las estructuras de rasgos están en términos prolog, las referencias principales son tomadas de la teoría para gramáticas de unificación presentadas por S. Shieber (1986), y de los trabajos desarrollados en análisis sintáctico por D. Gerdemann (1991, 1993) y N. La Serna (1997, 1998).

Las estructuras de rasgos en este formato son Dags (directed acyclic graphs) de acuerdo a la notación de Shieber (1986):

Dag (FS, RL), donde FS (Feature Structure) es una estructura de rasgos y RL (Reentrancy List) es una lista de reentrancias;

FS = [feat (Att1, Value), ..., feat (Attn, Value)], es decir una lista de términos prolog *feat*, donde cada término tiene dos argumentos (Att y Value). *Att* representa un rasgo y *Value* puede ser un valor atómico u otra estructura de rasgos (FS) o una variable indicando reentrancia;

RL = [re (RNum1, FS), ..., re (RNumn, FS)], es decir, una lista de términos *re*, donde cada término tiene dos argumentos, (Rnum, FS); *Rnum* es un número de reentrancia.

III. EL PROCESO DEL ANÁLISIS SINTÁCTICO

3.1 El Analizador

Varios son los procesos que hemos desarrollado e implementado para el análisis sintáctico. Podemos destacar los más importantes:

- El algoritmo de análisis, es un *Left-corner* o esquina de la izquierda para gramáticas de unificación.
- El programa que realiza el proceso de compilación de las reglas de la gramática.
- El algoritmo de unificación de las estructuras de rasgos.

La estrategias básicas que se aplican a los analizadores son: 1) *descendente* (top-down), y 2) *ascendente* (bottom-up); el algoritmo básico Left-corner o esquina de la izquierda (Rosenkrantz & Lewis 1970), toma ventaja de ambas estrategias y trabaja simultáneamente en forma descendente desde una categoría objetivo, y ascendente desde la categoría de la esquina de la izquierda en la parte derecha de una regla. Por ejemplo, en la regla s -> np vp, *s* es la categoría objetivo y *np* la categoría más a la izquierda en la parte derecha de la regla; cuando las categorías de nivel inferior de *np* han sido reconocidas, ésta es completada; y *vp* se convierte en la siguiente categoría de la esquina de la izquierda.

Los formalismos basados en unificación extienden el formalismo de las gramáticas de estructuras independientes del contexto en dos aspectos:

1. el uso de categorías complejas (estructuras de rasgos) en vez de simples etiquetas, y
2. utilizan la operación de unificación[1] para combinar la información lingüística.

A continuación describimos brevemente el algoritmo *Left-corner* para gramáticas de unificación. Representamos los estados intermedios del análisis mediante reglas punteadas x -> x1 * x2 ... xn donde x1, ..., xn son estructuras de rasgos o

atributos. La estructura de datos definida es una lista de la forma [N, L, ER], N indica la posición del punto en la regla, L es la regla de estructura de frase, ER la estructura de rasgos de la regla.

El algoritmo empieza con una regla-de-inicio, cuya categoría madre es la categoría objetivo, la regla se convierte entonces en una regla-objetivo. El análisis termina cuando: 1) todas las palabras de la cadena de entrada han sido reconocidas, y 2) todas las categorías de la parte derecha de la regla-de-inicio han sido unificadas; de otra manera el algoritmo falla.

- Para cada palabra de la cadena de entrada y la regla-objetivo actual se hace una búsqueda de la categoría más a la izquierda *(left-corner)*.
- En el proceso de búsqueda de la categoría más a la izquierda, se realiza uno de los dos procedimientos:
  1. *Left-corner-NT*, verifica por unificación si la categoría no terminal a la derecha del punto de la regla-objetivo es una categoría más a la izquierda
  2. *Left-corner-T*, verifica por unificación si la categoría léxica de la palabra es una categoría más a la izquierda de la regla-objetivo.

Cuando la unificación se produce se actualiza una posición del punto en la regla-objetivo.

Cabe destacar que un aspecto importante para mejorar la eficiencia del analizador es el módulo de compilación de las reglas implementado, que como se explicó en la sección 3, el formato DCG es compilado a estructuras de rasgos en términos prolog, las referencias principales son tomadas de la teoría para gramáticas de unificación presentadas por S. Shieber (1986), y de los trabajos desarrollados en análisis sintáctico por D. Gerdemann (1991, 1993) y N. La Serna (1997, 1998).

3.2. Evaluación de los resultados y trabajos relacionados

Realizamos la evaluación del analizador implementado, a partir del proceso del análisis sintáctico de 162 oraciones del corpus MULTEXT. Las oraciones presentan diferentes longitudes, es decir varían en el número de palabras que forman cada oración; y además, especialmente se ha tomado en cuenta que estas oraciones contengan las diferentes estructuras sintácticas de la gramática. Un componente importante en el proceso del análisis sintáctico es la gramática, en este trabajo se ha utilizado el prototipo de gramática de unificación para el español, que se ha descrito en la sección 2 de este artículo.

La medida seleccionada para medir la eficiencia del analizador es el tiempo que emplea el algoritmo para el análisis de una frase u oración, en este trabajo se utiliza como unidad de medida el microsegundo, el que es 10^{-6} segs (ìseg). Para la selección de la medida mencionada se han tomado como referencia los trabajos desarrollados en N. La Serna (1998) y R. Moore (2000-b), en los cuales se demuestra que el tiempo referido es una medida adecuada para evaluar la eficiencia de los algoritmos de análisis.

En los párrafos siguientes mostramos gráficamente los resultados obtenidos del proceso del análisis sintáctico de las oraciones, y a partir de ellos hacemos una estimación de los resultados. Sin embargo, podemos adelantar y concluir que el rendimiento del analizador es óptimo, debido a que el 91.40 % de la oraciones procesadas presentan un tiempo de respuesta sumamente bueno. Adicionalmente, estos resultados nos permiten proyectarnos en las siguientes tareas que se deben realizar.

En la tabla de la Figura N.º 6, se presenta el número de las oraciones que han sido seleccionadas y analizadas, ellas están distribuidas por su longitud (número de palabras que forman cada oración), y el tiempo de proceso (en microsegundos)[2] que emplearon en el análisis sintáctico. Se han considerado tres rangos de longitud de las oraciones, aquéllas que tienen menos de 20 palabras, las que están entre 20 y menos de 40 y las que están entre 40 y 60. Mientras que, el tiempo de proceso empleado se ha distribuido en cinco intervalos, expresados en microsegundos ellos son: de 1 a 199, de 200 a 399, de 400 a 599, de 600 a 999, y de 1000 a 35000. Debemos resaltar que 142 de las 162 oraciones procesadas, es decir, el 88% de las oraciones tienen una longitud de menos de 40 palabras. Esto corrobora el hecho de que la longitud promedio de las oraciones del corpus es de 28 palabras.

[2] La operación de unificación consiste en combinar la información desde dos estructuras de rasgos para obtener otra tercera estructura que incluya toda la información de las dos primeras.

Longitud de las oraciones	Tiempo de proceso del análisis en microsegundos (ìseg)					
	1-199	200-399	400-599	600-999	1000-35000	Total
1-19	60	4	-	-	-	64
20-39	36	24	8	-	10	78
40-60	-	5	5	6	4	20
Total	96	33	13	6	14	162

Figura N.º 6. Tiempo de ejecución empleados en microsegundos.

En el gráfico N.º 7, las barras de color, azul, lila, y crema, representan a cada rango de las longitudes de las oraciones (0-19, 20-39, y 40-59). Mientras que en la coordenada horizontal están distribuidas los intervalos de tiempo, tal como se presentó en la tabla de la figura 6; en la coordenada vertical se representa el porcentaje de oraciones. Podemos observar en el gráfico que para cada barra de color (rango de longitud de las oraciones), el porcentaje de las oraciones se distribuye de acuerdo al intervalo de tiempo que emplearon las oraciones en su proceso. De esta manera el gráfico ilustra los resultados del análisis sintáctico de las 162 oraciones ejecutadas, resultados que se evalúan en el siguiente párrafo.

A partir de las Figuras N.os 6 y 7, observamos que el 94% de las oraciones que tienen longitudes de menos de 20 palabras emplean un tiempo de análisis de menos de 200 ¼seg el que es un tiempo óptimo en este tipo de procesos. Igualmente, podemos destacar la eficiencia del analizador al concluir que, el 77% de las oraciones que tienen longitudes entre 20 y menos de 40 palabras emplean menos de 400 ¼seg. En el caso de las oraciones con longitudes entre 40 y menos de 60 palabras, concluimos también que el 80% de las oraciones tuvieron un análisis eficiente, empleando menos de 1000 ¼seg. De acuerdo a lo observado, podemos resumir que el 91.40 % de todas las oraciones procesadas presentan un tiempo de respuesta muy bueno. Aún el 8.60 % de las oraciones restantes (las que están en el intervalo de 1000 a 35000 ¼seg) presentan un tiempo de proceso bastante aceptable.

Sin embargo, también debemos mencionar las limitaciones del trabajo, la gramática computacional no es de amplia cobertura, motivos por los que muchas oraciones del corpus no han podido ser analizadas. Por ejemplo nos faltan definir algunas estructuras para oraciones subordinadas, definir los numerales, etc. Son tareas que abordaremos en la siguiente etapa del proyecto. Otra omisión es que no se ha tratado la ambigüedad sintáctica, es decir a través del sistema con algunas oraciones se puede obtener varias soluciones (estructuras) de análisis; debemos incluir un procedimiento que ayude al analizador a decidir la solución más acertada.

Un aspecto importante que permitió obtener un analizador eficiente es la compilación de las

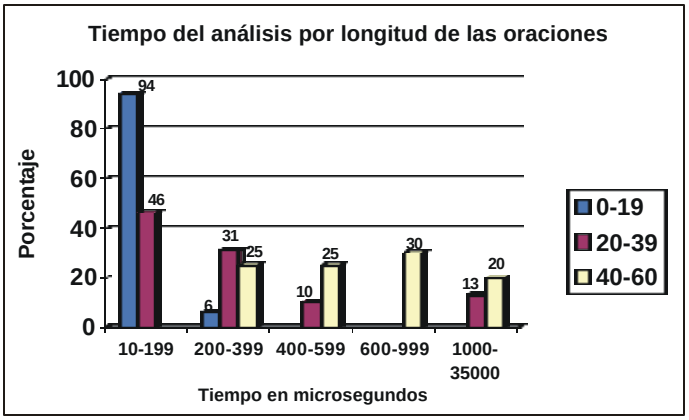

Figura N.º 7. Gráfico de barras, porcentaje de las oraciones en los intervalos de tiempo.

reglas de la gramática; varias son las investigaciones que utilizan una gramática compilada en el proceso del análisis sintáctico, especialmente aquéllas que trabajan en ambientes con gramáticas de unificación de amplia cobertura y reales, que es el objetivo que queremos llegar para el español. Podemos citar el trabajo de J. Carroll y T. Briscoe (1991) con el sistema ANLT (Alvey Natural Language Tools), en donde la gramática del Alvey se compila a una DCG (450 reglas). Otros sistemas que generan gramática compilada son el UNICORN y TROLL de D. Gerdemann y sus colegas (1991, 1993), nosotros adoptamos el mismo sistema de estructuras de rasgos compilado que ellos utilizan, que como mencionamos en la sección 3 se sigue la teoría para gramáticas de unificación presentadas por S. Shieber (1986).

Otro trabajo también relacionado con el nuestro es la versión mejorada del algoritmo Left-corner que presenta R. Moore (2000-a); en esta investigación él utilizó varias gramáticas de amplia cobertura, incluyendo gramáticas de unificación, las que luego son convierti das a gramáticas independiente del contexto.

IV. CONCLUSIONES

Hemos desarrollado un analizador sintáctico eficiente para gramáticas de unificación que utiliza un prototipo de gramática para el español, donde la gramática contiene las estructuras sintácticas básicas de la lengua. El algoritmo del analizador utilizado es el Left-corner o esquina de la izquierda, el que toma ventaja de otros algoritmos al hacer el análisis en forma ascendente y descendente simultáneamente. A partir de los resultados del análisis sintáctico de oraciones seleccionadas desde el corpus Multext, hemos evaluado el rendimiento del analizador, y concluido que proporciona un tiempo de análisis óptimo como se demuestra en la sección 4.2.; sin embargo, también hemos presentado algunas limitaciones del trabajo debido a la falta de una gramática de amplia cobertura, y procedimientos de desambigüación sintáctica. Para trabajos futuros, también nos planteamos comparar el desempeño del analizador con otros algoritmos para gramáticas de unificación, como J. Earley y Botton-up.

El trabajo se enmarca en un proyecto que pretende crear una gramática de amplia cobertura para el español utilizando formalismos de unificación, debido a que dichos formalismos permiten describir y tratar bien las gramáticas de una lengua. Naturalmente, el proyecto contempla el procesamiento eficiente de las frases y oraciones utilizando la gramática creada. También, debemos resaltar la riqueza del corpus Multext utilizado como fuente de información lingüística, especialmente en cuanto a la gran variedad de construcciones sintácticas que se pueden obtener.

Para continuar con el proyecto nos planteamos tareas inmediatas como, continuar con la construcción de la gramática utilizando la parte del corpus Multext no empleado, crear métodos automatizados de aprendizaje de las reglas y supervisión manual; tareas a mediano y largo plazo serían considerar otros recursos como etiquetadores y otros corpus.

V. BIBLIOGRAFÍA

1. Bello Andrés (1984), *Gramática de la Lengua Española*. Editorial EDAF, Madrid.
2. Carroll John & Briscoe Ted & Grover C. (1991), *A development Environment For Large Natural Language Grammars*. Technical Report No 233, Computer laboratory, Cambridge University, UK.
3. Earley J. (1970), *An efficient Context-free Parsing Algotithm*. Comunication of the ACM.
4. Gerdemann Dale (1991), *Parsing and Generation of Unification Grammars*. PhD Thesis, Technical Report CS-91-06 of The Beckman Institute, Illinois.
5. Gerdemann Dale (1993), *Troll, Type Resolution System, Fundamental Principles & User's Guide*. University of Tubingen, Germany.
6. Gili Gaya Samuel (1998), *Curso Superior de Sintaxis Española*. Ediciones Vox, España
7. La Serna Nora, Díaz A., Rodríguez H. (1997), *Parsers Optimization for Wide-Coverage Unification-Based Grammars using Restriction Technique*. International Workshop on Parsing Technologies, MIT, Cambridge, Ma.
8. La Serna Nora (1998), *Optimización de la técnica de restricción para obtener analizadores eficientes con gramáticas de unificación de amplia cobertura*. Tesis Doctoral, Universidad del País Vasco.
9. La Serna Nora (2001) *Un prototipo de gramática de unificación para el Español*. Second International Workshop on Spanish Language Processing and Language Technologies (SPLT-2), Jaen, Spain.

10. Moore Robert (2000) *Improved Left-Corner Chart Parsing for Large Context-Free Grammars*. Sixth International Workshop on Parsing Technologies, ACL/SIGPARSE, Trento, Italy.

11. Moore Robert (2000) *Time as a Measure of Parsing Efficiency*. th International Conference on Computational Linguistics (COLING'2000),.

12. Multilingual Text Tools and Corpora (MULTEXT), www.lpl.univ-aix.fr/projects/multext.

13. Rosenkrantz S. & Lewis P. (1970), *Deterministic Left Corner Parser*. In IEEE Conference of the 11th Annual Synposium on Switching and Automata.

14. Seco Rafael (1979), *Manual de Gramática Española*. Ediciones Aguilar, Madrid.

15. Shieber Stuart (1986), *An Introduction to Unification Based Approaches to Grammar*. CSLI Lecture Notes, University of Chicago Press.