

Sistema de adquisición automática de reglas gramaticales

J. Peral; P. Martínez-Barco; A. Ferrández, B. Navarro
{jperal, patricio, antonio, borja}@dlsi.ua.es
Grupo de Programación Lógica y Sistemas de Información
Dpto. de Lenguajes y Sistemas Informáticos
Universidad de Alicante
URL: <http://gplsi.dlsi.ua.es>

Resumen

En este trabajo presentamos un sistema automático de extracción de reglas sintácticas a partir de un corpus etiquetado con sus categorías gramaticales. Planteamos un sistema de definición de patrones sintácticos sencillo que es capaz de identificar las construcciones sintácticas de sintagmas nominales, sintagmas preposicionales y sintagmas verbales así como algunos subconstituyentes tales como las entidades. Además, el sistema está definido por niveles lo que le hace ser fácilmente adaptable a otros tipos de constituyentes y subconstituyentes según las necesidades del sistema. El sistema ha sido experimentado con un fragmento de corpus conteniendo 250 oraciones (aproximadamente 9600 palabras) etiquetadas y corregidas manualmente obteniendo en una primera aproximación un total de 335 reglas distintas que fueron analizadas manualmente detectando posibles fallos en la definición de patrones. Gracias a la flexibilidad que proporciona el sistema, una segunda definición de patrones que nos permite solucionar gran parte de los problemas detectados en el análisis mencionado, junto con una importante simplificación del conjunto de etiquetas gracias a la construcción de un interfaz previo, nos proporciona un conjunto de 72 reglas distintas acercando nuestro trabajo a los objetivos planteados.

Sistema de adquisición automática de reglas gramaticales¹

J. Peral; P. Martínez-Barco; A. Ferrández, B. Navarro
{jperal, patricio, antonio, borja}@dlsi.ua.es
Grupo de Programación Lógica y Sistemas de Información
Dpto. de Lenguajes y Sistemas Informáticos, Universidad de Alicante

Resumen

En este trabajo presentamos un sistema automático de extracción de reglas sintácticas a partir de un corpus etiquetado con sus categorías gramaticales. Planteamos un sistema de definición de patrones sintácticos sencillo que es capaz de identificar las construcciones sintácticas de sintagmas nominales, sintagmas preposicionales y sintagmas verbales así como algunos subconstituyentes tales como las entidades. Además, el sistema está definido por niveles lo que le hace ser fácilmente adaptable a otros tipos de constituyentes y subconstituyentes según las necesidades del sistema. El sistema ha sido experimentado con un fragmento de corpus conteniendo 250 oraciones (aproximadamente 9600 palabras) etiquetadas y corregidas manualmente obteniendo en una primera aproximación un total de 335 reglas distintas que fueron analizadas manualmente detectando posibles fallos en la definición de patrones. Gracias a la flexibilidad que proporciona el sistema, una segunda definición de patrones que nos permite solucionar gran parte de los problemas detectados en el análisis mencionado, junto con una importante simplificación del conjunto de etiquetas gracias a la construcción de un interfaz previo, nos proporciona un conjunto de 72 reglas distintas acercando nuestro trabajo a los objetivos planteados.

1. Introducción

Uno de los grandes retos del tratamiento automático del Lenguaje Natural es la adquisición automática del conocimiento lingüístico, o dicho de otro modo, la adquisición de reglas gramaticales. Para ello, en este trabajo presentamos un sistema sencillo, flexible y modular capaz de identificar las reglas sintácticas de SN, SP y SV; el interés de

este trabajo se centra en la posibilidad de reconocer constituyentes basados en expresiones regulares, además de ser un sistema de propósito general que, independiente del corpus a utilizar, sea capaz de adquirir un conjunto de reglas sintácticas. Esto se conseguirá mediante un interfaz universal que traduce etiquetas de diferentes tipos a un conjunto de etiquetas propias de nuestro sistema.

Posibles trabajos o aplicaciones a realizar con este sistema pueden ser la de realizar un etiquetado sintáctico parcial (según el estilo Tree Penn) o ampliar ese conocimiento lingüístico mediante n-gramas para detectar aquellas posibles reglas que no aparecen en el corpus de entrenamiento pero que forman parte del lenguaje a reconocer, uno de los problemas que se plantean en este caso es la sobregeneración de reglas gramaticales.

En la siguiente sección presentaremos una revisión de algunos sistemas empleados para la adquisición automática de conocimientos lingüísticos así como las motivaciones que nos han inducido a desarrollar el sistema propuesto. En la sección 3 describiremos el planteamiento genérico del algoritmo desarrollado para la adquisición de reglas gramaticales con una primera aproximación que analizaremos y corregiremos posteriormente. Para finalizar, presentaremos los resultados obtenidos sobre la prueba efectuada en un corpus concreto junto con un análisis de los mismos.

2. Antecedentes

Como se introduce en [Bri97] uno de los mayores desafíos en el procesamiento del lenguaje natural es conseguir que un ordenador contenga la información lingüística necesaria para desarrollar tareas basadas en el lenguaje. La solución que propone es que sea la propia máquina la que pueda adquirir este conocimiento.

[Abn94] propone una estrategia basada en aprovecharse del conocimiento lingüístico y las

¹ Este artículo ha sido subvencionado por el CICYT número TIC97-0671-C02-01/02.

reglas escritas manualmente cuando estén disponibles (caso de las teorías morfológicas y sintácticas) o su obtención mediante un esfuerzo moderado (como en las gramáticas de chunks), pero apoyándose en el uso de técnicas automáticas que obtengan gran parte de la información usada por el analizador, en particular los parámetros de probabilidad. En [Abn96a] se justifica lo anterior basándose en que la adquisición del lenguaje en los niños viene caracterizado por periodos de cambios durante los cuales el niño va aprendiendo o alterando una regla (o un parámetro de la regla) y que pueden llegar a durar varios meses. Si, como parece ser el caso, los cambios en la gramática del niño se reflejan en cambios en las frecuencias relativas de las estructuras, parece inmediato llegar a la conclusión de que el niño usa algún tipo de gramática probabilística. Se piensa que el niño intenta los cambios en las reglas durante un tiempo y que durante este tiempo de pruebas, tanto la versión antigua de la regla como la nueva coexisten, y la probabilidad de usar una u otra va variando con el tiempo, hasta que finalmente la probabilidad de usar la regla antigua se reduce a cero. Así mismo, se piensa que los cambios en el lenguaje de un adulto se basan en el uso de una gramática estocástica que coexiste con la gramática algebraica usada para mantener las reglas invariantes.

Tomando como partida sus teorías sobre la adquisición natural del conocimiento lingüístico en los humanos, [Abn97] define un analizador parcial (Cass) controlado por un autómata de estados finitos. Conceptualmente, está compuesto por una secuencia de reconocedores especializados. En el nivel inferior, el nivel 0, la entrada está formada por palabras con sus categorías gramaticales (símbolos terminales). En el siguiente nivel, el autómata encuentra como entrada algunas secuencias de palabras que concuerdan con un patrón determinado y las reduce a un elemento simple con su categoría apropiada (símbolos no terminales). La salida del nivel 1 se toma como entrada para el nivel 2 que deducirá a su vez nuevos símbolos no terminales, y así sucesivamente. El autómata por niveles descrito se define usando una gramática de expresiones regulares que formará los distintos patrones a derivarse en cada nivel. Por ejemplo:

:chunk

$NP \rightarrow D? N+;$

$VP \rightarrow V-tns / Aux V-ing;$

:pp

$PP \rightarrow P NP;$

:clause

$S \rightarrow PP* NP PP* VP PP*;$

Esta es una gramática regular simplificada en 3 niveles cada uno de los cuales es capaz de derivar reglas de la forma especificada usando como base lo derivado en niveles anteriores. La gramática regular que usa Cass por defecto se define en nueve niveles de derivación siendo ampliable por el usuario, añadiendo niveles o introduciendo nuevos patrones. Cada patrón se define con una categoría seguida de una expresión regular terminada con un punto y coma.

Sin embargo, aunque el sistema expuesto permite una definición flexible, la gramática regular que resulta requiere unas cinco páginas de código para poder expresar todos los casos posibles que deberá considerar el analizador. Y su adaptación a otros dominios particulares requiere la realización de un estudio lingüístico para definir los nuevos patrones necesitados. Sin embargo, una vez más, aparece la compleja y tediosa tarea de definición manual de reglas.

[Dae96] presenta un generador de etiquetadores gramaticales basado en corpus capaz de construir automáticamente un etiquetador para etiquetar textos nuevos. Para ello utiliza un sistema de aprendizaje basado en memoria que es una forma supervisada de aprendizaje inductivo basado en ejemplos. Los ejemplos se representan como un vector de características con una etiqueta de categoría asociada. Durante el entrenamiento, se presenta un conjunto de ejemplos (el conjunto de entrenamiento) en forma incremental al clasificador y se cargan en memoria. En la prueba se presenta al sistema un conjunto de formas desconocidas. Para cada forma se calcula su distancia a todos los ejemplos residentes en memoria y la que alcanza la distancia más corta se elige como categoría predefinida para la forma de la prueba. Esta aproximación se basa en la suposición de que el razonamiento es una reutilización de experiencias almacenadas más que la aplicación de un conocimiento (mediante reglas).

Estudiando el sistema Cass y basándonos en la idea del generador de etiquetadores anterior nos hemos planteado buscar un método que pueda evitar la complejidad en la definición de la gramática de partida. Para ello definiremos un sistema basado en niveles que a diferencia del Cass no tendrá como misión la realización

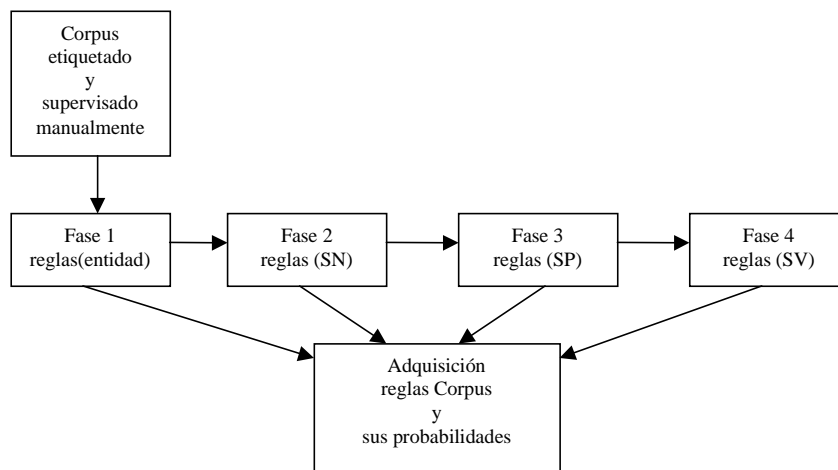


Fig 1. Proceso de adquisición automática de reglas gramaticales.

de un análisis sino la derivación de las reglas gramaticales que permitan generar una gramática para un analizador posterior. El sistema se basa en el empleo de un autómata finito definido mediante patrones al que se le introduce como entrada un texto etiquetado y corregido manualmente y obtendrá como salida una serie de reglas gramaticales que concuerdan con los patrones introducidos y las probabilidades de cada una de ellas, para permitir la realización posterior de análisis sintácticos estadísticos sobre otros tipos de textos.

A diferencia de cualquier analizador, el sistema extractor de reglas gramaticales maneja una gramática de entrada simple puesto que no debe analizar sino simplemente extraer la información de un texto que se sabe que es correcto (es decir, tiene una cobertura y precisión de etiquetado léxico del 100%). Esto nos permite manejar un método de especificación de patrones sencillo y flexible para la adquisición posterior de las reglas.

3. El método.

3.1. Descripción.

En esta sección pasamos a detallar el sistema propuesto para la adquisición automática de reglas gramaticales mediante un mecanismo que está basado en las expresiones regulares. El sistema utiliza como entrada un texto etiquetado y corregido manualmente y obtiene como resultado una serie de reglas gramaticales con las probabilidades de cada una de ellas, de acuerdo con las expresiones regulares introducidas (tipos de componentes que deseamos encontrar).

El algoritmo para la obtención de reglas gramaticales se aplica sobre la salida de un etiquetador. Trabajaremos sobre el corpus etiquetado léxicamente utilizado dentro de

CRATER. Este corpus contiene el manual de la International Telecommunications Union CCITT, conocido como *The Blue Book*, en su versión en castellano. En concreto, para la obtención de resultados, hemos utilizado un fragmento formado por 250 oraciones (aproximadamente 9600 palabras) donde cada palabra está formada por una terna donde aparece la palabra original, el lema (raíz) y su etiqueta, que además de la categoría gramatical contiene su información morfológica (número, género y persona).

El método se compone de varias fases que tratan de identificar componentes distintos, estando cada una de ellas basada en las anteriores. Para nuestro caso hemos planteado cuatro fases en las que identificaremos cuatro componentes distintos, pudiéndose plantear otro número de fases según los algoritmos de análisis planteados y los componentes que nos interese encontrar. En cada fase, mediante expresiones regulares, se especifica el conjunto de etiquetas que pueden llegar a formar parte de un componente, teniendo en cuenta que los elementos no terminales incluidos deben de haber sido derivados en fases anteriores. En la figura 1 se muestra gráficamente la secuencia de fases que conforman el proceso final.

El algoritmo, por tanto, se basará en una serie de autómatas finitos que representan cada uno de los componentes a identificar en cada fase. La entrada para el primer nivel será la oración etiquetada léxicamente y tendrá como salida la misma oración a la que se le añadirán las etiquetas sintácticas de los componentes detectados. Esta salida será la entrada para el siguiente nivel y así sucesivamente. Las etiquetas que no sean aceptadas por el autómata finito que define la expresión regular quedarán como una etiqueta aislada o pertenecerán a otro componente superior.

3.2. Propuesta Previa.

Para el caso concreto de nuestro estudio hemos definido las siguientes expresiones regulares que denotan los componentes a tratar en cada fase:

1. *entidad*: [*tratamiento* *, *nompropio* *] +
2. *sn*: [*determinante* *, *nombre* *, *entidad* *, *adjetivo* *] +
3. *sp*: [*preposicion* , *sn*]
4. *sv*: [*verbo*] + [*sn* * , *sp* *] *

La primera fase consistirá en la identificación de reglas gramaticales para derivar entidades. En esta fase tratamos de aprender aquellas entidades que estén formadas exclusivamente por etiquetas que representan tratamiento (por ejemplo Dña., Exma., Sr., etc.) y nombres propios (topónimos, antropónimos, etc.). Para el analizador parcial que queremos obtener no es necesario tener en cuenta si las entidades están separadas por preposiciones, conjunciones o artículos, ya que esto se determinará en módulos posteriores al análisis parcial.

Para identificar si una etiqueta puede formar parte de una entidad se realiza una lista Prolog con las etiquetas de los componentes posibles de una entidad. Cuando analizamos una etiqueta, el algoritmo comprueba si forma parte de esta lista. Si no se encuentra, la etiqueta se escribe directamente en la lista de salida, es decir, no hemos encontrado un componente del tipo entidad. En caso contrario, el algoritmo la inserta en una nueva lista que representa la entidad y analiza las etiquetas siguientes hasta que se encuentre una que no pertenezca a la lista de etiquetas del componente entidad.

Cuando se ha detectado un componente del tipo entidad se inserta la regla correspondiente, en la que aparece el tipo de componente (entidad) y las etiquetas de los subcomponentes que lo forman. En la figura 2 se muestra alguna de las reglas obtenidas y sus equivalentes en DCG utilizando las etiquetas del proyecto CRATER [San95]².

Regla Prolog	Regla DCG
r(ent(['NPTOP','NPTOS']))	Ent → NPTOP NPTOS
r(ent(['NPTOS']))	Ent → NPTOS

Fig 2. Equivalencia Prolog – DCG.

Cuando se llega a este punto, además, se intercalan en la lista de salida intermedia los

² Por ejemplo, NPTOS: nombre propio topónimo singular o nombre colectivo; NPTOP: nombre propio topónimo plural o nombre colectivo.

componentes entidad detectados; esta lista servirá de entrada para la fase posterior. En el siguiente ejemplo se muestra un fragmento de la lista de entrada y la lista de salida intermedia:

Lista de entrada:

..., w(la, el, ARTDFS), w(Organización, organización, NCFS), w(de, de, PREP), w(las, el, ARTDFP), w(Naciones Unidas, Naciones Unidas, NPTOP), w(y, y, CC), w(la, el, ARTDFS), w(Unión, unión, NCFS), w(Internacional, internacional, ADJGFS), ...

Lista intermedia de salida:

..., w(la, el, ARTDFS), w(Organización, organización, NCFS), w(de, de, PREP), w(las, el, ARTDFP), **entidad([w(Naciones Unidas,Naciones Unidas, NPTOP)], _1873, ENTIDAD)**, w(y, y, CC), w(la, el, ARTDFS), w(Unión, unión, NCFS), w(Internacional, internacional, ADJGFS), ...

Como se puede observar en la lista de salida aparecerán las palabras aisladas con los tres argumentos ya mencionados y los componentes de tipo entidad identificados, cuyos argumentos se corresponden con: la lista de etiquetas que forman la entidad, una variable Prolog sin instanciar y la nueva etiqueta ENTIDAD que hemos añadido al conjunto de etiquetas para que pueda ser interpretada en las expresiones regulares de niveles superiores (se mantienen los tres argumentos para conservar el paralelismo entre los símbolos terminales y los no terminales).

La segunda fase trata de identificar los sintagmas nominales para obtener las reglas asociadas. Para ello, especificamos que un sintagma nominal estará formado por un conjunto de:

- símbolos terminales: determinantes (artículos, cuantificadores, etc.), nombres, adjetivos (demostrativos, posesivos, ordinales, etc.).
- símbolos no terminales: entidades.

El funcionamiento del algoritmo para detectar los sintagmas nominales es similar al tratamiento de las entidades. Se analiza una etiqueta y se comprueba que forme parte de la lista de etiquetas que constituyen un sintagma nominal. A continuación se procede de forma análoga al análisis de las entidades. En la lista de salida aparecerá el nuevo tipo de componente (sintagma nominal) con sus argumentos: la lista de palabras formando el sintagma nominal, una variable Prolog sin instanciar y la nueva etiqueta SN que se ha añadido al conjunto de etiquetas.

```

r(entidad(['NPTOP']), 2, 0.05).
r(entidad(['NPTOS']), 38, 0.95).
r(sn(['ARTDMS', 'NCMS', 'CARDXS']), 1, 0.00040666937779585).
r(sn(['ARTDFP', 'ENTIDAD']), 2, 0.0008133387555917).
r(sn(['ARTDMP', 'NCMP', 'NCMS']), 1, 0.00040666937779585).
r(sn(['DMDXFS', 'NCFS']), 1, 0.00040666937779585).
r(sn(['ENTIDAD', 'ARCAMS', 'NCMS', 'ADJGMS']), 1, 0.00040666937779585).
r(sn(['ARTDMP', 'NCMP', 'ADJGMS']), 1, 0.00040666937779585).
r(sn(['NCFS', 'CODE', 'NCFS']), 2, 0.0008133387555917).
r(sn(['ENTIDAD', 'CARDXP']), 1, 0.00040666937779585).
r(sn(['ARTDMP', 'ADJGMP', 'NCMP']), 2, 0.0008133387555917).
r(sv(['NCFS', 'ARTDMP', 'NCMP', 'ADJGMP']), 1, 0.00040666937779585).
r(sn(['NCMS', 'ADJGMS', 'ADJGMS']), 11, 0.00447336315575437).
r(sn(['CARDXP', 'NCMP']), 11, 0.00447336315575437).
r(sn(['ARTDFP', 'NCFP', 'ADJGFP']), 17, 0.00691337942252948).
r(sn(['ARTDFS', 'NCFS', 'ADJGFS']), 47, 0.01911346075640504).
r(sn(['ARTDMP', 'NCMP']), 74, 0.03009353395689305).
r(sn(['ARTDFS', 'NCFS']), 223, 0.09068727124847499).
r(sn(['NCMS']), 269, 0.10939406262708418).
r(sp(['PAL', 'SN']), 13, 0.00984102952308857).
r(sp(['PREP', 'SN']), 1189, 0.90007570022710066).
r(sp(['PDEL', 'SN']), 119, 0.09008327024981075).
r(sv(['VLPS3P', 'SN']), 1, 0.00176678445229682).
r(sv(['VLPS3S', 'SP', 'SP', 'SP']), 1, 0.00176678445229682).
r(sv(['VSCI3S', 'SN', 'SP']), 1, 0.00176678445229682).
r(sv(['VLPXMP', 'SN']), 1, 0.00176678445229682).
r(sv(['VLPXMP', 'SN', 'SP', 'SP', 'SP']), 1, 0.00176678445229682).
r(sv(['VLPXFS', 'SN']), 1, 0.00176678445229682).
r(sv(['VEPI3S', 'SN', 'SP']), 1, 0.00176678445229682).
r(sv(['VLPXFS', 'SP']), 11, 0.01943462897526502).
r(sv(['VLPXMS', 'SP']), 17, 0.03003533568904593).
r(sv(['VLPI3S', 'SN', 'SP']), 18, 0.03180212014134276).
r(sv(['VSPi3S', 'SN']), 18, 0.03180212014134276).
r(sv(['VLPI3S', 'SP', 'SP']), 11, 0.01943462897526502).
r(sv(['VLPI3S', 'SN']), 26, 0.04593639575971731).
r(sv(['VLPI3P', 'SP', 'SP']), 7, 0.01236749116607774).
r(sv(['VLPI3S', 'SP']), 21, 0.03710247349823321).
...

```

Fig. 3. Algunas reglas aprendidas con la primera definición de patrones.

La tercera fase consiste en detectar los sintagmas preposicionales y obtener sus reglas asociadas. Un sintagma preposicional está formado por una preposición (símbolo terminal) seguida de un sintagma nominal (símbolo no terminal). En la lista de salida aparecerá el nuevo tipo de componente sintagma preposicional con la nueva etiqueta SP.

La última fase consiste en obtener las reglas gramaticales para los sintagmas verbales. Un sintagma verbal está formado por:

- un símbolo terminal: verbo
- un conjunto de símbolos no terminales: complementos del verbo (sintagmas nominales y sintagmas preposicionales).

Se ha especificado que los sintagmas verbales deben empezar por un verbo que puede ir acompañado o no de los complementos del verbo (sintagmas nominales y sintagmas preposicionales). Cuando se analiza una etiqueta se comprueba que sea un verbo. Si es así, comienza el sintagma verbal hasta que se encuentre un componente que no pertenezca a la lista de complementos del verbo.

El módulo final del algoritmo cuenta las ocurrencias de cada regla calculando su probabilidad respecto al total de reglas derivadas sobre el mismo constituyente, siendo

la salida final del algoritmo una lista con cada regla, sus ocurrencias y la probabilidad, tal como se muestra en el ejemplo:

```
r(sn(['ARTDMP', 'NCMP']), 74, 0.0300935).
```

Estas probabilidades se interpretan como la probabilidad de expandir un constituyente, por ejemplo un SN, usando una regla particular frente a cualquiera de las otras reglas que podrían usarse para expandir este constituyente.

3.3. Resultados experimentales.

Como resultado del algoritmo descrito anteriormente junto con las expresiones regulares definidas se obtiene una serie de reglas gramaticales para cada tipo de componente encontrado (entidad, sn, sp y sv) y los subcomponentes que lo forman, indicando la probabilidad para cada regla. Como salida de las 250 oraciones se han obtenido 335 reglas, algunas de las cuales se muestran en la figura 3.

3.4. Análisis de resultados y detección de problemas.

Una vez finalizado este primer experimento se efectúa una revisión manual de las reglas derivadas detectando los siguientes problemas:

- El conjunto de etiquetas de partida es demasiado extenso (aproximadamente 500).

Esto nos lleva a derivar reglas equivalentes, que sin aportar información adicional, están alterando el resultado. En el siguiente ejemplo vemos dos reglas equivalentes que únicamente varían en las formas de los tiempos verbales (tercera persona plural del presente subjuntivo y tercera persona singular del presente indicativo).

```
r(sv(['VLPS3P', 'SN']), 1, 0.001766784452).
r(sv(['VLPIS3', 'SN']), 26, 0.04593639575).
```

- Se ha detectado la existencia de algunos errores al identificar los límites de los constituyentes derivados:

Los errores en la limitación por defecto que se han encontrado son:

- Problemas de coordinación de elementos: En esta primera aproximación no se ha contemplado la obtención de reglas que permitan la coordinación de los elementos, lo que provoca que un constituyente coordinado se identifique fragmentado en dos (o más) constituyentes.
- Tratamiento de los pronombres: los pronombres son núcleos del sintagma nominal y deben incluirse en la lista de constituyentes de sintagmas nominales. Por simplificación, no se han incluido en esta primera experiencia. Esto ha provocado que las reglas que reconocen sintagmas nominales encabezados por pronombres no hayan sido derivadas (o estén incompletas y sin núcleo).
- Tratamiento de las formas no personales del verbo: el infinitivo en tanto que sustantivo verbal, el participio en tanto que adjetivo verbal y el gerundio en tanto que adverbio podrían formar parte de los sintagmas nominales, aparte de su función verbal. Por simplificación tampoco han sido incluidos en esta primera aproximación, por lo que se han derivado algunas reglas incompletas.
- Tratamiento de los adverbios: igual que en los casos anteriores, el adverbio no ha sido tratado, provocando el mismo problema.
- Tratamiento de los verbos: es necesario redefinir el patrón para la adquisición de reglas de sintagmas verbales para poder incluir los verbos compuestos y perífrasis verbales. Además, no está claramente definido el problema de la limitación del sintagma verbal.

Por otra parte, se han encontrado los siguientes problemas de limitación de constituyentes por exceso:

- Unificación de sintagmas nominales: algunos sintagmas nominales pueden aparecer unidos debido a que no existe ninguna palabra de separación entre ellos. Esto ocurre por ejemplo en la frase *...hay que tener en cuenta los constituyentes...*

3.5. Replanteamiento y solución de problemas.

Gracias a la flexibilidad del sistema, tras la revisión manual efectuada se han propuesto algunos cambios a la plantilla de patrones anterior para solucionar algunos de los problemas derivados:

- Construimos un interfaz que reduce las 500 etiquetas disponibles en el etiquetador Xerox empleado para etiquetar léxicamente el corpus de entrenamiento a 21 etiquetas eliminando la información morfológica de género y número así como los tiempos verbales. Dicha información se introduce añadiendo argumentos en los hechos que se obtienen en la lectura de la cadena de entrada. De esta forma usaremos el dato para reforzar posteriormente la concordancia de los elementos que se relacionan, a la vez que conseguimos una mayor precisión en los resultados al disminuir el dominio del problema. Un ejemplo de la transformación de este interfaz es:

```
NCFP → ncomun(pl, fem)
NCFS → ncomun(sing, fem)
NCMP → ncomun(pl, masc)
NCMS → ncomun(sing, masc)
```

- Resolvemos, en parte, el problema del límite en los sintagmas nominales consecutivos mediante la definición de un método de concordancia en género y número de los elementos que forman parte de un mismo sintagma nominal. De esta forma limitamos el problema de los sintagmas nominales unidos al caso en el que ambos tengan el mismo género y el mismo número, es decir, hemos solucionado el problema en un 75% de las ocurrencias (aproximadamente).

- Exceptuando aquellos que van dentro de perífrasis verbales hemos incluido los infinitivos como elementos del sintagma nominal ya que, a pesar de su forma verbal, se reinterpreta léxicamente como forma nominal tal y como se insinúa en [Bos91] basándose en los trabajos de [Sal82]. En cuanto al participio, exceptuando los casos de verbos compuestos y de perífrasis verbales, lo consideramos adjetivo por su similitud funcional.

```

r(entidad([nprop]), 131, 1.0).
r(sn([art,adjSimple,ncomun,adjSimple]), 3, 0.00101660454083362).
r(sn([ncomun,adjSimple,adjSimple]), 24, 0.00813283632666893).
r(sn([pron,ncomun]), 5, 0.00169434090138936).
r(sn([art,ncomun,vpart,adjSimple]), 1, 0.00033886818027787).
r(sn([adjSimple,ncomun,adjSimple]), 14, 0.00474415452389021).
r(sn([art,ncomun,ncomun]), 6, 0.00203320908166723).
r(sn([letra]), 199, 0.06743476787529651).
r(sn([adjSimple,adjSimple,ncomun]), 4, 0.00135547272111149).
r(sn([art,ncomun,adjSimple,vpart]), 15, 0.00508302270416808).
r(sn([vinf,adjSimple]), 4, 0.00135547272111149).
r(sn([ncomun,vpart]), 40, 0.01355472721111488).
r(sn([vinf,vpart]), 4, 0.00135547272111149).
r(sn([art,ncomun,adjSimple,adjSimple]), 20, 0.00677736360555744).
r(sn([vinf]), 68, 0.02304303625889529).
r(sn([ncomun,adjSimple,vpart]), 10, 0.00338868180277872).
r(sn([entidad]), 91, 0.03083700440528634).
r(sn([art,entidad]), 34, 0.01152151812944764).
r(sn([ncomun,adjSimple]), 203, 0.06879024059640799).
r(sn([pron]), 25, 0.0084717045069468).
r(sn([art,adjSimple,ncomun]), 24, 0.00813283632666893).
r(sn([adjSimple,ncomun]), 88, 0.02982039986445273).
r(sn([art,ncomun,vpart]), 27, 0.00914944086750254).
r(sn([ncomun]), 723, 0.24500169434090138).
r(sn([art,ncomun]), 552, 0.1870552355133853).
r(sn([art,ncomun,adjSimple]), 231, 0.07827854964418841).
r(sv([verbo,vinf,vpart]), 2, 0.00552486187845304).
r(sv([verbo,vinf]), 22, 0.06077348066298343).
r(sv([verbo,vger]), 3, 0.00828729281767956).
r(sv([verbo,vpart]), 15, 0.04143646408839779).
r(sv([verbo,vreflex]), 24, 0.06629834254143646).
r(sv([verbo]), 294, 0.81215469613259668).
r(sp([prepSimple,sn]), 1376, 1.0).
...

```

Fig. 4. Algunas reglas aprendidas con la segunda definición de patrones

- Se incluyen los pronombres en la lista de constituyentes de los sintagmas nominales.

- Modificamos el patrón de adquisición de reglas para los sintagmas verbales permitiendo la identificación de perífrasis verbales a la vez que lo limitamos a la obtención del núcleo del sintagma verbal y sus auxiliares. De esta forma, los sintagmas nominales y preposicionales que complementan al sintagma verbal no se derivan en este conjunto de reglas, por lo que no se incluirá en el análisis parcial posterior y se dejará para tratar en posteriores fases utilizando técnicas específicas como, por ejemplo, las planteadas en [Pal97].

De esta forma, los nuevos patrones de expresiones regulares correspondientes a los nuevos niveles son:

1. *entidad*: [tratamiento*, nompropio*]⁺
2. *sv*: [verbo]⁺ [infinitivo*, participio*, gerundio*]^{*}
3. *sn*: [artículo*, nomcomún*, entidad*, adjetivo*, infinitivo*, participio*, pronombre*]⁺
4. *sp*: [preposicion, sn]

Nótese que se ha hecho una variación en el orden de la identificación de componentes. Así se obtendrán primero los sintagmas verbales, para que cualquier infinitivo o gerundio que no esté junto a un verbo pueda ser identificado posteriormente como parte de un sintagma nominal.

Como consecuencia de la reducción en el conjunto de etiquetas hemos obtenido ahora un total de 72 reglas distintas que representan el conocimiento lingüístico aprendido del corpus de entrenamiento incluyendo ahora sólo la información pertinente. Algunas de las reglas obtenidas se pueden ver en la figura 4.

4. Conclusiones

El sistema que hemos desarrollado nos permite obtener de forma automática un conjunto de reglas sintácticas válidas para construir la gramática de un analizador parcial. Para ello nos basamos en un sistema de definición de patrones sintácticos por niveles mediante expresiones regulares que son capaces de aprender reglas desde un corpus de entrenamiento. Sobre este sistema hemos realizado una primera propuesta de patrones cuyos resultados hemos analizado y evaluado de forma manual posteriormente. Basándonos en este estudio se han redefinido los patrones y las fases de análisis obteniendo en una segunda propuesta una gramática idónea para los objetivos planteados.

Aun así, de los problemas detectados en la primera definición de patrones, el problema del tratamiento de los adverbios sigue sin estar solucionado en la segunda propuesta analizada. Aprovechando la flexibilidad que nos permite

el sistema, planteamos como trabajo futuro la inclusión de un módulo que contemple los sintagmas adverbiales y sus relaciones con constituyentes superiores.

Referencias

- [Abn94] Abney, S., and Rooth, M. *The B7 Project: Partial Parsing and the Acquisition of Lexical Syntax and Semantics. Project Proposal.* <http://www.sfs.nphil.uni-tuebingen.de/~abney/b7home.html>, 1994.
- [Abn96] Abney, S. Statistical Methods and Linguistics. In *The Balancing Act.* Judith Klavans and Philip Resnik, Eds., MIT Press, Cambridge, M.A. 1996.
- [Abn97] Abney, S. *The SCOL Manual. Version 0.1b.* <http://www.sfs.nphil.uni-tuebingen.de/~abney/scol.ps.gz>, 1997.
- [Bos91] Bosque, I. *Las categorías gramaticales. Relaciones y diferencias.* Síntesis, Madrid, 1991.
- [Bri97] Brill, E., and Mooney R.J. An Overview of Empirical Natural Language Processing. *American Association for Artificial Intelligence Magazine* (Winter 1997).
- [Dae96] Daelemans, W., and Zavrel, J. MBT: A Memory-Based Part of Speech Tagger- Generator. In *Proceedings of WVLC* (Copenhagen, 1996).
- [Pal97] Palmer, D.D., and Hearst, M.A. Adaptive multilingual sentence boundary disambiguation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics, ACL'97* (1997).
- [Sal82] Salvi, G. L'Infinito articolato e la struttura del SN. *Rivista di Grammatica Generativa*, 6 (1982).
- [San95] Sánchez León, F. *Spanish tagset for the CRATER project.* Informe Interno. Laboratorio de Lingüística Informática, Universidad Autónoma de Madrid, 1995.