Procesamiento del Lenguaje Natural, nº39 (2007), pp. 5-12 recibido 18-05-2007; aceptado 22-06-2007

# Desarrollo de un analizador sintáctico estadístico basado en dependencias para el euskera

**Kepa Bengoetxea, Koldo Gojenola**
Universidad del País Vasco UPV/EHU
Escuela Universitaria de Ingeniería Técnica
Industrial de Bilbao
{kepa.bengoetxea, koldo.gojenola}@ehu.es

**Resumen:** Este artículo presenta los primeros pasos dados para la obtención de un analizador sintáctico estadístico para el euskera. El sistema se basa en un *treebank* anotado sintácticamente mediante dependencias y la adaptación del analizador sintáctico determinista de Nivre *et al*. (2007), que mediante un análisis por desplazamiento/reducción y un sistema basado en aprendizaje automático para determinar cuál de 4 opciones debe realizar, obtiene un único análisis sintáctico de la oración. Los resultados obtenidos se encuentran cerca de los obtenidos por sistemas similares.
**Palabras clave:** Análisis sintáctico. Análisis basado en dependencias. Treebank.

**Abstract:** This paper presents the first steps towards a statistical syntactic analyzer for Basque. The system is based on a syntactically dependency annotated treebank and an adaptation of the deterministic syntactic analyzer of Nivre et al. (2007), which relies on a shift/reduce deterministic analyzer together with a machine learning module that determines which one of 4 analysis options to take, giving a unique syntactic dependency analysis of an input sentence. The results are near to those obtained by similar systems.
**Keywords:** Syntactic analysis. Dependency-based analysis. Treebank.

## *1 Introducción*

Este artículo presenta los primeros pasos dados para la obtención de un analizador sintáctico estadístico para el euskera. El sistema se basa en un treebank anotado sintácticamente mediante dependencias y la adaptación del analizador sintáctico determinista MaltParser (Nivre et al., 2007), que mediante un análisis por desplazamiento/reducción y un sistema basado en aprendizaje automático para determinar, en cada paso de análisis, cuál de 4 opciones debe realizar, obtiene un único análisis sintáctico de la oración. Los resultados obtenidos se encuentran cerca de otros sistemas similares.

En el resto del artículo presentaremos en el apartado 2 el treebank utilizado (3LB) que será la base del analizador sintáctico, y las modificaciones realizadas para su procesamiento de manera automática. El apartado 3 contextualiza los sistemas de análisis sintáctico estadístico, presentando el sistema elegido para este trabajo, que es el analizador determinista Maltparser. En la sección 4 se presentan los experimentos realizados junto con los resultados obtenidos. La sección 5 compara el trabajo realizado con sistemas similares que han sido desarrollados. El artículo acaba presentando las principales conclusiones y líneas futuras de trabajo.

## *2 3LB: un treebank anotado sintácticamente para el euskera*

El proyecto 3LB desarrolló corpus anotados a nivel morfológico y sintáctico para el catalán, euskera y español (Palomar et al., 2004).

La anotación para el catalán y español está basada en constituyentes, mientras que el euskera está anotado mediante dependencias (Carroll, Minnen y Briscoe, 1998). Seguidamente se presentarán primero las

ISSN: 1135-5948 

```
@@00,06,2,1201,6
Ika-mika     baten    ostean,   funtzionarioak   14:00etan  itzultzeko   esan      zien.
(discusión) (de una) (después),(el funcionario) (a las 14)  (volver)    (decir) (él a ellos/pasado)
Después de una discusión, el funcionario les dijo que volvieran a las 14:00.

                meta      (-,            root,       esan)
                ncmod     (gen_post_ine, esan,       Ika-mika)
                detmod    (-,            Ika-mika,   baten_ostean)
                ncsubj    (erg,          esan,       funtzionarioak)
                ncmod     (ine,          itzultzeko, 14:00etan)
                xcomp_obj (konp,         esan,       itzultzeko)
                auxmod    (-,            esan,       zien)
```

Figura 1: Ejemplo de anotación de una oración.

características generales del treebank original (apartado 2.1) y la adaptación que se hizo del treebank para convertirlo a un formato apropiado para el análisis automático (apartado 2.2).

## 2.1 El treebank 3LB para el euskera

El corpus 3LB (Palomar et al., 2004) contiene 57.000 palabras anotadas sintácticamente. Las características del euskera, como por ejemplo el orden libre de constituyentes de la oración, aconsejaron realizar una anotación mediante dependencias, de manera similar a la realizada para idiomas como el checo (Hajic, 1999), aunque también planteada para idiomas de orden menos libre como el inglés (Jarvinen y Tapanainen, 1998).

La figura 1 muestra un ejemplo de anotación de una oración en el corpus 3LB. Básicamente, la anotación indica el tipo de dependencia (*meta*, *ncsubj*, …) seguida de tres atributos que representan:

- Información morfosintáctica útil como es el caso, o el tipo de oración subordinada (*konp*[1] en el ejemplo). Aunque la figura muestra que la anotación incluye una mínima información morfosintáctica, en general, la anotación está basada en palabras. Este hecho supuso un problema, ya que los analizadores sintácticos estadísticos requieren el uso de rasgos morfosintácticos (categoría, número, caso, …) no presentes en este corpus original.
- Núcleo de la dependencia (con el valor especial *root* para indicar el núcleo de la oración).
- Elemento dependiente.

[1] Oración subordinada completiva.

## 2.2 Adaptación del treebank

La anotación original del treebank para el euskera, válida lingüísticamente, plantea varios problemas a la hora de ser usada en un tratamiento computacional:

- Fenómenos como la aparición de palabras repetidas en una misma oración requieren la explicitación del elemento oracional correspondiente a cada aparición de la palabra, no presente en la anotación original
- Elementos no explícitos. En la anotación original se permitió la anotación de elementos nulos correspondientes a fenómenos como la elipsis o coordinación. Sin embargo, la gran mayoría de los analizadores basados en dependencias actuales no admite la aparición de elementos que no corresponden a palabras de la oración.
- Ambigüedad morfosintáctica. La anotación original se hizo enlazando palabras entre sí. Esta alternativa tiene el inconveniente de que, siendo cada palabra morfológicamente ambigua (cada palabra tiene una media de 2,81 interpretaciones), no se conoce con certeza cuál es la interpretación correcta. Aunque el tipo de dependencia que une dos palabras proporciona información útil para la desambiguación (por ejemplo, la dependencia "ncsubj" generalmente une el núcleo de un sintagma nominal, normalmente de categoría nombre, con un verbo), hay un alto grado de ambigüedad no resoluble automáticamente. La figura 1 muestra que las palabras no contienen ningún tipo de anotación morfosintáctica, a excepción de las dependencias.
- Términos multipalabra. Al etiquetar el corpus, los lingüistas no disponían de una guía sistemática para la anotación de

| P | Forma | Lema | Cat | Cat+subcat | Info | Núcleo | Dependencia |
|---|---|---|---|---|---|---|---|
| 1 | Ika-mika | Ika-mika | IZE | IZE_ARR | ABS\|MG | 7 | ncmod |
| 2 | baten_ostean | bat | IZE | IZE_ARR | DEK\|GEN_oste_INE\|NUMS\|MUGM\|POS | 1 | ncmod |
| 3 | , | , | PUNT | PUNT_KOMA | _ | 2 | PUNC |
| 4 | funtzionarioak | funtzionario | IZE | IZE_ARR | ERG\|NUMS\|MUGM | 7 | ncsubj |
| 5 | 14:00etan | 14:00 | DET | DET_DZH | NMGP\|INE\|NUMP\|MUGM | 6 | ncmod |
| 6 | itzultzeko | itzuli | ADI | ADI_SIN | ADIZE\|KONPL\|ABS\|MG | 7 | xcomp_obj |
| 7 | esan | esan | ADI | ADI_SIN | PART\|BURU | 0 | ROOT |
| 8 | zien | *edun | ADL | ADL | B1\|NR_HURA\|NK_HARK\|NI_HAIEI | 7 | auxmod |
| 9 | . | . | PUNT | PUNT_PUNT | _ | 8 | PUNC |

Figura 2: Ejemplo de anotación de una oración.

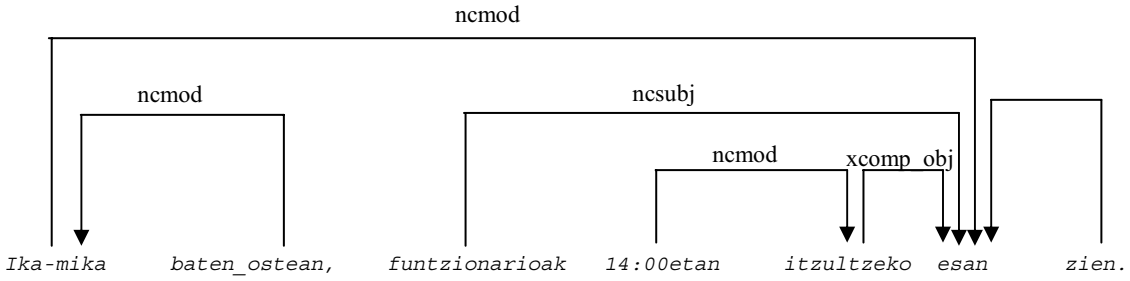


Figura 3: Representación gráfica del árbol de dependencias.

estos elementos, que incluyen elementos como entidades, postposiciones complejas o locuciones. Esto dio lugar a que sea difícil emparejar las palabras del treebank con las de la oración original. Como ejemplo, la figura 1 muestra que la postposición compleja "*baten ostean*" se ha agrupado en una sola unidad.

Por estos motivos se hizo imprescindible reetiquetar el corpus para obtener una versión tratable computacionalmente. Aunque se realizaron programas de ayuda al reetiquetado, este proceso fue muy costoso, al ser en su mayor parte manual, y exigió la revisión completa del treebank. Las figuras 2 y 3 muestran la oración anterior etiquetada en un formato de dependencias utilizable computacionalmente y su representación gráfica. El formato elegido es el de la conferencia CoNLL[2] (CoNLL 2007), que tiene las siguientes características:

- Componentes explícitos. Todas las relaciones deben ser de palabra a palabra, es decir, no se permite eliminar o añadir elementos a la oración en el análisis.
- Es suficientemente versátil para permitir su conversión a otros formatos de manera automática, como el formato Penn (Marcus, Santorini y Marcinkiewiecz, 1993) o el formato aceptado por el parser de (Collins et al. 1999).

La figura 2 contiene un ejemplo de la sentencia en el nuevo formato. Este formato contiene ocho campos: posición (P), forma, lema, categoría (*coarse postag*), categoría + subcategoría, información morfosintáctica, identificador del núcleo y relación de dependencia.

## *3 Análisis sintáctico estadístico*

La popularidad de los Treebanks está ayudando al desarrollo de analizadores sintácticos estadísticos que empezó con el Penn Treebank para el inglés (Marcus, Santorini y Marcinkiewiecz, 1993), para el que se han desarrollado parsers de referencia (Collins, 1996; Charniak, 2000), que marcan el estado del arte actual. Aunque las características del inglés llevaron a una anotación inicial basada en constituyentes, diversos factores, fundamentalmente la extensión a idiomas de características muy diferentes al inglés y también la dificultad de evaluación de las estructuras jerárquicas subyacentes, han llevado a desarrollar modelos sintácticos basados en dependencias.

El apartado 3.1 examinará brevemente los analizadores sintácticos basados en

[2] Computational Natural Language Learning.

dependencias. En el punto 3.2 se describirá el analizador sintáctico de Nivre et al. (2007) que ha sido usado en el presente trabajo.

### 3.1 Análisis sintáctico basado en dependencias

Los analizadores sintácticos basados en dependencias han sido utilizados en diversos trabajos, con propuestas que van desde analizadores que construyen directamente estructuras de dependencias (Jarvinen y Tapanainen 1998, Lin 1998) hasta otras que se basan en las tradicionales estructuras de constituyentes permitiendo adicionalmente la extracción de dependencias (Collins 1999; Briscoe, Carroll y Watson, 2006).

Entre los analizadores estadísticos basados en dependencias podemos citar los experimentos realizados por (Eisner, 1996) y los trabajos realizados para el turco (Eryiğit y Oflazer, 2006), que comparte con el euskera la propiedad de ser un idioma aglutinativo. En general, los últimos años este tema ha sido avivado por la competición realizada en la conferencia CoNLL[3] sobre analizadores de dependencias (CoNLL, 2006, 2007), en la que se plantea el reto de utilizar diferentes parsers para analizar un conjunto de treebanks de un amplio abanico de idiomas.

### 3.2 Maltparser: un analizador sintáctico estadístico determinista

El analizador sintáctico determinista de Nivre et al. (2007) es un sistema independiente del lenguaje que permite inducir un parser o analizador sintáctico a partir de un treebank, usando conjuntos de datos de entrenamiento limitados. El analizador se basa en:

- Algoritmos deterministas para análisis de dependencias. Mediante un análisis por desplazamiento/reducción y un sistema basado en el uso de una pila y una cadena de entrada.
- Modelos de características basados en historia (*History-based feature models*) para predecir la acción a realizar. En este algoritmo concreto, el sistema debe elegir entre 4 opciones (enlazar dos palabras con un arco hacia la izquierda, ídem con arco hacia la derecha, reducir o desplazar), y para ello hace uso de los rasgos de la pila y/o de la cadena de entrada. Aplicando sucesivamente este paso, se obtiene un único análisis sintáctico de la oración.
- Técnicas de aprendizaje automático discriminativas para enlazar historias con acciones. En este momento el sistema permite utilizar dos de las alternativas de aprendizaje automático más exitosas: aprendizaje basado en memoria (*Memory Based Learning*, Daelemans y Van den Bosch, 2005) y *Support Vector Machines* (SVM, Chang y Lin, 2001).

Este analizador ha sido probado con multitud de idiomas de diversa tipología, obteniendo resultados que se acercan al estado del arte para el inglés, que es tomado generalmente como referencia y punto de comparación. En la competición CoNLL de 2007, una versión de este sistema ha quedado en primera posición, de un total de 20 sistemas presentados.

## *4 Experimentos y resultados*

En este apartado vamos a presentar los experimentos realizados junto con los resultados que se han obtenido.

El primer paso consiste en seleccionar los atributos utilizados para el análisis sintáctico. Aunque el uso de una mayor cantidad de información puede en principio ayudar a mejorar los resultados, el tamaño del corpus usado (57.000 palabras) es pequeño, por lo que se pueden presentar problemas de *data sparseness*.

El analizador usado permite especificar distintos tipos de información a utilizar para el entrenamiento, distinguiendo:

- Información léxica. Se podrá usar tanto la forma como el lema de cada palabra.
- Información categorial. Se puede seleccionar tanto la categoría sintáctica (nombre, adjetivo, verbo, …) como la subcategoría (nombre común, nombre propio, …).
- Información morfosintáctica. El euskera presenta una gran variedad de informaciones de este tipo, incluyendo el caso y número para los elementos integrantes del sintagma nominal, o información de concordancia con sujeto, objeto directo e indirecto en verbos, así como distintos tipos de oraciones subordinadas. Entre los idiomas presentados a CoNLL (2007) es el

[3] *CoNLL (Computational Natural Language Learning) shared task on dependency parsing*.

idioma que presenta, de lejos, un mayor número de rasgos morfosintácticos (359).

- Etiquetas de dependencia. Se ha definido un conjunto de 35 etiquetas.

El analizador usado se basa en la técnica de reducción y desplazamiento utilizando, por tanto, una pila donde va añadiendo elementos de la cadena de entrada. Por ello, se pueden especificar elementos tanto de la pila como de la cadena de entrada para su uso en la fase de aprendizaje automático. Además, como el analizador va construyendo el árbol de dependencias, también se pueden especificar rasgos del antecesor o los descendientes de un elemento de la pila o del primer elemento que queda sin analizar de la cadena de entrada[4].

| | **Especificación** | **Descripción** |
|---|---|---|
| 1 | $p(\sigma_0)$ | Categoría del símbolo del tope de la pila |
| 2 | $d(h(\sigma_0))$ | Etiqueta de dependencia del símbolo del tope de la pila con su núcleo |
| 3 | $p(\tau_0)$ | Categoría de la primera palabra de la cadena de entrada por analizar |
| 4 | $f(\tau_1)$ | Rasgos morfosintácticos de la palabra siguiente a la primera de la cadena de entrada |
| 5 | $w(l(\sigma_1))$ | Forma de la palabra correspondiente al descendiente más a la izquierda del elemento debajo del tope de la pila. |

Tabla 1: Ejemplos de especificación de parámetros para el sistema de aprendizaje.

La tabla 1 muestra un ejemplo de especificación de los parámetros de aprendizaje del sistema. Se permite especificar elementos de la pila ($\sigma$) o de la cadena de entrada ($\tau$), mediante su posición relativa (empezando desde el cero). Por ejemplo, la especificación 1 hace referencia a la categoría *p(art of speech)* del símbolo en el tope de la pila. Las etiquetas *w(ord),* L(ema), d(ependencia), *h(ead), l(eft)* y *r(ight)* se refieren a la forma, dependencia, al núcleo, descendiente izquierdo y descendiente derecho, respectivamente. Estas etiquetas se pueden combinar para formar especificaciones más complejas, como en los ejemplos 1-5 de la tabla 1. Por ejemplo, la especificación número 5 de la tabla hace referencia a la forma del dependiente más a la izquierda del símbolo que se encuentra debajo del tope de la pila.

Los datos del treebank se han separado en una parte para entrenamiento (50.123 palabras) y otra para la prueba final (*gold test*, 5.318 palabras[5]). Los experimentos se han analizado aplicando la técnica de 10 *fold cross-validation* sobre los datos de entrenamiento y finalmente sobre los datos del *gold-test*.

| Características | $\Phi 1$ | $\Phi 2$ |
|---|---|---|
| $p(\sigma 1)$ | + | + |
| $p(\sigma 0)$ | + | + |
| $p(\tau 0)$ | + | + |
| $p(\tau 1)$ | + | + |
| $p(\tau 2)$ | + | |
| $p(\tau 3)$ | + | |
| $p(l(\sigma 0))$ | | + |
| $p(l(\tau 0))$ | | + |
| $w(h(\sigma 0))$ | + | |
| $w(\sigma 0)$ | + | + |
| $w(\tau 0)$ | + | + |
| $w(\tau 1)$ | + | |
| $L(\sigma 0)$ | | + |
| $L(\tau 0)$ | | + |
| $L(\tau 1)$ | | + |
| $d(l(\sigma 0))$ | + | + |
| $d(\sigma 0)$ | + | + |
| $d(r(\sigma 0))$ | + | |
| $d(l(\tau 0))$ | + | |
| $f(\tau 0)$ | | + |
| $f(\sigma 0)$ | | + |
| $f(h(\sigma 0))$ | | + |

Tabla 2. Modelos de características.

En las pruebas efectuadas se ha querido valorar la importancia del uso de la información morfosintáctica para el entrenamiento, probando si el uso de dicha información mejora significativamente los resultados obtenidos por el parser. A la hora de seleccionar los atributos utilizados por el parser se han especificado los parámetros de la tabla 2 siguiendo las especificaciones de la tabla 1. Se han realizado múltiples pruebas con diferentes clases de parámetros.

La tabla 2 muestra dos clases de pruebas que se han realizado. La columna $\Phi 1$ presenta la

[4] Al ser el análisis de izquierda a derecha, solo el primer símbolo de la entrada puede tener antecesor o descendientes.

[5] Debido a errores en la conversión del treebank original, el número de palabras original se ha visto reducido respecto al total de palabras del corpus.

combinación de características estándar usada por Nivre et al. (2007) para una gran variedad de lenguas. La columna Φ2 muestra la combinación más exitosa obtenida en el total de los experimentos, donde se han añadido rasgos correspondientes a información morfosintáctica.

La tabla 3 muestra cómo el uso de información morfosintáctica presenta una mejora de 8 puntos en *Labeled Attachment Score*[6] *(LAS)* de Φ1 sobre Φ2.

| | Φ1 | Φ2 |
|---|---|---|
| 10 fold cross-validation average | 67,64 | 75,06 |
| Gold-Test | 65,08 | 74,41 |

Tabla 3. Resultados obtenidos (*LAS*).

Los experimentos anteriores se han realizado utilizando el corpus en su estado original y cambiando las especificaciones de los parámetros. Teniendo en cuenta que el número de rasgos morfológicos distintos para el euskera es el mayor de todos los idiomas presentados a CoNLL (359) hemos pensado en reducir su número teniendo en cuenta conocimiento específico del euskera, eliminando algunos rasgos que se han considerado poco significativos y unificando rasgos que se considera que tienen un comportamiento común de cara al análisis (por ejemplo, un subconjunto importante de las marcas de caso indican el mismo tipo de dependencia *ncmod*, modificador no clausal, por lo que decidimos agruparlas). Con esto se espera facilitar la tarea de aprendizaje y reducir el tiempo de aprendizaje y análisis. El resultado no muestra una mejoría (ver tabla 4), al no superar un *LAS* de 74,41% obtenido con un mayor conjunto de rasgos, aunque sí lo hace en cuanto al tiempo de entrenamiento y de análisis, siendo 3 y 8 veces más rápido, respectivamente.

Aunque no se ha mostrado en las tablas, se ha comprobado, en concordancia con los resultados de Nivre et al. (2007), que el uso de SVM mejora los resultados de MBL cerca de un 3%. Por ello, los resultados presentados corresponden al uso de SVM.

[6] Porcentaje de palabras en las que el sistema predice correctamente tanto su núcleo como la relación de dependencia existente entre ellos.

| Nº de rasgos | 10 fold cross-validation average (Φ2) | Gold-test |
|---|---|---|
| 359 | 75,06 | **74,41** |
| 163 | **75,13** | 73,45 |

Tabla 4. Resultados (*LAS*) obtenidos al reducir el número de rasgos morfosintácticos.

## *5 Comparación con otros trabajos*

Este trabajo se enmarca en el ámbito del análisis sintáctico estadístico basado en dependencias, cuyo máximo exponente actualmente son las competiciones CoNLL 2006 y 2007. En cuanto a los resultados generales, el indicador de asignación de etiqueta correcta (*Labeled Attachment Score, LAS)* conseguido (74,41%) sitúa a nuestro sistema cerca de los mejores resultados presentados (76,94%). De hecho, este resultado iguala a los obtenidos con un único sistema, ya que el mejor resultado de CoNLL se da al combinar varios analizadores.

En otro trabajo, Cowan y Collins (2005) presentan los resultados de aplicar el analizador de Collins al castellano, que presenta como novedad una mayor flexión que el inglés. El trabajo experimenta con el uso de diferentes tipos de información morfológica, concluyendo que esta información ayuda a mejorar los resultados del analizador.

Eryiğit, Nivre, y Oflazer (2006) experimentan con el uso de distintos tipos de información morfológica para el análisis del turco, comprobando cómo el aumento de la riqueza de la información inicial aumenta la precisión. En un trabajo relacionado, Eryiğit y Oflazer (2006) comprueban que el uso de los morfemas como unidad de análisis (en vez de palabras) también mejora el analizador.

Aranzabe, Arriola, y Díaz de Ilarraza (2004) están desarrollando un analizador sintáctico basado en dependencias para el euskera. Este analizador está basado en conocimiento lingüístico, donde la gramática se ha escrito en el formalismo *Constraint Grammar* (Tapanainen, 1996). No se tienen en este momento resultados publicados sobre la precisión y cobertura de este analizador, por lo que no es posible establecer comparaciones directas con el sistema aquí presentado.

## *6 Conclusiones*

Este artículo ha presentado la preparación del treebank 3LB para el euskera para su tratamiento computacional, así como la adaptación del analizador de Nivre et al. (2007) al tratamiento del euskera. Este lenguaje presenta como características principales el orden libre de constituyentes de la oración y el uso de información morfosintáctica rica en comparación con otras lenguas.

El trabajo presentado supone la primera aproximación al análisis sintáctico estadístico del euskera, en paralelo con la competición CoNLL 2007, en la que hemos colaborado en la fase de preparación de datos.

Se han probado diferentes tipos de parámetros y algoritmos, obteniendo una precisión superior al 74%, que se acerca a los resultados obtenidos por los mejores sistemas de (CoNLL 2007) para la misma tarea. Se ha probado que incorporar distintos tipos de información morfosintáctica mejora notablemente los resultados. Entre las acciones para continuar esta investigación planteamos:

- Análisis no proyectivo. Los algoritmos empleados en este trabajo requieren que las dependencias sean proyectivas, es decir, no puede haber arcos que se crucen. El análisis de los datos del euskera muestra que un 2,9% de las dependencias en el treebank son no proyectivas. Para estos casos, Nivre y Nilsson (2005) plantean un algoritmo que convierte arcos no proyectivos en proyectivos. Al ser el algoritmo reversible, permite volver el treebank a la configuración inicial después del análisis sintáctico, para realizar la evaluación final. Esta conversión permite usar algoritmos de análisis que en principio solo son válidos para la construcción de árboles proyectivos.
- Hemos comprobado cómo una de las categorías sintácticas que peores resultados presenta es el nombre (*LAS* de 66%). Al ser el nombre una de las categorías más frecuentes, presenta un gran porcentaje del total de errores realizados (cerca del 50% de todos los errores). Una de las hipótesis que planteamos es que puede deberse al hecho de que el nombre es comúnmente enlazado con el verbo, pero la dependencia se hace en función del caso gramatical, que muchas veces pertenece a otra palabra[7]. Por ello estamos planteando la posibilidad de separar el caso gramatical como un elemento distinto, es decir, tomar morfemas como unidad de análisis. Esta idea aplicada a la alineación de textos en traducción automática ha producido mejoras significativas (Agirre et al., 2006).
- Estudio del efecto que tiene el tipo de corpus en los resultados. El corpus utilizado dispone de dos clases de textos: literarios y periodísticos. Aunque el tamaño reducido del corpus usado no ha permitido realizar pruebas por separado para cada uno de ellos, hemos comprobado que los resultados mejoran (cerca de un 5%) cuando el corpus de entrenamiento está formado solo por textos de un tipo. La ampliación del treebank, que pasará en breve a tener cerca de 300.000 palabras, permitirá realizar estas pruebas con más precisión. Esto también posibilitará el estudio de la aportación del tamaño del corpus.
- Estudio del efecto de la fase de desambiguación morfosintáctica. En este momento, el analizador ha sido probado con una sola interpretación por palabra, es decir, la entrada del analizador es *perfecta*. La fase de desambiguación previa introducirá errores que se acumulan a los del analizador sintáctico. Aunque los errores de la fase de etiquetado morfológico no son tan importantes para otras lenguas, la alta ambigüedad del euskera (2,81 interpretaciones por palabra, Ezeiza et al. 1998) supone un reto añadido.

**Agradecimientos**

Este trabajo está subvencionado por el Departamento de Industria y Cultura del Gobierno Vasco (proyecto AnHITZ 2006, IE06-185).

**Bibliografía**

Agirre E., A. Díaz de Ilarraza, G. Labaka, y K. Sarasola. 2006. Uso de información

[7] Por ejemplo, en el sintagma nominal "*etxe handi horrekin*" (con esa casa grande), la palabra *etxe* debe asociarse con el verbo, pero el tipo de dependencia viene dado por el sufijo *–ekin*, que aparece dos palabras más adelante.

morfológica en el alineamiento Español-Euskara. *XXII Congreso de la SEPLN*.

Aranzabe M., J.M. Arriola, y A. Díaz de Ilarraza. 2004. Towards a Dependency Parser of Basque. Proceedings of the *Coling 2004 Workshop on Recent Advances in Dependency Grammar*. Geneva.

Briscoe, E., J. Carroll, y R. Watson. 2006. The Second Release of the RASP System. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, Sydney.

Carroll, J., G. Minnen, y E. Briscoe. 1999. Corpus annotation for parser evaluation. In *Proceedings of the EACL-99 Post-Conference Workshop on Linguistically Interpreted Corpora,* Bergen. 35-41.

Chang, C.-C. y Lin, C.-J. 2001. *LIBSVM: A library for support vector machines*. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Collins M. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. PhD Dissertation, University of Pennsylvania.

Collins M., J. Hajic, E. Brill, L. Ramshaw, y Tillmann C. 1999. A Statistical Parser for Czech. In: *Proceedings of the 37th Meeting of the ACL*, pp. 505-512. University of Maryland, College Park, Maryland.

CoNLL 2006 y 2007. *Proceedings of the Tenth/Eleventh Conference on Computational Natural Language Learning*.

Cowan B. y M. Collins. 2005. Morphology and Reranking for the Statistical Parsing of Spanish. Proceedings of the *Conference on Empirical Methods in NLP* (EMNLP).

Daelemans, W. y A. Van den Bosch. 2005. *Memory-Based Language Processing*. Cambridge University Press.

Eryiğit G., J. Nivre, y K. Oflazer. 2006. The incremental use of morphological information and lexicalization in data-driven dependency parsing. In *Proceedings of the 21st International Conference on the Computer Processing of Oriental Languages (ICCPOL),* Springer LNAI 4285.

Eryiğit G., y K. Oflazer. 2006. Statistical Dependency Parsing for Turkish. Proceedings of EACL 2006 - *The 11th Conference of the European Chapter of the Association for Computational Linguistics*, April 2006, Trento, Italy

Ezeiza N., I. Aduriz, I. Alegria, J.M. Arriola, y R. Urizar. 1998. Combining Stochastic and Rule-Based Methods for Disambiguation in Agglutinative Languages, *COLING-ACL'98*, Montreal (Canada). August 10-14, 1998.

Eisner J. 1996. Three new probabilistic models for dependency parsing: an exploration. Proceedings of *COLING-1996,* Copenhagen.

Hajič J. Building a Syntactically Annotated Corpus: The Prague Dependency Treebank. 1998. In: E. Hajičová (ed.): *Issues of Valency and Meaning. Studies in Honour of Jarmila Panevová*, Karolinum, Charles University Press, Prague, pp. 106-132.

Jarvinen T., y P. Tapanainen. 1998. Towards an implementable dependency grammar. CoLing-ACL'98 workshop *'Processing of Dependency-Based Grammars'*, Kahane and Polguere (eds), p. 1-10, Montreal, Canada.

Tapanainen P. 1996. *The Constraint Grammar Parser CG-2*. Number 27 in Publications of the Department of General Linguistics, University of Helsinki.

Lin D. 1998. Dependency-based Evaluation of MINIPAR. In *Workshop on the Evaluation of Parsing Systems*, Granada, Spain, May, 1998.

Marcus M., B. Santorini y M. Marcinkiewiecz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19 (2), 313--330.

Nivre, J. y J. Nilsson. 2005. Pseudo-Projective Dependency Parsing. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics* (ACL), 99-106.

Nivre, J., J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov, y E. Marsi. 2007. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2).

Palomar M., M. Civit , A. Díaz de Ilarraza , L. Moreno, E. Bisbal, M. Aranzabe, A. Ageno, M.A. Martí, y B. Navarro. 2004. 3LB: Construcción de una base de árboles sintáctico-semánticos para el catalán, euskera y castellano. *XX Congreso de la SEPLN.*