Universitat Politècnica de València
Departament de Sistemes Informàtics i Computació

Etiquetado Léxico y Análisis Sintáctico Superficial basado en Modelos Estadísticos

Ferran Pla i Santamaría

Memoria para optar al grado de Doctor en Informática

bajo la dirección de

Natividad Prieto Sáez

Lluís Padró i Cirera

València, septiembre de 2000

Agraïments

Una tesi, malgrat ser una tasca individual, també és un treball d'equip. És per això, que sense la col·laboració de molts companys i companyes, que més que això han sigut amics i amigues, probablement, aquest treball no hauria vist mai la llum.

Tot va començar fa ara poc més de quatre anys quan preparavem un projecte de recerca. Aleshores, ens vam ajuntar un grup de gent que volíem treballar en un camp nou per a nosaltres. Si bé ja teniem experiència en temes relacionats, no sabiem massa bé com començar, i ni de bon tros, disposavem dels recursos per emprendre el nou repte. Parle de, i els anomeraré alfabèticament pel nom, Antonio Molina, Emilio Sanchis, Encarna Segarra, Lidia Moreno i Natividad Prieto. Justament aquest ordre, no sé si per casualitat, ha fet que la darrera haja estat la directora de tesi, i el primer, el que més ha ajudat que les idees s'hagen plasmat en codi executable, i que les llarges estones davant l'ordinador s'hagen fet més curtes. Es clar, les idees no haurien eixit sense les moltes converses mantigudes amb Encarna, Lidia i Emilio; ah!, m'oblidava dels d'Alacant, que han estat i estan en projectes comuns, i que com en són tants, millor no escriure els seus noms. I, com no, també vull agrair Enrique Vidal els seus suggeriments i les estones dedicades a escoltar els meus projectes.

D'altra banda, seria injust no fer referència a l'ajuda rebuda dels companys de la Universitat Politècnica de Catalunya, particularment d'Horacio Rodríguez i de Lluís Padró, aquest últim també director de tesi, que de manera totalment desinteressada, sempre m'han donat el seu suport. Així, gràcies a ells, vam disposar dels primers corpus, i d'algunes eines, sense les quals, segurament, no estaria escrivint aquest full.

A tots els esmentats, i als que de segur m'he deixat, sense oblidar-me dels que més a prop he tingut: Maite, Albert i la xicoteta Maria que acaba d'arribar, a tots, us done el més sincer agraïment.

Índice General

Índice de Figuras

Índice de Tablas

Capítulo 1

Introducción

El objetivo general de todo sistema de Procesamiento del Lenguaje Natural (PLN) es el de obtener alguna representación del mensaje contenido en las frases. El tratamiento automático de una lengua es un problema de gran complejidad en el que intervienen diversas y complejas fuentes de conocimiento: fonética, morfología, sintaxis, semántica, pragmática, conocimiento del mundo, etc. Aunque en algunos casos estas fuentes de información se pueden considerar independientes, en general, presentan una interrelación, sin la cual, no se puede conseguir una correcta interpretación del significado y de la función de las palabras de una oración.

Debido a esta complejidad, para abordar el problema de comprensión de una lengua se suele seguir una de las siguientes vías: 1) Se resuelven ciertos subproblemas más sencillos que, en algunos casos, deben adoptar simplificaciones para poder ser tratados de manera automática, tales como: análisis morfológico, etiquetado léxico de textos, análisis sintáctico superficial o parcial de oraciones, ligamiento preposicional, desambiguación del sentido de las palabras, tratamiento de fenómenos lingüísticos específicos como la anáfora, elipsis, etc. 2) Se simplifica el lenguaje considerando tareas restringindas, en la talla del vocabulario, la complejidad de las estructuras sintácticas utilizadas o el dominio semántico de la aplicación.

Durante los últimos años podemos encontrar una gran cantidad de ejemplos que toman alguna de las vías comentadas. En reconocimiento del habla hay aplicaciones que se restringen a vocabularios acotados, consultas a bases de datos específicas, sistemas de diálogo sobre tareas concretas, etc. En otros campos, más directamente

relacionados con el PLN, encontramos aplicaciones de traducción automática, extracción y recuperación de información, resúmenes de textos, etc, en las que, en mayor o menor medida, se restringen a dominios específicos para conseguir resultados aceptables.

Por otra parte, el hecho de disponer de grandes corpus de datos, textuales u orales, anotados con información lingüística de diferente naturaleza –información morfosintáctica, análisis sintáctico total o parcial, información semántica– junto con el creciente desarrollo de los computadores, lenguajes de programación y sistemas operativos, ha propiciado la aparición y uso de *aproximaciones inductivas* o *métodos basados en corpus*, dentro del campo de la Lingüística Computacional, que aplicados a diferentes tareas de PLN obtienen un alto grado de prestaciones.

Las aproximaciones inductivas, con o sin información estadística, resultan de gran interés para conseguir la desambiguación del Lenguaje Natural (LN) ya que, además de proporcionar resultados aceptables, utilizan modelos relativamente sencillos y sus parámetros se pueden estimar a partir de datos. Esto las hace especialmente atractivas, puesto que en el cambio de una tarea a otra, o incluso de lengua, se reduce substancialmente la intervención humana. No obstante, algunos casos de ambigüedad no pueden ser resueltos de esta forma y se debe recurrir a un experto humano para introducir, por ejemplo, ciertas reglas o restricciones que ayuden a su resolución.

1.1 Visión General

En este trabajo se abordan dos problemas que simplifican substancialmente la tarea de procesamiento de oraciones escritas en LN: la desambiguación léxica y el análisis sintáctico superficial de textos no restringidos. Se ha desarrollado un sistema de desambiguación que es capaz de obtener de manera conjunta o separada, el etiquetado léxico (*POS tagging*) –o proceso mediante el cual se elige la categoría léxica correcta para las palabras de una frase– y el análisis sintáctico superficial (*Shallow Parsing o Chunking*) –consistente en la detección de ciertos grupos no solapados de palabras relacionadas sintácticamente como, sintagmas nominales (SN), verbales (SV), preposicionales (SP), ...–, para textos no restringidos.

La aproximación propuesta se basa en modelos de lenguaje (ML) obtenidos a partir de corpora etiquetados con información lingüística. Para ello, se utilizan técnicas de aprendizaje automático derivadas del campo de la inferencia gramatical y de los modelos estadísticos. Los modelos inferidos se representan utilizando un formalismo homogéneo: máquinas de estados finitos. Éstos incluyen desde modelos de n-gramas, hasta cualquier modelo regular estocástico aprendido por medio de técnicas de inferencia gramatical u obtenido mediante cualquier otro método. En particular se ha utilizado el algoritmo *Error Correcting Grammatical Inference (ECGI)* para inferir ML gramaticales, los cuales han sido extendidos, mediante técnicas de suavizado, para garantizar una cobertura total del lenguaje.

El sistema inicialmente construido utiliza ML de categorías léxicas que se utilizan para resolver el problema del etiquetado léxico de textos. La misma metodología se ha generalizado para modelizar ciertas unidades lingüísticas (SN, SV, SP, ...), con el fin de poder construir ML estructurados en dos niveles que sean capaces de resolver el problema del análisis sintáctico superficial. El nivel superior refleja las estructuras de las oraciones en términos de categorías léxicas y descriptores de unidades sintácticas, mientras que en el nivel inferior se representa el modelo de cada una de las unidades sintácticas consideradas en el nivel superior. Estos modelos se pueden utilizar para resolver, de manera integrada, dos problemas que tradicionalmente se han abordado de manera separada: el etiquetado léxico y el análisis sintáctico superficial de textos.

Finalmente, el sistema ha sido ampliado para enriquecer los ML con información de las palabras, además de las categorías léxicas, mediante lo que hemos denominado modelos contextuales especializados. Éstos permiten reflejar dependencias léxico-contextuales, que en muchos casos, ayudan de manera notable a resolver ciertas ambigüedades estructurales.

El sistema propuesto se ha evaluado experimentalmente sobre diferentes corpora en inglés y en castellano, estableciendo comparaciones con aproximaciones desarrolladas por otros investigadores. El trabajo experimental se ha centrado en los siguientes aspectos:

- Evaluación de los métodos de suavizado definidos sobre los modelos de lenguaje ECGI.
- Evaluación de los diferentes ML utilizados para el problema de etiquetado

léxico, tanto los modelos simples como los integrados.

- Evaluación de los ML integrados en diferentes tareas de análisis superficial, definidas sobre varios corpora y considerando diversas definiciones de unidades sintácticas.

1.2 Aportaciones

A continuación se resumen brevemente los aspectos más novedosos del sistema de desambiguación léxica y sintáctica desarrollado en este trabajo, tanto desde el punto de vista teórico, como principalmente, desde el punto de vista práctico.

En esta tesis se ha definido un marco homogéneo –autómatas de estados finitos estocásticos– para aprender y describir ML que se puedan incorporar en sistemas de etiquetado léxico y análisis sintáctico superficial de textos no restringidos.

Desde un punto de vista tecnológico, es necesario destacar que se ha desarrollado un sistema que integra de manera modular tanto las fases de aprendizaje como la de su aplicación a tareas de desambiguación. Se ha diseñado de tal manera que su modificación y ampliación resulten tareas sencillas. Además, se ha hecho especial énfasis en estandarizar las entradas y salidas del sistema para facilitar su incorporación a otros sistemas similares, así como simplificar la comparación de las prestaciones de nuestro sistema con otros que abordan la misma problemática.

Los objetivos conseguidos los clasificaremos en los siguientes puntos:

- **Modelización del lenguaje.** Los ML considerados en este trabajo abarcan desde modelos estadísticos (n-gramas) hasta modelos sintácticos obtenidos mediante técnicas de inferencia gramatical, en concreto, utilizando el método ECGI. Estos modelos, a parte de poder utilizarse individualmente (en el etiquetado léxico), se pueden combinar para construir modelos estructurados en dos niveles para abordar tareas de detección de unidades sintácticas (SN, SV, SP, etc.).

- **Suavizado de modelos regulares**. Se han propuesto métodos de suavizado aplicables a cualquier autómata de estados finitos, inspirados en los que

se utilizan para modelos de n-gramas. Estos métodos se han utilizado para extender los modelos obtenidos mediante el algoritmo ECGI (Pla, 1999) y así poder garantizar una cobertura total del lenguaje estudiado.

- **Utilización de los ML para el etiquetado léxico.** Los ML propuestos se han utilizado con éxito en diferentes tareas y para diferentes lenguas. En (Pla and Prieto, 1998), se propuso el primer prototipo de etiquetado léxico y se aplicó para el castellano. Las carencias del sistema inicial (falta de robustez, cobertura, eficiencia algorítmica) se han corregido. Además, se ha enriquecido el sistema con la incorporación de un analizador morfológico para el castellano y de los métodos de suavizado propuestos en (Pla, 1999). Con el nuevo sistema se han abordado nuevas tareas. Se ha incorporado el etiquetador léxico como entrada a un sistema de Análisis Parcial (Molina et al., 1999a; Molina et al., 1999b). Se ha realizado el etiquetado léxico y supervisión del corpus en castellano *BDGEO* (Pla and Molina, 1999) compuesto por frases de consulta a una base de datos geográfica.

- **Integración de las tareas de etiquetado léxico y análisis sintáctico superficial.** Mediante el uso de ML estructurados se ha ampliado el sistema para abordar tareas de detección de ciertos constituyentes sintácticos de las oraciones (SN, SV, SP, ..., etc). La validación se ha realizado sobre el corpus en inglés *Wall Street Journal (WSJ)*. En (Pla et al., 2000a) se presentan resultados de etiquetado léxico y de detección de SN usando ML estructurados de bigramas. En (Pla et al., 2000b), se extienden los ML combinando bigramas y modelos ECGI y se hace un estudio comparativo de estos modelos sobre la tarea de detección de SN. En (Pla et al., 2000c) se amplía a otras unidades (SV, SP, SADJ, ...) también para el inglés. No sabemos que existan corpora disponibles en castellano anotados con información de unidades sintácticas. Debido a esta dificultad, el análisis superficial para el castellano se ha hecho sobre el corpus *LexEsp* y sólo considerando SN. Estás unidades se han obtenido a partir de la salida del analizador parcial APOLN (Molina et al., 1999a).

- **Incorporación de las palabras en los ML.** Se ha propuesto una extensión de los ML para permitir la incorporación de ciertas palabras, junto con las categorías léxicas, en los modelos contextuales. Para ello, se hace intervenir en el ML un conjunto de palabras que se eligen del corpus de aprendizaje,

siguiendo un criterio preestablecido: las más frecuentes, las pertenecientes a categorías cerradas, las de más difícil desambiguación, ... Con esta aportación, se obtienen mejoras en el etiquetado léxico y la detección de unidades sintácticas (Pla et al., 2000c).

- **Desarrollo de la aplicación.** Se han desarrollado una serie de recursos que se pueden combinar e integrar fácilmente en tareas de PLN. Estos recursos abarcan desde los módulos de entrenamiento, hasta los de aplicación y evaluación desarrollados, o los incorporados de otros sistemas. Se han utilizado formatos estándar de entrada/salida para facilitar las comparaciones con sistemas similares que abordan los mismos problemas. Se ha desarrollado un entorno gráfico (Ribera et al., 2000) que permite una utilización muy sencilla, especialmente adecuada para usuarios no expertos. Esta aplicación permite realizar de una forma amigable las tareas de supervisión por lingüistas de las diferentes salidas del sistema: etiquetado léxico y análisis sintáctico en formato de árbol. Además, permite completar por parte del usuario el análisis superficial obtenido de manera automática.

1.3 Esquema de la Tesis

El trabajo presentado se estructura en los siguientes capítulos. En el presente capítulo se marcan los objetivos y se enumeran las principales aportaciones de esta tesis.

En el **Capítulo 2** se hace una revisión bibliográfica de las principales aproximaciones al etiquetado léxico de textos y al análisis sintáctico superficial.

En el **Capítulo 3** se presenta con detalle la aproximación al etiquetado léxico de textos basada en modelos de Markov o n-gramas que son la base de todos los desarrollos siguientes, tanto a nivel teórico, como experimental. Se destacan las aportaciones más relevantes, incidiendo en el ML empleado, las técnicas de suavizado utilizadas y los algoritmos de etiquetado comúnmente usados. Se introducen los modelos contextuales especializados con el fin de poder representar restricciones contextuales de ciertas palabras, además de las categorías léxicas.

En el **Capítulo 4** se presenta el método de inferencia gramatical ECGI y se

propone una extensión de los modelos obtenidos (*modelos ECGIE*) introduciendo técnicas de suavizado inspiradas en las que se usan para los modelos de *n-gramas*. Además, se hace una evaluación experimental de los distintos suavizados en una tarea de etiquetado léxico.

En el **Capítulo 5** se describe el sistema de etiquetado léxico propuesto y se hace un estudio comparativo de las prestaciones de etiquetado con los diferentes ML introducidos. La experimentación se realiza sobre diversos corpus, en castellano y en inglés, de uso extendido entre la comunidad científica. Se presentan resultados preliminares del comportamiento del sistema para la tarea de etiquetado léxico y supervisión de nuevos corpora.

En el **Capítulo 6** se amplía el sistema propuesto para abordar la tarea del análisis sintáctico superficial. Se describe el proceso de construcción de los modelos estructurados y se realiza un estudio experimental para validar la aproximación propuesta sobre diversas tareas en inglés y en castellano.

En el **Capítulo 7** se describe un entorno gráfico para facilitar el uso de las herramientas desarrolladas y para simplificar la supervisión de corpora por expertos humanos.

En el **Capítulo 8** de conclusiones se hace una recopilación de las técnicas utilizadas y de los principales objetivos conseguidos. A partir de éstos se establecen una serie de conclusiones y trabajos futuros a realizar.

En el apartado de **Apéndices** se describe el conjunto de categorías léxicas utilizado en los corpora empleados en la validación de la aproximación propuesta. En el Apéndice A se detallan las etiquetas léxicas y sintácticas utilizadas. En el Apéndice B se presentan algunos ejemplos de frases etiquetadas automáticamente pertenecientes al corpus BDGEO. Finalmente, en el Apéndice C se lista el conjunto de palabras que se han tenido en cuenta en los modelos de lenguaje especializados.

Capítulo 2

Desambiguación Léxica y Sintáctica de Textos

En este capítulo se hace una revisión de las aportaciones más relevantes a la resolución de la ambigüedad léxica y al análisis sintáctico superficial de textos. Se presentan los problemas y se clasifican las soluciones aportadas en dos grandes familias: *Aproximaciones Lingüísticas* y *Aproximaciones de Aprendizaje Automático basado en corpus*, haciendo especial énfasis en las segundas.

2.1 Etiquetado Léxico de Textos

El etiquetado léxico de textos es un problema de desambiguación bien conocido dentro del campo del PLN. Un etiquetador toma como entrada frases escritas en una cierta lengua y proporciona como salida la etiqueta o categoría léxica más adecuada para cada palabra, atendiendo al contexto en que aparece. Generalmente, estas categorías se toman de un conjunto previamente definido por expertos y dan cuenta de diferentes aspectos léxicos de la lengua objeto de estudio. Cuando a una palabra se le puede asignar más de una categoría, el proceso de desambiguación se lleva a cabo considerando la información del contexto en el cual aparece.

El proceso de etiquetado léxico (figura 2.1), básicamente, tiene en cuenta dos fuentes de conocimiento: el *Modelo de Lenguaje* (ML) o modelo contextual, que

describe las posibles o probables secuencias de categorías léxicas, y el *Modelo Léxico* que representa la relación entre el vocabulario de la aplicación y el conjunto de categorías.

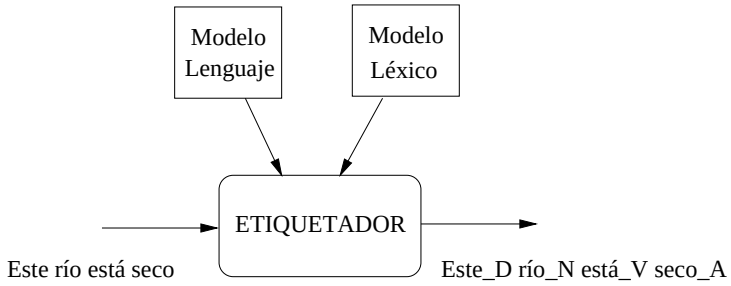

Figura 2.1: Proceso de etiquetado léxico.

Por ejemplo, en la frase *"Este río está seco"*, se observa que hay varias palabras que admiten diferentes categorías léxicas: *"Este"* puede ser Determinante (D), Pronombre (P) o Nombre (N), *"río"* puede funcionar como Verbo (V) o como Nombre (N), *"está"* sólo como Verbo (V) y por último *"seco"* como Verbo (V) o Adjetivo (A). El objetivo de un etiquetador es determinar, de entre todas las posibilidades expuestas, la que en el contexto de la frase sea la correcta para cada palabra: *"Este_ D río_ N está_ V seco_ A"*.

Es indudable el gran interés que tiene el desarrollo de etiquetadores de textos y sus múltiples y diversas aplicaciones. Se pueden utilizar como preprocesadores de oraciones en sistemas de PLN, para permitir que los complejos analizadores sintácticos y semánticos utilicen como preterminales categorías, en lugar de palabras, y hacerlos de esta manera más sencillos. En sistemas de síntesis de voz, ya que la entonación de una determinada palabra puede ser una u otra según la categoría de la misma. En sistemas de extracción de información, debido a que en una frase el contenido semántico de unas categorías es mayor que el de otras. En aplicaciones de recuperación de información, resúmenes de textos, y en general, para la obtención de grandes cantidades de texto etiquetado con información lingüística, para ser utilizados como datos de aprendizaje en aproximaciones inductivas de PLN. En reconocimiento automático del habla para la construcción de ML basados en categorías, en vez de palabras, con el fin de reducir el número de parámetros a estimar en el modelo. En todos estos casos, es interesante disponer de un etiquetador au-

tomático o semiautomático que nos realice de manera rápida y fiable lo que a un experto lingüista le ocuparía meses de rutinario trabajo. El uso de herramientas automáticas, no excluye en ningún caso, la cooperación con expertos humanos para la supervisión de las diferentes salidas obtenidas de manera automática y para el estudio de ciertos problemas de difícil desambiguación.

2.2 Aproximaciones al Etiquetado Léxico de Textos

Atendiendo a las técnicas utilizadas para establecer el ML, las soluciones propuestas en la literatura al problema de etiquetado léxico se pueden clasificar en dos grandes grupos: *Aproximaciones Lingüísticas* basadas en un conjunto de reglas establecidas manualmente por expertos o aprendidas de manera (semi)automática y las *Aproximaciones de Aprendizaje Automático* (basadas en corpus) que usan textos, generalmente anotados con información lingüística para establecer los modelos subyacentes. También se pueden encontrar *Aproximaciones Híbridas* que combinan ciertos aspectos de las anteriores.

Todas estas aproximaciones tienen en común que para una palabra dada, sólo unas ciertas categorías son posibles: las que aparecen en el diccionario (si se dispone), o las que se derivan de su análisis morfológico si es posible. Además, cuando una palabra puede pertenecer a varias categorías léxicas, generalmente, la correcta se determina teniendo en cuenta el contexto en el que se encuentra, contexto que se describe a través de diferentes mecanismos: reglas contextuales definidas por un experto o aprendidas, modelos de n-gramas, árboles de decisión, autómatas y traductores de estados finitos, etc.

2.2.1 Aproximaciones Lingüísticas

Bajo esta aproximación encontramos las primeras tentativas de resolución de la ambigüedad léxica ya en los años 60 y 70. Los primeros etiquetadores, (Greene and Rubin, 1971),(Klein and Simmons, 1963), estaban compuestos por un conjunto de reglas, escritas manualmente por lingüistas, con el objeto de predecir o restringir las posibles categorías de una palabra. El sistema *TAGGIT* fue utilizado como un primer paso en la construcción de grandes corpus como el *Brown* (Francis and Kučera,

1982). Este corpus fue utilizado, y se sigue utilizando, como base de conocimiento en las aproximaciones probabilísticas y/o como conjunto de prueba para otras aproximaciones.

Estos sistemas eran totalmente dependientes de la lengua para la que se habían diseñado y requerían un alto coste humano para la definición de la reglas. Además, el léxico que se usaba era restringido, por lo que aparecían muchas situaciones no contempladas. Debido a esto, la cobertura (conjunto de casos de ambigüedad considerados en las reglas) era baja y se dificultaba la transportabilidad de unas tareas a otras, y mucho más, entre diferentes lenguas.

El formalismo basado en gramáticas de restricción –*Constraint Grammar*– (Karlsson, 1990; Voutilainen, 1993) es el que con mayor éxito se ha aplicado en este campo. Más recientemente, el sistema *EngCG* (Voutilainen, 1993; Voutilainen and Järvinen, 1995) para el inglés, en el campo del análisis sintáctico superficial, constituye la solución más relevante al problema desde esta perspectiva. Para otras lenguas como el turco (Oflazer and Kuruöz, 1994) o el vasco (Aduriz et al., 1995) también se ha aplicado este tipo de formalismo.

A pesar de que el esfuerzo humano requerido por estas aproximaciones es muy elevado, la principal ventaja es que se construyen ML desde un punto de vista lingüístico, por lo que se pueden incluir muchas y complejas fuentes de información, difíciles de capturar de manera automática. Este hecho las hace más expresivas, por lo que en general, suelen proporcionar mejores prestaciones en tareas de desambiguación si se comparan con otro tipo de aproximaciones.

2.2.2 Aproximaciones de Aprendizaje Automático

En este apartado se consideran aquellas propuestas que construyen un ML utilizando métodos de aprendizaje a partir de datos. Estas aproximaciones difieren entre sí en el método de aprendizaje y en la complejidad del modelo construido. Muchos son los formalismos utilizados: Modelos de Markov o n-gramas, reglas de transformación, árboles de decisión, redes neuronales, autómatas y traductores de estados finitos, etc. A continuación pasamos a detallar los más utilizados.

Modelos de Markov Ocultos.

La aproximación estadística más extendida es la basada en Modelos de Markov (MM) o n-gramas. Estos formalismos, ampliamente usados en el campo del reconocimiento automático del habla, fueron aplicados por primera vez al problema del etiquetado léxico en (Bahl and Mercer, 1976; Derouault and Merialdo, 1984; Church, 1988) y ya se han convertido en un referente para los posteriores trabajos. La técnica consiste en construir un ML estadístico que se utiliza para obtener, a partir de una frase de entrada, la mejor secuencia de estados en el modelo (secuencia de categorías léxicas) compatible con las probabilidades léxicas, aplicando el criterio de máxima verosimilitud.

El sistema *CLAWS* (Leech et al., 1983; Garside et al., 1987) utiliza un ML de bigramas para capturar la información contextual. Posteriormente, en (DeRose, 1988), se propone un esquema de programación dinámica para el cálculo eficiente de la mejor secuencia de estados en un MM. Además, en este trabajo el contexto se extiende a trigramas.

La estimación de los parámetros del modelo contextual se realiza a partir de textos etiquetados con la información de las categorías léxicas –*Métodos Supervisados*– (Church, 1988), (DeRose, 1988), (Merialdo, 1994) , (Weischedel et al., 1993) o bien usando textos no etiquetados –*Métodos No-Supervisados*– (Kupiec, 1992), (Cutting et al., 1992), (Samuelsson, 1993; Samuelsson, 1995). En el primer caso, el ML consiste en una estimación de las frecuencias de secuencias de categorías (de longitud prefijada) y de las apariciones de las palabras en las categorías. En el segundo, se reestima un ML inicial utilizando el algoritmo Baum-Welch (Baum, 1972). En ambos casos el ML se puede representar mediante un MM.

Las ventajas de los métodos *No-Supervisados* radica en que el aprendizaje se puede realizar a partir de textos no etiquetados. Este hecho los dota de una gran flexibilidad en la elección de etiquetas léxicas y en consecuencia, se facilita la transportabilidad de unas lenguas a otras, ya que el hecho de disponer de grandes cantidades de texto sin etiquetar, no supone ningún problema actualmente. Sin embargo, se requiere de un diccionario que suministre las posibles etiquetas para cada palabra. Además, para conseguir resultados de etiquetado aceptables, como se apunta en (Merialdo, 1994), el modelo inicial se debe estimar a partir de textos etiquetados

supervisados. En ese sentido los métodos *Supervisados* son los más ampliamente utilizados bajo esta aproximación.

Reglas de Transformación.

Los trabajos de *Brill* constituyen la principal aportación dentro de este paradigma. En (Brill, 1992), se introduce una técnica que consiste en el aprendizaje automático de reglas de transformación, que se construyen instanciando unos patrones previamente definidos, con el fin de corregir ciertos casos de ambigüedad contextual. La principal ventaja radica en que las reglas son aprendidas de manera automática a partir de corpus supervisados. Para ello, se hace un análisis de la salida de un etiquetador de unigramas (eligiendo la etiqueta de máxima probabilidad léxica) y se compara con la correspondiente referencia (texto desambiguado léxicamente y supervisado). Los patrones o reglas contextuales son de la forma "$tag_i \rightarrow tag_j si P''$, donde se indica que se debe cambiar la etiqueta (tag_i) por la (tag_j), para una determinada palabra, si se cumple la propiedad P. Las propiedades P reflejan las relaciones estructurales que sirven para deshacer la ambigüedad, por ejemplo, si la siguiente palabra a la que se está analizando (o la anterior o anteriores y posteriores) está etiquetada con cierta categoría, se deberá realizar el cambio indicado en la regla.

En trabajos posteriores (Brill, 1993a; Brill, 1995) se aplica la misma técnica de aprendizaje, pero utilizando la salida de un etiquetador probabilístico basado en trigramas, y se amplía el conjunto de reglas haciendo intervenir información contextual de las palabras. En este caso, las propiedades P pueden hacer referencia a ciertas palabras que sean muy frecuentes o que ayuden a resolver determinados casos de ambigüedad (preposiciones, relativos, etc.). Además, se añade un conjunto de reglas para predecir la categoría de las palabras desconocidas (no vistas en el entrenamiento), haciendo intervenir información morfológica de las mismas: contenido de letras mayúsculas, números, prefijos y/o sufijos de longitud predeterminada, etc.

En el proceso de aprendizaje también se debe determinar el orden de aplicación de las reglas de transformación. Para ello, se deben probar todas las combinaciones posibles de éstas (sobre un conjunto distinto al de evaluación) para elegir la que proporcione mejor resultado de etiquetado. Para optimizar el proceso de etiquetado

en (Roche and Schabes, 1995) se usa un formalismo mediante el cual las reglas se reescriben como traductores de estados finitos y se combinan en un único traductor. A pesar de que con esta transformación aumenta el tamaño del ML, también aumenta de manera considerablemente la rapidez del proceso de etiquetado. En (Samuel, 1998) se aplica una técnica que consiste en reducir el espacio de búsqueda de las reglas aplicables. Para ello, se utiliza una aproximación que emplea la técnica conocida como *'Monte Carlo'*, con lo que se consigue explorar sólo un subconjunto de reglas. La simplificación impuesta no reduce de manera significativa las prestaciones de etiquetado.

Árboles de Decisión.

Muchas son las aplicaciones que utilizan el formalismo de los árboles de decisión (Black et al., 1992)(Schmid, 1994)(Magerman, 1996)(Màrquez and Rodríguez, 1998), etc. En todas ellas se plantea el problema de etiquetado como un problema de clasificación. El ML está constituido por un conjunto de árboles de decisión estadísticos que se corresponden con ciertas clases de ambigüedad predefinidas por el usuario, por ejemplo, la clase de ambigüedad entre *Nombre-Adjetivo* o entre *Nombre-Verbo*, etc. Teniendo en cuenta estas clases, las probabilidades léxicas a priori de las palabras se recalculan dependiendo del camino seguido en el árbol.

Para un estudio más completo de estas aproximaciones, se puede consultar el trabajo (Màrquez, 1999), donde además, se presenta una excelente recopilación de las técnicas de aprendizaje de árboles de decisión y su uso en diferentes tareas de desambiguación léxica, sintáctica y semántica.

Aprendizaje basado en Memoria.

La idea básica del aprendizaje basado en memoria consiste en representar en memoria un conjunto de ejemplos extraídos de un corpus de entrenamiento. Cada ejemplo se corresponde con un caso de ambigüedad para determinadas palabras. Se representa para los diferentes contextos en que aparece la palabra (categorías y/o palabras que le preceden y le siguen) la categoría léxica correcta para ese caso. El proceso de etiquetado consiste en determinar para cada palabra de una frase de entrada, el ejemplo más adecuado almacenado en la memoria. Se trata de un problema de

clasificación, en el que se asigna a una palabra, la categoría adecuada atendiendo a sus vecinos más próximos, utilizando cierta función de disimilitud. Las prestaciones del sistema dependen en gran manera de esta función y de la representación elegida para los ejemplos. Si no se realiza ninguna simplificación, el método resulta excesivamente costoso. Por una parte, se necesita mucho espacio en memoria para almacenar los ejemplos. Por otra, el proceso de clasificación presenta un excesivo coste computacional, ya que para cada palabra a etiquetar, se debe comparar con todos los ejemplos representados en memoria.

El formalismo más ampliamente utilizado para la representación de los ejemplos son los árboles de decisión. Los trabajos de Daelemans (Daelemans et al., 1996b; Daelemans et al., 1996a), utilizan el algoritmo *IGTree* para optimizar tanto la representación, comprimiendo cada ejemplo base mediante una estructura de árbol, como para la clasificación, claramente favorecida por la estructura elegida.

Máxima Entropía.

Esta aproximación probabilística importa el concepto de *Máxima Entropía* introducido por Rosenfeld (Rosenfeld, 1996) en tareas de modelización del lenguaje y su posterior aplicación en tareas de reconocimiento automático de habla. Bajo este paradigma se construye un ML con el objetivo de maximizar la entropía de una distribución de probabilidad sujeta a ciertas restricciones. El ML debe ser consistente con los sucesos vistos en el entrenamiento y las restricciones impuestas, sin asumir ningún conocimiento sobre los sucesos no vistos.

El trabajo más relevante bajo esta aproximación es el de (Ratnaparkhi, 1996) donde se define un conjunto de características o heurísticos a explorar en el conjunto de datos de entrenamiento. Así, por ejemplo, para la desambiguación léxica de una palabra (w_i) se tienen en cuenta diversas fuentes de información al mismo tiempo. *Probabilidades léxicas*: $f(w_i, c_i)$, *información morfológica*: prefijos y sufijos de longitud predefinida, caracteres especiales, etc., *información contextual de categorías*: bigramas (c_{i-1}), trigramas (c_{i-2}, c_{i-1}), *información contextual de palabras*: palabras que le preceden w_{i-2}, w_{i-1} o le suceden w_{i+1}, w_{i+2}. La inferencia del ML consiste en la estimación de los parámetros que combinan estas características maximizando la entropía.

Aunque con esta aproximación se pueden combinar muchas y diversas características, en la práctica, se deben asumir ciertas simplificaciones para acotar el número de parámetros a estimar y así poder reducir el coste computacional del proceso de entrenamiento. Por ejemplo, en los trabajos de *Ratnaparkhi,* aquellas características observadas con frecuencia menores que un cierto umbral, no se consideran. A pesar de la simplificación, el entrenamiento es excesivamente costoso si se compara con otras aproximaciones. En el proceso de etiquetado, se enumeran todas las posibles secuencias de categorías compatibles con las restricciones y se elige la de máxima probabilidad, utilizando la técnica de búsqueda en haz ('*beam-search*') para optimizar el proceso.

Métodos híbridos.

En este apartado incluimos un conjunto de trabajos, que si bien utilizan métodos estadísticos, se caracterizan por la capacidad de combinar diferentes fuentes de conocimiento en la construcción del ML (*métodos híbridos*) o bien combinan diferentes etiquetadores para incrementar las prestaciones (*métodos combinados*).

En (Padró, 1996; Padró and Màrquez, 1998) encontramos un modelo híbrido basado en métodos de relajación. Mediante esta técnica se puede combinar diferentes fuentes de conocimiento –bigramas, trigramas, reglas de desambiguación definidas por lingüistas, árboles de decisión, etc.– para obtener ML más robustos.

Recientemente una de las tendencias seguidas para mejorar las prestaciones de etiquetado consiste en la combinación de diversos sistemas de etiquetado con el fin de intentar aprovechar las ventajas de cada uno de ellos. Los trabajos de (Halteren et al., 1998) y (Brill and Wu, 1998) constituyen las principales aportaciones en este campo. En estos trabajos, se hace cooperar los principales etiquetadores disponibles en la actualidad y que se basan en ML distintos: MM, árboles de decisión, reglas de transformación, máxima entropía, etc. La combinación se puede realizar utilizando una técnica de '*votación simple*', consistente en elegir como etiqueta correcta para una palabra, la que proponen la mayoría de los etiquetadores, bien dándoles el mismo peso a todos, o bien ponderándolos de acuerdo a algún criterio preestablecido: mejor precisión total, mejor precisión a nivel de categorías concretas, etc. También se aplican otras técnicas más sofisticadas para determinar en qué casos de ambigüedad

un etiquetador se comporta mejor que otro. Independientemente del método de combinación elegido, las prestaciones de etiquetado siempre se ven incrementadas.

La técnica de '*votación simple*' también se utiliza en (Màrquez et al., 1998), para obtener corpus de entrenamiento mayores y de mayor fiabilidad. Así, se amplia el conjunto de entrenamiento inicial, con datos etiquetados no supervisados, en los que un conjunto de etiquetadores coinciden en el etiquetado y en consecuencia, presentan un precisión global mayor mediante la combinación.

2.3 Evaluación de las Prestaciones de Etiquetado Léxico

La contrastación de las prestaciones de etiquetado entre las diferentes aproximaciones es una tarea difícil. Para que los resultados fueran directamente comparables, se tendrían que realizar en las mismas condiciones: la misma lengua, el mismo conjunto de datos de entrenamiento y de evaluación, el mismo conjunto de etiquetas léxicas, las mismas fuentes de información, etc.

Cada lengua presenta una problemática particular de ambigüedad, que en muchos casos, requiere de técnicas específicas que no es necesario aplicar en otras. Por ejemplo, para abordar el problema de las *palabras desconocidas*, para lenguas muy poco flexionadas, como el inglés, en muchos casos, es suficiente con unas cuantas reglas 'ad hoc' para predecir las posibles categorías de una palabras atendiendo a sus sufijos. Así, por ejemplo, sobre el corpus *WSJ*, el 98% de las palabras acabadas en *'able'* son adjetivos mientras que el resto son nombres. Esta situación no se da para otras lenguas como el castellano, catalán, francés, etc., que presentan una problemática mucho más profunda. Así, el tratamiento de los verbos y sus conjugaciones es mucho más complejo, lo cual requiere utilizar analizadores morfológicos, lematizadores y diccionarios para determinar las posibles formas verbales sin tener que almacenarlas explícitamente. El problema de los nombres propios es de más difícil solución, y requiere para un tratamiento correcto el uso de grandes diccionarios. Sin embargo, en la práctica con ciertos heurísticos, como por ejemplo considerar la inicial de la palabra, proporciona resultados aceptables para las palabras intermedias de una frase, no para los inicios evidentemente.

La elección del *conjunto de categorías léxicas* también tiene gran importancia en el proceso de desambiguación. En muchos casos será suficiente con la información de la categoría (verbo, nombre, adjetivo, ...), mientras por el contrario, ciertas ambigüedades requieren información más detalla, por ejemplo género y número para los nombres, modo, tiempo, persona para los verbos, etc.

En los trabajos de etiquetado léxico encontramos desde conjuntos reducidos de etiquetas, como el usado en el corpus *Wall Steet Journal (WSJ)* (Marcus et al., 1993), compuesto por 48 etiquetas, hasta otros más extensos de varios centenares como el *Susane* (Sampson, 1995) o en el usado por el etiquetador de *Xerox* tanto para su versión en inglés (Cutting et al., 1992) como para el castellano (Sánchez and Nieto, 1995). El incremento en el número de etiquetas, si bien proporciona mayor información lingüística para la desambiguación, también aumenta de manera considerable el número de parámetros a estimar en el ML, por lo que se debe llegar a una situación de compromiso entre estos dos factores.

Las prestaciones de un etiquetador léxico usualmente se evalúa mediante el parámetro denominado *Precisión (P)*. La precisión indica el número de etiquetas correctamente desambiguadas ($N_{correctas}$) en un conjunto de prueba compuesto por N muestras respecto a cierta referencia.

$$P = \frac{N_{correctas}}{N}$$

Para un determinado experimento se puede estimar un intervalo de confianza (I_C) asumiendo que la salida del etiquetador léxico –secuencia de etiquetas asociadas a las palabras– sigue una distribución binomial, es decir, una secuencia aleatoria de sucesos cuyo valor es correcto o incorrecto. En este caso, se define un I_C al 95% mediante la expresión

$$I_C = P \pm 1.96 \cdot \sqrt{\frac{P \cdot (1-P)}{N}}$$

donde N se debe considerar suficientemente grande para poder aproximar la distribución binomial por una distribución normal.

Por ejemplo, el I_C para un etiquetador que proporciona una P=97%, sobre un conjunto de prueba de N=100,000 palabras, valdrá:

$$I_C = 0.97 \pm 1.96 \cdot \sqrt{\frac{0.97 \cdot (1-0.97)}{100,000}} = 97.0\% \pm 0.1\%$$

Los resultados presentados en los últimos años sobre tareas de etiquetado alcanzan una precisión entre el 95% y 97%, para las aproximaciones de aprendizaje y resultados superiores para las aproximaciones lingüísticas. Estos resultados no son directamente comparables entre sí puesto que las condiciones de la experimentación no son las mismas: distintas lenguas y categorías léxicas, diferentes conjuntos de aprendizaje y evaluación, ... Una comparación exhaustiva entre una aproximación probabilística, basada en MM, y una basada en reglas definidas por expertos, se puede encontrar en (Samuelsson and Voutilainen, 1997).

Últimamente, han venido apareciendo diversos trabajos que intentan establecer comparaciones entre diferentes aproximaciones definiendo un marco común para el aprendizaje y la evaluación.

Recientemente, el trabajo de (Halteren et al., 1998) sobre el corpus LOB (Johansson, 1986) y el de (Brill and Wu, 1998) sobre el corpus WSJ, constituyen un ejemplo de comparación y cooperación entre diferentes aproximaciones inductivas al etiquetado léxico de textos.

En (Brants, 2000) también se presenta un etiquetador basado en MM y se hace una comparación con los trabajos anteriormente comentados. En este caso, se obtienen resultados superiores sobre el corpus WSJ, utilizando un modelo de trigramas (96.7%), frente al mejor de los resultados comparados, correspondiente a (Ratnaparkhi, 1996) que obtiene una precisión del (96.6%). Además, se destaca que, incluso utilizando los mismos datos y la misma aproximación, hay muchos factores que influyen en el resultado total: métodos de suavizado utilizados, tratamiento de las palabras desconocidas, método de evaluación, etc.

2.4 Análisis Sintáctico

El análisis sintáctico de textos escritos en lenguaje natural consiste en recuperar la estructura sintáctica o árbol sintáctico asociado a cada oración. Los algoritmos que llevan a cabo el análisis global proporcionan la estructura asociada a la oración cuando ésta pertenece al lenguaje definido por una gramática. En caso contrario, cuando la oración no pertenece al lenguaje definido, el análisis falla. Una descripción de distintos algoritmos 'clásicos' que realizan análisis global la podemos encontrar en (Allen, 1995). Estos algoritmos ofrecen buenos resultados para un lenguaje restringido, es decir, definido por una gramática de cobertura limitada.

La utilización de este tipo de algoritmos en el procesamiento de textos no restringidos presenta diversos problemas. Estos se derivan de la necesidad de definir una gramática de amplia cobertura capaz de recoger todas las estructuras del lenguaje. La definición de estas gramáticas es una tarea costosa y, además, presenta otros problemas (Briscoe, 1994) como:

1. La correcta *segmentación* en oraciones o unidades cuyos elementos mantengan una relación sintáctica en un texto.

2. La subgeneración ('*undergeneration*') que supone que la gramática no da cobertura a todo el lenguaje. Esto se produce debido a la complejidad de construir una gramática que reconozca todas las estructuras existentes en una lengua, que se ve sometida a continuos cambios y que puede adoptar diferentes formas dependiendo de factores como el entorno, el dominio, el grado de formalidad, etc. Además, por la propia naturaleza evolutiva de la lengua, siempre existirán oraciones cuya estructura gramatical no pueda derivarse a partir de la gramática, o en las que simplemente aparezcan palabras desconocidas que impidan que el análisis continúe.

3. La *ambigüedad* sintáctica según la cual a una oración pueden corresponderle varios análisis sintácticos. Cuanto mayor es la cobertura de una gramática, mayor es la ambigüedad que se produce. Para vocabularios o dominios restringidos, pueden aplicarse métodos, como es el uso de preferencias léxicas o de restricciones de selección, para escoger el árbol de análisis correcto entre todos los posibles. Sin embargo, estas técnicas no son directamente aplicables

en dominios no restringidos, ya que necesitan codificar una gran cantidad de conocimiento léxico, sintáctico y/o semántico, lo cual es una tarea realmente difícil y costosa. Por ello, en los últimos años se vienen desarrollando métodos inductivos, que utilizan técnicas de aprendizaje automático, que permiten resolver distintos tipos de ambigüedad: léxica, semántica, estructural, etc. Estos métodos construyen modelos del lenguaje a partir de corpora anotados con la información necesaria. Por contra, estas aproximaciones presentan el problema del elevado coste de anotación de los corpora. En (Young and Bloothooft (editors), 1997) se presentan diversas técnicas inductivas utilizadas en procesamiento del lenguaje tanto escrito como hablado.

Una posibilidad para abordar el análisis de textos no restringidos, y garantizar que éste sea robusto, consiste en aplicar técnicas de análisis parcial. El análisis parcial permite obtener la segmentación de la oración en unidades sintácticas de manera rápida y con una alta fiabilidad. Muchas aplicaciones no necesitan de un análisis completo de los textos de entrada y por lo tanto, son aplicables estas técnicas. Es el caso de tareas como la extracción de información, recuperación de información, generación de resúmenes, generación de índices, etc. Además, los analizadores parciales se utilizan para construir corpora de árboles sintácticos (*'treebanks'*) que puedan servir como información para el aprendizaje en aproximaciones inductivas. En este sentido, el analizador parcial reduce el coste de anotación del corpus por parte de expertos lingüistas, los cuales interactúan con el analizador, corrigiendo y completando los análisis propuestos.

Las principales características de un analizador parcial son:

- Utiliza algoritmos de análisis robustos, lo que significa que, independientemente de la estructura de la oración de entrada, se obtendrá una representación sintáctica de la misma, aunque sea parcial. Esto permite procesar cualquier texto no restringido. Además, se debe afrontar el problema de las palabras desconocidas, por lo que habitualmente un analizador parcial toma como entrada el texto previamente etiquetado con las categorías proporcionadas por un *etiquetador léxico*.

- Los algoritmos de análisis son más eficientes y menos costosos que los algoritmos de análisis global. Esto, junto a la alta fiabilidad en la detección de

determinadas estructuras sintácticas, permite que puedan utilizarse como una primera fase de procesamiento antes de construir el análisis global.

- En muchos casos, estos analizadores utilizan mecanismos o heurísticos que combinan los distintos análisis parciales para establecer relaciones sintácticas entre ellos, o incluso construir la representación completa de la oración.

La salida proporcionada por un analizador global es un árbol completo de análisis, si la oración es correcta gramaticalmente. El analizador parcial pospone las decisiones de ligamiento de constituyentes gramaticales si no tiene información suficiente. Por ejemplo, en la oración "Luis ve al hombre con el telescopio", un analizador global proporcionaría alguno de los análisis sintácticos de las figuras 2.2 y 2.3. Un analizador parcial proporciona el análisis de los constituyentes que puede identificar, según muestra la figura 2.4. Las decisiones de ligamiento entre constituyentes pueden resolverse a posteriori aplicando heurísticos, modelos probabilísticos, preferencias léxicas, etc. Así pues, la salida de un analizador parcial es un bosque de subárboles no entrelazados, es decir, que no comparten ningún nodo. Cada subárbol se corresponde con la estructura sintáctica de un constituyente oracional.

2.4.1 Análisis Parcial y Análisis Superficial

Existen diversos términos en la literatura que, aunque muchas veces se utilizan de forma indistinta, presentan diferencias en cuanto a la profundidad del análisis sintáctico que se lleva a cabo. Estos términos son análisis parcial (*'partial parsing'*) y análisis superficial (*'shallow parsing'* o *'chunking'*).

Según (Abney, 1997) el *análisis parcial* tiene como objetivo recuperar información sintáctica de forma eficiente y fiable, desde texto no restringido, sacrificando la completitud y profundidad del análisis global. Por lo tanto, las técnicas de análisis parcial deben permitir el análisis sintáctico de oraciones, obteniendo una representación solamente para aquellos constituyentes de la oración que pueden analizarse, sin preocuparse inicialmente de la construcción de una estructura sintáctica completa para la oración. Es decir, las técnicas de análisis parcial son capaces de identificar estructuras sintácticas recursivas.

El *análisis superficial* consiste en dividir el texto en segmentos no solapados que

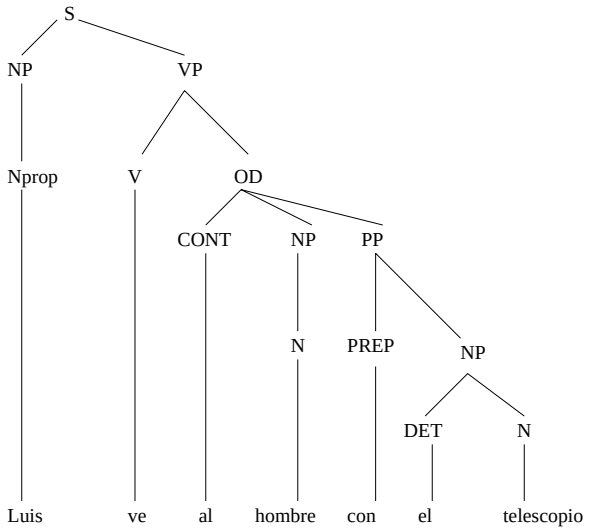

Figura 2.2: Análisis global a) de la oración "Luis ve al hombre con el telescopio"

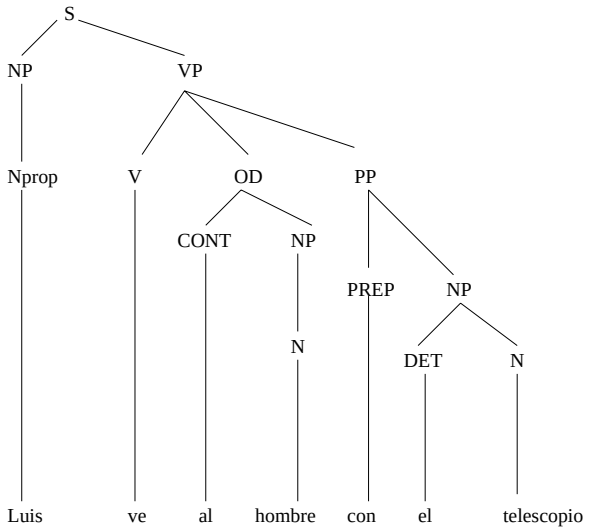

Figura 2.3: Análisis global b) de la oración "Luis ve al hombre con el telescopio"

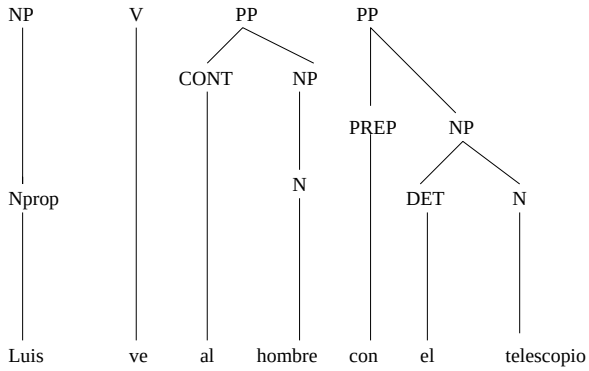

Figura 2.4: Análisis parcial de la oración "Luis ve al hombre con el telescopio"

se corresponden con ciertas estructuras sintácticas no recursivas, también llamadas con el término inglés '*chunk*'. Por ejemplo, se identifican sintagmas nominales (SN) no recursivos, también conocidos como sintagmas nominales básicos ('*baseNP*'), que son sintagmas nominales que no contienen a otros sintagmas nominales. Otras estructuras sintácticas que se pueden identificar en un texto son sintagmas adjetivos básicos (SADJ), núcleos verbales (SV), sintagmas adverbiales básicos (SADV), etc.

La segmentación de un texto en '*chunks*' puede representarse agrupando entre paréntesis o corchetes ('*brackets*') aquellas secuencias de palabras que forman parte del mismo sintagma. Por ejemplo, en la siguiente oración se han marcado los sintagmas nominales básicos:

$[_{SN}$ El cartero] da al $[_{SN}$ hombre] $[_{SN}$ una carta] .

El problema del análisis superficial también puede verse como un problema de etiquetado (Ramshaw and Marcus, 1995). En este caso, la identificación de '*chunks*' consiste básicamente en asignar a cada palabra la etiqueta del '*chunk*' correspondiente ya que cada palabra sólo puede pertenecer a un *'chunk'*. El tipo de representación utilizado es importante, sobretodo, en las aproximaciones basadas en métodos de aprendizaje automático que aprenden a partir de un corpus etiquetado. La utilización de un formato de etiquetado u otro puede influir incluso en las prestaciones de los analizadores. En (Tjong-Kim-Sang et al., 2000) se describen distintos tipos de etiquetado para identificar sintagmas nominales básicos. Todos estos formatos utilizan dos etiquetas: la *etiqueta I*, para las palabras que están dentro de un SN

básico, y la *etiqueta O*, para las palabras que están fuera de un SN básico[1]. Los formatos se diferencian en el tratamiento de las palabras que marcan el inicio o el final del SN. Se proponen los siguientes:

IOB1: La primera palabra dentro de un SN básico que sigue a otro SN básico recibe la *etiqueta B*.

IOB2: Todas las palabras que inician un SN básico reciben la *etiqueta B*.

IOE1: La última palabra dentro de un SN básico que precede a otro SN básico recibe la *etiqueta E*.

IOE2: Todas las palabras que finalizan un SN básico reciben la *etiqueta E*.

La oración anterior utilizando estas notaciones se presentaría así:

IOB1: El/I cartero/I da/O al/O hombre/I una/B carta/I ./O

IOB2: El/B cartero/I da/O al/O hombre/B una/B carta/I ./O

IOE1: El/I cartero/I da/O al/O hombre/E una/I carta/I ./O

IOE2: El/I cartero/E da/O al/O hombre/E una/I carta/E ./O

2.4.2 Medidas de Evaluación

Las medidas más habituales para comparar el rendimiento de los analizadores parciales son *Precisión ('precision')* y *Cobertura ('recall')*:

$$Precisión(P) = \frac{\#\ constituyentes\ correctos\ en\ el\ análisis\ propuesto}{\#\ constituyentes\ en\ el\ análisis\ propuesto}$$

$$Cobertura(C) = \frac{\#\ constituyentes\ correctos\ en\ el\ análisis\ propuesto}{\#\ constituyentes\ en\ el\ análisis\ de\ referencia}$$

[1]Cuando se consideran más unidades sintácticas, estas etiquetas se deben ampliar para identificar de qué *'chunk'* se trata ($\mathrm{I}_X\mathrm{OB}_X$, para indicar que se trata del *'chunk'* X).

Mediante la precisión medimos el grado de corrección que presentan los constituyentes detectados, mientras que con la cobertura se intenta medir si éstos cubren el conjunto que se pretende detectar.

Por ejemplo, si un determinado analizador proporciona la salida:

$[_{SN}$ El cartero] da al hombre $[_{SN}$ una carta] .

se observa que se han detectado 2 *SN* correctos, respecto a la referencia mostrada anteriormente, en la que aparecen 3 *SN*. En este caso,

$$P = \frac{2}{2} = 100\%$$
$$C = \frac{2}{3} = 66.6\%$$

Con el fin de establecer una medida que tenga en cuenta la precisión y la cobertura, se define el factor F_β como:

$$F_\beta = \frac{(\beta^2 + 1) \times P \times C}{\beta^2 \times P + C}$$

Con esta medida, y variando el valor de β, se puede dar más peso a un parámetro que a otro. Normalmente se toma β igual a 1, que significa que precisión y cobertura se consideran igualmente ponderados, con lo que

$$F_{\beta=1} = \frac{2 \times P \times C}{P + C}$$

El análisis superficial puede verse también como un problema de etiquetado, de manera que a cada palabra se le asigna una etiqueta que indica si pertenece o no a un determinado *'chunk'*. Por lo tanto, puede utilizarse la medida de *precisión de etiquetado ('accuracy')* –aunque no es una medida muy estandarizada y no existe una relación directa entre ésta y la precisión y la cobertura– definida como:

$$Precisión\,de\,etiquetado = \frac{\#\ etiquetas\,correctas\,en\,el\,análisis\,propuesto}{\#\ etiquetas\,en\,el\,análisis\,de\,referencia}$$

Por último, otra medida es el número de *Paréntesis Cruzados ("crossing brackets")*, es decir, el número de constituyentes que violan las marcas o fronteras de constituyente respecto a algún constituyente del corpus de referencia. Normalmente esta medida se utiliza para evaluar las prestaciones de los analizadores parciales.

2.5 Aproximaciones al Análisis Parcial y Superficial

Desde principios de los 90 se han desarrollado diversas aproximaciones para llevar a cabo análisis superficial. Al igual que en otros campos de la Lingüística Computacional, como es el caso del etiquetado léxico de textos, estas aproximaciones pueden clasificarse en dos grupos principales: las *Aproximaciones Lingüísticas*, que utilizan reglas gramaticales definidas manualmente por expertos mediante algún formalismo, y las aproximaciones que utilizan métodos de *Aprendizaje Automático*. La mayoría de estas aproximaciones tienen en común que toman como entrada la secuencia de etiquetas léxicas proporcionada por un etiquetador léxico. Es decir, el etiquetado léxico es un proceso previo al análisis parcial. Las distintas técnicas no solamente difieren en el método en sí, sino también en cuanto a la información que manejan: algunas técnicas únicamente tienen en cuenta la información de la categoría léxica, otras utilizan información morfológica, la propia palabra o su lema, la distancia entre palabras, las dependencias existentes entre núcleos y modificadores, etc. Muchas aproximaciones no son directamente comparables entre sí debido a que trabajan sobre distintos corpora, no identifican los mismos tipos de sintagmas, la profundidad del análisis varía, etc. Algunas aproximaciones, además de identificar los sintagmas o '*chunks*' correspondientes, también asignan roles o funciones sintácticas a los sintagmas nominales. A continuación se describen distintos trabajos aparecidos en la literatura, dentro de las aproximaciones comentadas, haciendo especial énfasis en la técnica de análisis utilizada.

2.5.1 Aproximaciones Lingüísticas

Estas aproximaciones consisten en definir un conjunto de reglas mediante algún formalismo gramatical y aplicar un método de análisis que permita procesar textos no restringidos de manera robusta. Los primeros trabajos, que pueden relacionarse con el análisis parcial, utilizan técnicas basadas en máquinas de estados finitos. En (Ejerhed, 1988) se definen patrones mediante expresiones regulares para el reconocimiento o segmentación del texto en cláusulas. Estos patrones se definen utilizando como terminales categorías léxicas y se aplicaron sobre el corpus *Brown*.

Fidditch, (Hindle, 1983), es uno de los analizadores más antiguos y que mejores resultados proporciona. Está especialmente diseñado para analizar texto no restrin-

gido o texto resultante de transcripciones de lenguaje hablado. Es un analizador determinista muy rápido (hasta 5,600 palabras por segundo en una estación de trabajo SGI). Identifica cláusulas, sujetos y predicados. Aquellas frases o cláusulas que no reconoce o que no puede ligar con el resto de constituyentes, no forman parte del árbol final de análisis. La estructura resultante es un bosque de subárboles. *Fidditch* ha sido utilizado para realizar análisis sintáctico de corpora en inglés, por ejemplo, en (Marcus et al., 1993) se analiza el corpus *Penn Treebank* corrigiendo y completando de forma manual la salida proporcionada por *Fidditch*.

Los primeros trabajos de Abney (Abney, 1991) introducen la noción de *'chunk'* y en ellos implementa el primer analizador que procesa oraciones según una gramática de *'chunks'*. El analizador se compone de un *'chunker'* que identifica diferentes *'chunks' (SN, SV, SP, ...)*, utilizando el algoritmo de análisis LR, y un *'attacher'* que enlaza los segmentos según las restricciones de selección impuestas por los núcleos y, en caso de ambigüedad, utiliza una serie de heurísticos. Ya que un *'chunk'* se corresponde con un sintagma no recursivo, su definición puede realizarse mediante patrones o definiciones regulares. En (Abney, 1996), se presenta un analizador basado en máquinas de estados finitos para la detección de *'chunks'* y cláusulas. Cada constituyente se define mediante una expresión regular. Estos patrones se organizan de manera incremental en cascada o niveles de análisis, así cualquier expresión regular de un nivel puede definirse utilizando patrones de niveles previos. Abney implementa el analizador *Cass* que parte de la secuencia de etiquetas léxicas y, primero realiza la segmentación y posteriormente, el ligamiento de los segmentos intentando completar el análisis. El análisis superficial se realiza usando únicamente patrones definidos con etiquetas léxicas. Para realizar el ligamiento, además se tiene en cuenta información léxica.

Bourigault (Bourigault, 1992) extrae SN mediante el analizador *LECTER*, utilizando reglas gramaticales y un conjunto de heurísticos. Consigue una cobertura del 95%, pero no se indica nada acerca de la precisión del analizador, lo cual no da una idea de su rendimiento real.

Voutilainen (Voutilainen, 1993) construye el analizador *NPtool*, basado en gramáticas de restricciones, para el reconocimiento de sintagmas nominales básicos. Mediante este formalismo se expresan restricciones que invalidan la formación de SN. Son restricciones del tipo "un determinante no puede preceder inmediatamente

a un verbo". Los SN identificados son el resultado de la intersección de dos modos de análisis: amigable (en caso de ambigüedad marca el segmento candidato como SN) y hostil (en caso de ambigüedad no lo marca como SN). Los resultados de precisión que ofrece están comprendidos entre el 95% y el 98% y la cobertura entre el 98.5% y el 100%, con lo que son los mejores presentados en la literatura para la detección de SN. Sin embargo, a la vista de los ejemplos mostrados como salida del analizador, se pueden observar ciertas incongruencias según se apunta en (Ramshaw and Marcus, 1995).

En (Grefenstette, 1996) se presenta una arquitectura basada en traductores de estados finitos que procesan texto previamente etiquetado con información morfológica: identifican grupos nominales y verbales, marcan los núcleos de estas agrupaciones, y establecen relaciones sintácticas entre núcleos.

Aït-Mokhtar (Aït-Mokhtar and Chanod, 1997) presenta una arquitectura incremental que consiste en una secuencia de traductores construidos a partir de expresiones regulares. Cada transductor lleva a cabo una tarea lingüística determinada y el analizador funciona en tres pasos: 1) reconoce estructuras sintácticas no recursivas o segmentos (SN, SP, SV); 2) identifica roles sintácticos tales como sujetos y complementos aplicando un conjunto de restricciones; 3) expande los sintagmas verbales hacia la derecha, lo cual da buenos resultados en textos escritos en un lenguaje controlado. Esta aproximación se desarrolló para el francés y, posteriormente, en (Pavia, 1999) se presentó para el castellano.

2.5.2 Aproximaciones de Aprendizaje Automático

Las técnicas de aprendizaje automático construyen un modelo de lenguaje a partir de corpora parentizados y etiquetados con categorías léxicas. Como se comentó anteriormente, estas aproximaciones difieren entre sí en el método de aprendizaje y en los parámetros que forman parte del modelo y que, por lo tanto, deben estimarse. Existen métodos que trabajan únicamente con la información de las categorías léxicas y el parentizado, otros incorporan información morfológica, del lema, distancia entre palabras, etc. Ya que el problema del análisis superficial puede verse como un problema de etiquetado, las técnicas aplicadas para el etiquetado léxico pueden generalizarse para este caso. Así, los trabajos más significativos aplican las mismas

técnicas utilizadas en el campo del etiquetado léxico de textos: modelos de Markov, reglas de transformación, árboles de decisión, máxima entropía, etc. Éstas y otras aproximaciones se presentan a continuación.

Modelos de Markov Ocultos.

La idea de aplicar un modelo estadístico para realizar análisis superficial fue presentada por K.W. Church. En concreto, en (Church, 1988) se aplica esta técnica para la detección de sintagmas nominales no recursivos. El método calcula, a partir de un corpus etiquetado con categorías léxicas y parentizado semiautomáticamente, una matriz de probabilidades. Esta matriz indica con que probabilidad puede ocurrir un delimitador de SN, bien sea comienzo ('[') o final (']'), entre dos etiquetas léxicas. Dada una secuencia de etiquetas léxicas como entrada, el analizador estocástico inserta los comienzos o finales de SN más probables, utilizando un algoritmo de programación dinámica basado en el algoritmo de Viterbi (Viterbi, 1967). Aunque esta aproximación ofrece buenos resultados hay que tener en cuenta que la definición de la estructura de los NP es muy sencilla, la evaluación se ha realizado sobre una muestra pequeña (solamente 243 SN) y, finalmente, la entrada presenta un error de etiquetado entorno al 0.5% lo cual no corresponde a una situación real utilizando los sistemas existentes de etiquetado automático (del orden del 3%).

En (Skut and Brants, 1998b) se presenta una aproximación estocástica, que puede verse como una generalización del trabajo de Church, para el reconocimiento de estructuras sintácticas de profundidad limitada. La información estructural se codifica en una etiqueta (r_i) que expresa la relación entre una palabra w_i y su antecesora w_{i-1}. Así, se indica si w_i y w_{i-1} tienen el mismo nodo padre o si el nodo padre de alguna de ellas, es a su vez ancestro de la otra, en estructuras de profundidad menor o igual a 3. Además, el modelo se enriquece con la información de la categoría léxica (t_i) y la categoría sintáctica de la palabra (c_i). El análisis parcial se resuelve como un problema de etiquetado que asigna la secuencia más probable de *etiquetas estructurales* $\mathcal{S} = \langle S_0, S_1, \ldots, S_n \rangle$ a la secuencia de etiquetas léxicas $\mathcal{T} = \langle t_0, t_1, \ldots, t_n \rangle$, donde cada etiqueta estructural S_i es una tupla $\langle r_i, t_i, c_i \rangle$. El modelo contextual se estima como un modelo de trigramas suavizado mediante interpolación lineal con bigramas y unigramas.

Otra aproximación que combina técnicas de modelos de Markov con reglas incontextuales se presenta en (Brants, 1999). El análisis se realiza en cascada, siguiendo una arquitectura similar a la desarrollada en los trabajos de Abney, pero la estructura de cada nivel se representa mediante un MM previamente estimado a partir de un corpus analizado. Cada nivel proporciona como salida las n mejores hipótesis según el modelo, representadas mediante un grafo en el cual los nodos son posiciones entre palabras y los vértices son categorías léxicas o sintácticas. El nivel 0 de análisis toma como entrada la secuencia de palabras y produce el etiquetado léxico. Los niveles posteriores toman como entrada el grafo de posibles hipótesis y se resuelven aplicando una modificación del algoritmo de Viterbi. El nivel 1 se corresponde con la tarea de segmentado en *'chunks'*. Los estados de cada MM pueden representar categorías léxicas o categorías no-terminales (sintagmas o frases). En el primer caso, los estados emiten palabras; en el segundo caso, los estados emiten árboles parciales de análisis, que se corresponden con una estructura sintáctica determinada, y la probabilidad de emisión se corresponde con la probabilidad de la correspondiente regla incontextual. Las probabilidades de transición entre estados se estiman de la forma habitual.

Reglas de Transformación.

En (Ramshaw and Marcus, 1995) se aplica el método de aprendizaje basado en reglas de transformación (*'Transformation-Based Learning'*, TBL) desarrollado por Brill (Brill, 1993a). Este mismo método había sido utilizado anteriormente para tareas de etiquetado léxico (Brill, 1992) y análisis sintáctico de corpus (Brill, 1993b). En este trabajo se realiza únicamente detección de SN básicos. Es el primer trabajo en el cual se plantea al análisis superficial como una técnica de etiquetado. Utiliza el conjunto de etiquetas IOB1: una palabra se puede etiquetar como I (si pertenece a un SN), O (si está fuera de un SN) o B (si es principio de SN, siendo la palabra anterior un final de SN).

El método de aprendizaje consta básicamente de los siguientes pasos: 1) se realiza una asignación inicial de etiquetas de *'chunk'* a cada palabra del corpus. Esta inicialización consiste en asignar la etiqueta de *'chunk'* más probable para cada etiqueta léxica, según el corpus de referencia. 2) Se define un conjunto de reglas o patrones formados por un número limitado de rasgos. Los rasgos se definen

basándose en las palabras situadas a una distancia limitada respecto a la palabra actual, (p.e. la palabra actual, la palabra situada una posición a la izquierda, etc.) y a etiquetas (la etiqueta de la palabra actual, la etiqueta de la palabra situada a la izquierda, etc.). 3) Se compara el análisis del corpus de referencia respecto al análisis resultado de la inicialización. En cada punto en el cual la predicción inicial no ha sido correcta se instancian los patrones definidos y así, se construye un conjunto de reglas candidatas (estas reglas indican que una etiqueta i debe cambiarse por la etiqueta j si se da el patrón P). Posteriormente, siguiendo un proceso iterativo, se ordenan las reglas de manera que el efecto positivo de su aplicación sobre el corpus sea máximo.

El método de etiquetado consiste en aplicar las reglas de transformación, en el orden establecido, a un etiquetado inicial. Con esta aproximación se obtienen buenos resultados en la detección de SN básicos (ver tabla 2.1), aunque las prestaciones disminuyen cuando no se utiliza información léxica. El principal problema de esta aproximación, como se ha comentado anteriormente, radica en el elevado coste computacional del proceso de aprendizaje, a la hora de establecer el mejor orden de aplicación de las reglas.

Aprendizaje basado en Memoria.

Existen varios trabajos que utilizan la aproximación conocida como *'Memory-based Learning'* (MBL), normalmente como una generalización de las técnicas de etiquetado léxico. Bajo esta aproximación se construye un clasificador, para una determinada tarea, almacenando un conjunto de ejemplos de la misma. En primer lugar, cada ejemplo, obtenido del conjunto de entrenamiento, se representa mediante un vector de características o rasgos que definen una clase determinada. En la fase de análisis, dado un nuevo vector de características, asociado a una palabra del conjunto de prueba, el algoritmo de clasificación le asigna una clase de entre aquellas correspondientes a los vectores más similares almacenados en la memoria. Para que esta aproximación sea eficiente, tanto en términos de coste computacional y espacial como en términos de precisión, es necesario definir correctamente varios aspectos: 1) La estructura de datos para almacenar eficientemente los casos o ejemplos; es habitual utilizar árboles de decisión, en los que los arcos de un camino almacenan valores de rasgos y las hojas del árbol son la clase correspondiente a ese camino.

Además se suelen utilizar diversos mecanismos de compresión que permiten almacenar y recorrer los árboles de manera eficiente (Veenstra, 1998). 2) La definición de los rasgos o contexto, que evidentemente es dependiente de la tarea. 3) La función de disimilitud, que influye de manera determinante en la precisión del método y que también puede variar según el tipo de tarea.

En (Veenstra, 1998) se aplica MBL para la detección de sintagmas nominales básicos. En concreto, se utiliza el algoritmo *IGTree* (Daelemans et al., 1997) que compacta los ejemplos en árboles y permite recuperar la información desde estos. Los vectores de rasgos almacenan la palabra foco, dos palabras a la izquierda de la palabra foco y una palabra a la derecha, además de la etiqueta léxica asociada a cada una de estas palabras. La salida de este clasificador (IGTree1) puede corregirse aplicando un nuevo clasificador (IGTree2) cuyo modelo consta de los siguientes rasgos: etiqueta léxica y etiqueta de chunk de la palabra foco, de dos palabras a su izquierda y de una palabra a su derecha. En (Daelemans et al., 1999), se aplica MBL para la detección de sintagmas nominales y verbales básicos. En este caso, se considera como contexto cinco palabras y etiquetas léxicas a la izquierda de la palabra foco y tres a su derecha. En (Veenstra, 1999) se amplía el trabajo anterior para la detección de sintagmas preposicionales básicos, obteniendo valores de precisión y cobertura entre el 94% y el 95%. Estos resultados se obtienen partiendo del etiquetado proporcionado por el *WSJ*, en lugar de tomar la salida de un etiquetador, como se realiza en otras aproximaciones. El algoritmo *IB1-IG* se utiliza en (Tjong-Kim-Sang and Veenstra, 1999), aplicado a la detección de sintagmas nominales básicos. Además, en este trabajo se estudia cómo afecta el formato de representación de los *'chunks'* (IOB1, IOB2, IOE1, IOE2) sobre las prestaciones del clasificador. En este caso, la utilización de etiquetas IOB1 es la que ofrece mejores resultados.

En (Argamon et al., 1998) se utiliza el algoritmo *'Memory-Based Sequence Learning'* (MBSL) para la detección de sintagmas nominales básicos. En la fase de entrenamiento, en lugar de codificar el contexto, se almacenan todas las subcadenas o secuencias de categorías que contienen un *'chunk'* teniendo en cuenta un contexto limitado. Para cada subcadena se contabilizan las evidencias positivas (aquellas que contienen un delimitador de *'chunk'*) y las negativas (la misma secuencia de categorías, pero que no contienen el delimitador de *'chunk'* o éste aparece en otra posición). Para cada oración de entrada todos los parentizados son posibles y se

selecciona aquél que tenga una mayor puntuación. Una secuencia de categorías que contiene un delimitador de *'chunk'* es un *'tile'*. Cada hipótesis candidata puede verse como varios *'tiles'* conectados. La puntuación de cada hipótesis depende de una serie de parámetros: la puntuación (frecuencia de evidencias positivas frente al total) de cada *'tile'*, el número de diferentes combinaciones de *'tiles'* que cubren la hipótesis, el solapamiento entre *'tiles'*, etc.

Máxima Entropía.

En (Skut and Brants, 1998a) se aplica el método de estimación de Máxima Entropía, el cual permite combinar distintos parámetros o fuentes de conocimiento para estimar el modelo contextual. Es una aproximación estadística similar a la presentada en (Skut and Brants, 1998b), que difiere en la manera de calcular el modelo contextual: en este caso obtiene una distribución de probabilidad que maximiza la entropía del modelo. Se utilizan las etiquetas estructurales, S_i, definidas de la forma $\langle r_i, t_i, c_i \rangle$, explicadas anteriormente. Teniendo en cuenta un modelo de n-gramas ($n \leq 3$) se define un conjunto de patrones de parámetros extrayendo los atributos del contexto del n-grama que sean relevantes. Esto da lugar a n-gramas parciales en los que no se tienen en cuenta todos los atributos de las etiquetas estructurales. Esta aproximación mejora el rendimiento del modelo de trigramas utilizado en (Skut and Brants, 1998b), aunque se observa que ambos métodos ofrecen prestaciones similares cuando crece el tamaño del conjunto de entrenamiento.

Otras aproximaciones.

La aproximación descrita en (Cardie and Pierce, 1998) identifica sintagmas nominales básicos aplicando las reglas gramaticales extraídas del corpus anotado con estos sintagmas. En primer lugar se obtiene, directamente desde el corpus de entrenamiento, una gramática inicial para SN. El método de análisis es sencillo: se emparejan secuencias de etiquetas léxicas con las reglas gramaticales; si puede aplicarse más de una regla, se escoge aquella que cubra un número mayor de etiquetas léxicas (*'longest matching'*). Para mejorar las prestaciones de la gramática, se eliminan aquellas reglas cuya precisión este por debajo de un determinado umbral. En esta aproximación solamente se tiene en cuenta la información proporcionada por la

etiqueta léxica.

El método de aprendizaje SNoW (*'Sparse Network of Winnows'*) se aplica en (Muñoz et al., 1999) para la detección de sintagmas nominales básicos. En este trabajo se estudia cómo afecta sobre las prestaciones del analizador, el tipo de representación utilizada para marcar los *'chunks'*: se obtienen mejores resultados usando los marcadores corchete abierto/cerrado (*'Open/Close Predictors')* en lugar de las etiquetas IOB (*'Inside/Outside Predictors').*

Métodos Híbridos.

Algunos trabajos combinan la aproximación lingüística con los métodos de aprendizaje automático. En (Voutilainen and Padró, 1997), se combina un modelo de n-gramas con un conjunto de restricciones sintácticas contextuales para la detección de sintagmas nominales. Se utiliza el *método de relajación*: dado un conjunto de etiquetas, de variables y de restricciones, obtiene la combinación de etiquetas asociadas a cada variable que maximiza el valor de 'consistencia global'. Las restricciones que intervienen en el modelo son bigramas, trigramas y restricciones lingüísticas definidas manualmente. En (Chen and Chen, 1995) se construye un sistema que extrae algunos tipos de sintagmas nominales recursivos. Inicialmente, segmenta el texto en sintagmas nominales básicos utilizando un analizador probabilístico basado en bigramas. Posteriormente, conecta los segmentos mediante un gramática regular definida manualmente.

En (Osborne, 1999) se implementa un método inductivo, basado en el principio de *longitud de descripción máxima*, para aumentar la cobertura de una gramática DCG probabilística, partiendo de un conjunto inicial de reglas escritas a mano. Debido a esto, también puede considerarse un método híbrido ya que parte de una gramática definida manualmente por un experto. No es un trabajo comparable al resto ya que se evalúa sobre a una tarea más compleja como es la detección de sintagmas nominales recursivos.

Métodos Combinados.

Otra reciente aproximación consiste en combinar el resultado proporcionado por distintos analizadores. Es una generalización de trabajos similares llevados a cabo para sistemas de etiquetado léxico (Halteren et al., 1998). En (Tjong-Kim-Sang, 2000a), se aplica un algoritmo de MBL (*IB1-IG*, (Daelemans et al., 1996b)) utilizando distintos modos de representación de los *'chunks'*: etiquetado dentro/fuera (IOB1, IOB2, IOE1, IOE2) o corchetes abierto/cerrado (O+C). Ya que la forma de representar los datos de aprendizaje influye en las prestaciones de los clasificadores, se obtienen cinco clasificadores distintos. La técnica de selección de la etiqueta correcta es la de 'votación': se asigna un peso a los resultados proporcionados por cada clasificador; para cada token de entrada, se escoge como etiqueta aquella cuya suma de pesos sea mayor. La asignación de pesos se puede realizar de diversas maneras: asignar el mismo pesos a todos (mayoría o *votación simple*), asignar como peso la precisión del etiquetador, asignar la precisión computada para cada etiqueta , etc. Los resultados obtenidos mejoran las prestaciones en comparación con cada sistema individual, aunque la forma de asignación de los pesos no influye de forma significativa. La misma técnica se ha aplicado utilizando analizadores diferentes (Tjong-Kim-Sang et al., 2000) y sobre un conjunto mayor de *'chunks'* (Tjong-Kim-Sang, 2000b).

2.6 Resultados sobre Análisis Superficial

Uno de los principales problemas cuando se evalúan distintos métodos es la elección del conjunto de datos, tanto para la fase de entrenamiento de los modelos como para la fase de prueba. En este sentido, el conjunto de datos utilizado por L. Ramshaw y M. Marcus en su trabajo (Ramshaw and Marcus, 1995) se ha convertido en un conjunto de datos estándar para evaluar un analizador superficial[2]. Este conjunto está formado por las secciones 15 a 18 del corpus *Wall Street Journal* como material de entrenamiento y la sección 20 de este corpus para prueba. Estos corpora están etiquetados utilizando el etiquetador de Brill (Brill, 1994) sin realizar ningún tipo de supervisión. Cualquier método de aprendizaje debería evaluarse respecto a este conjunto de datos con el fin de poder comparar sus prestaciones con otras

[2]Este conjunto de datos puede obtenerse en la dirección ftp://ftp.cis.upenn.edu/pub/chunker/

aproximaciones. La tabla 2.1 resume los resultados publicados correspondientes a la detección de sintagmas nominales básicos, para ese conjunto estándar de datos.

	P (%)	C (%)	$F_{\beta=1}$	Técnica
(Tjong-Kim-Sang, 2000a)	93.6	92.9	93.3	Combinación
(Muñoz et al., 1999)	92.4	93.1	92.8	SNoW
(XTAG, 1998)	91.8	93.0	92.4	TAG
(Tjong-Kim-Sang and Veenstra, 1999)	92.5	92.3	92.4	MBL (IB1-IG)
(Ramshaw and Marcus, 1995)	91.8	92.3	92.0	TBL
(Argamon et al., 1998)	91.6	91.6	91.6	MBSL *
(Veenstra, 1998)	89.0	94.3	91.6	MBL (IGTree2)
(Daelemans et al., 1999)	91.6	91.5	91.6	MBL (IB1-IG)
(Cardie and Pierce, 1998)	90.7	91.1	90.9	PTG *
(Ramshaw and Marcus, 1995)	90.5	90.7	90.6	TBL *
(Veenstra, 1998)	83.1	97.0	90.1	MBL (IGTree1)
(Cardie and Pierce, 1999)	89.0	90.9	89.9	PTG *

Tabla 2.1: Resultados de diversas aproximaciones en la detección de SN básicos. Aquellas aproximaciones marcadas con (*) únicamente utilizan la información de la etiqueta léxica.

Capítulo 3

Etiquetado Léxico basado en Modelos de Markov

3.1 Introducción

En este capítulo se plantea el problema de etiquetado léxico de textos desde la perspectiva estadística. Un formalismo ampliamente usado son los Modelos de Markov o n-gramas. Por simplicidad en la presentación, se particularizará para el caso de bigramas. La extensión para n-gramas con n distinto de 2 se puede realizar de manera directa. Se presentaran las diferentes técnicas de aprendizaje de los parámetros, así como las técnicas de suavizado asociadas propuestas. Se introducirán los algoritmos utilizados para el etiquetado así como los trabajos más relevantes publicados que utilizan estas aproximaciones. Finalmente se presenta una extensión de los modelos contextuales con el fin de establecer restricciones léxico-contextuales en los mismos.

3.2 Formulación Probabilística del Problema de Etiquetado Léxico

Sea W una frase de longitud T, formada por la secuencia de palabras $w_1, w_2, \ldots, w_T$. Un etiquetador léxico (ver figura 3.1) se puede considerar como una función φ que asigna a cada palabra w_i de la frase de entrada W, su correspondiente categoría c_i

perteneciente a un conjunto de categorías léxicas $\mathcal{C}$ definido previamente.

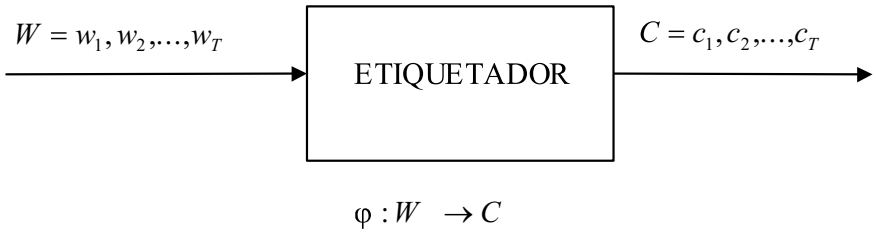

Figura 3.1: Descripción funcional de un etiquetador.

Desde un punto de vista estocástico, se trata de un problema de optimización en el que, dada una frase $w_1, w_2, \ldots, w_T$ y un conjunto de categorías léxicas $\mathcal{C}$, se desea encontrar la secuencia de categorías que maximice la siguiente función de probabilidad condicionada:

$$\widehat{C} = \arg \max_{C=c_1,...,c_T} (P(c_1, ..., c_T | w_1, ..., w_T)), \text{ donde } C \in \mathcal{C}^T \tag{3.1}$$

Utilizando la regla de Bayes, podemos reescribir la ecuación (3.1) como

$$\widehat{C} = \arg \max_{C} \left(\frac{P(c_1, ..., c_T) \cdot P(w_1, ..., w_T | c_1, ..., c_T)}{P(w_1, ..., w_T)} \right) \tag{3.2}$$

Dado que el proceso de maximización se realiza sobre las secuencias de categorías C y $P(w_1, ..., w_T)$ es independiente de éstas, es suficiente con maximizar el numerador

$$\widehat{C} = \arg \max_{C} \; P(c_1, ..., c_T) \cdot P(w_1, ..., w_T | c_1, ..., c_T) \tag{3.3}$$

El primer término de esta expresión $P(c_1, ..., c_T)$ se corresponde con lo que llamaremos *probabilidades de contexto o ML contextual* y el segundo $P(w_1, ..., w_T | c_1, ..., c_T)$ con las llamadas *probabilidades léxicas*[1] que representan la relación entre el vocabulario y las categorías léxicas.

[1]En algunas publicaciones (Allen, 1995) a estas probabilidades se les llama de *generación léxica* y se reserva el nombre de *probabilidad léxica* para las simétricas $P(c_i|w_i)$.

Para obtener $\widehat{C}$ (ecuación 3.3) , se debería considerar todas las posibles secuencias de categorías de longitud T que se pueden formar con un conjunto de N categorías (N^T secuencias) y elegir de entre ellas la de mayor probabilidad.

Por ejemplo, si tenemos la frase W= *"Este río esta seco"*, y considerando el conjunto de categorías $\mathcal{C} = \{$"D", "P", "V", "N", "A"$\}$[2], el número de secuencias posibles, asumiendo que cada palabra puede pertenecer a todas las categorías, es de $5^4 = 625$. En la figura 3.2 se representa gráficamente todas las secuencias posibles[3].

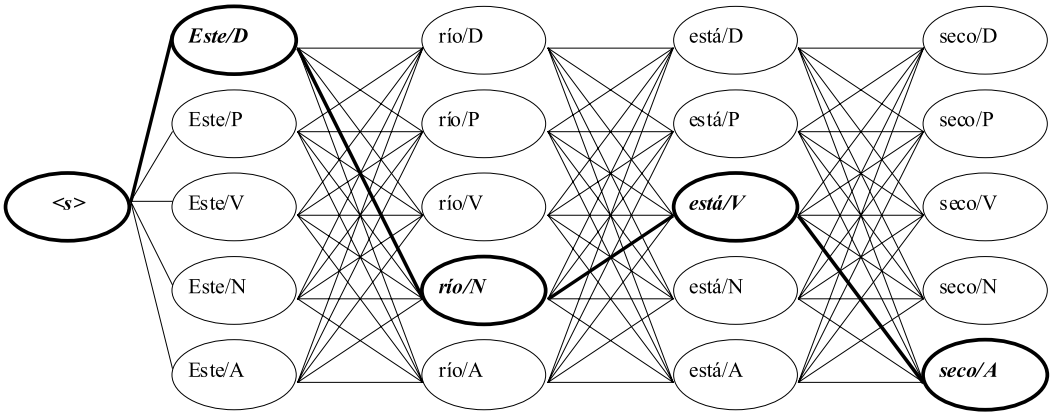

Figura 3.2: Representación de las secuencias de categorías léxicas posibles para la frase "Este río está seco".

De entre todas las secuencias de categorías posibles para esta frase, el etiquetador deberá seleccionar la de mayor probabilidad. En este caso, sólo hay una secuencia gramaticalmente correcta (marcada en trazo más grueso) y ésta deberá coincidir con la secuencia de mayor probabilidad ($\widehat{C}$ = D N V A):

$$P(\text{ D N V A} \mid \text{Este río está seco}) =$$

$$= [P(D| < s >) \cdot P(N|D, < s >) \cdot P(V|N, D, < s >) \cdot P(A|V, N, D, < s >)] \cdot$$

$$\left[\begin{array}{c} P(Este|D) \cdot P(río|N, Este\_D) \cdot P(está|V, río\_N, Este\_D) \cdot \\ P(seco|A, está\_V, río\_N, Este\_D) \end{array} \right]$$

[2] **D**eterminante, **P**ronombre, **V**erbo, **N**ombre, **A**djetivo.

[3] <s> Representa el símbolo de inicio de frase.

donde el primer producto representa las probabilidades de contexto y el segundo las probabilidades léxicas.

En general, una palabra no pertenece a todas las categorías léxicas definidas, por eso, en la práctica se utilizan analizadores morfológicos y/o diccionarios que nos proporcionan las posibles categorías léxicas para cada palabra. Esta información reduce substancialmente el número de secuencias posibles de categorías para una determinada frase y por lo tanto, el espacio de búsqueda de la solución óptima.

Para el ejemplo de la figura 3.2, y considerando la información citada, el número de secuencias se reduce de 625 a 12 (3 x 2 x 1 x 2), como se puede observar en la figura 3.3. El inconveniente que representa el uso de analizadores morfológicos es que no siempre proporcionar todas las posibles categorías léxicas de una palabra (en la práctica se da esta situación). Cuando esto ocurre, algunas de las secuencias lingüísticamente posibles no serán consideradas y por lo tanto, la solución encontrada nunca será la correcta.

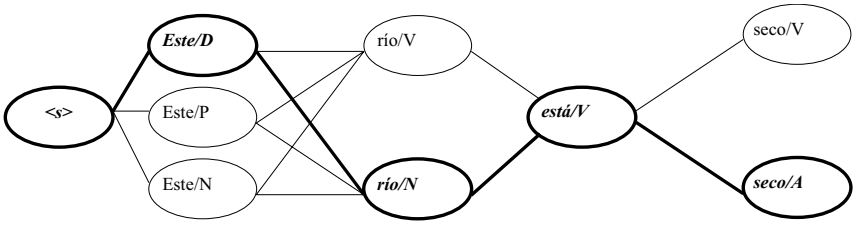

Figura 3.3: Representación de las secuencias posibles para la frase "Este río está seco" compatibles con el análisis morfológico.

3.2.1 Algunas Simplificaciones al Problema de Etiquetado

Para resolver la ecuación 3.3 se suelen introducir algunas aproximaciones (asunciones de Markov), que aunque no siempre proporcionan la solución exacta, permiten obtener resultados bastante precisos de etiquetado léxico, con costes computacionales aceptables y estableciendo los modelos a partir de un número de muestras de aprendizaje razonable.

Sobre el Modelo Probabilístico Contextual

En este caso *se supone que la probabilidad de aparición de una categoría sólo depende de un cierto número de categorías que le preceden.* Esta es la aproximación conocida como n-gramas, donde los n más utilizados son $n = 2$ (*bigramas*) y $n = 3$ (*trigramas*). Para el caso de bigramas tendremos que la probabilidad contextual se puede aproximar como:

$$P(c_1, ..., c_T) \approx \prod_{i=1...T} P(c_i|c_{i-1}) \tag{3.4}$$

En el ejemplo presentado, tendríamos que la probabilidad de contexto sería:

$$P(D| < s >)P(N|D)P(V|N)P(A|V)$$

Sobre las Probabilidades Léxicas

En este caso *se considera que el hecho de que una palabra aparezca en una determinada categoría es independiente de las palabras que aparecen en las categorías que le preceden y le siguen.*

$$P(w_1, ..., w_T|c_1, ..., c_T) \approx \prod_{i=1...T} P(w_i|c_i) \tag{3.5}$$

En el ejemplo visto las probabilidades léxicas vendrán dadas por la expresión

$$P(Este|D)P(río|N)P(está|V)P(seco|A)$$

Considerando las aproximaciones (3.4) y (3.5), el problema (3.3) se reduce a maximizar la siguiente expresión:

$$\widehat{C} = \arg\max_{C} \left(\prod_{i=1...T} P(c_i|c_{i-1}) \cdot P(w_i|c_i) \right) \tag{3.6}$$

3.2.2 Modelos de Markov y Etiquetado Léxico

La información involucrada en la ecuación (3.6) se puede representar mediante un Modelo de Markov (MM) de primer orden.

Un modelo de Markov (MM) se define, utilizando la notación presentada en (Rabiner and Juang, 1986), como un triplete $\mathcal{M} = (\Pi, A, B)$, donde:

- N es el número de estados[4].
- M es la talla del vocabulario.
- $Q = \{q_1, q_2, ..., q_N\} \cup \{q_0, q_F\}$ es el conjunto de estados, siendo q_0 el estado inicial y q_F el estado final.
- $V = \{v_1, v_2, ..., v_M\}$ es el conjunto de símbolos.
- $A = \{a_{ij}\}$ es la probabilidad de transición del estado q_i al estado q_j. Alternativamente se presentará como $P(q_j|q_i)$ donde $1 \leq i, j \leq N$, o como $P(C_j|C_i)$, para el caso de bigramas, entendiendo que C_i es la categoría asociada al estado q_i .
- $B = \{b_{jk}\}$ es la probabilidad de emisión del símbolo v_k en el estado q_j. Alternativamente usaremos $P(v_k|q_j)$ donde $1 \leq j \leq N, 1 \leq k \leq M$, o como $P(w_j = v_k|C_j)$ para el caso de bigramas.
- $\Pi = \{\pi_i\}$ probabilidad de transición del estado inicial (q_0) al estado q_i, representado también como $P(q_i|q_0), 1 \leq i \leq N$ o también como $P(q_i| < s >)$ para bigramas si tenemos el estado inicial etiquetado con este símbolo.

En la figura 3.4 se representa mediante un MM las informaciones utilizadas en el problema de etiquetado léxico. A cada estado se le asocia una etiqueta léxica, las probabilidades de contexto se corresponden con las probabilidades de transición entre estados y las probabilidades léxicas con las probabilidades de emisión de símbolos en cada estado.

[4]En el caso de un modelo de bigramas, N coincide con el número de categorías léxicas utilizado. Para trigramas N coincide con el número de pares de categorías y así sucesivamente para cualquier n-grama de orden superior.

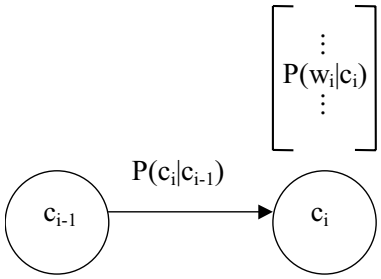

Figura 3.4: Representación de las probabilidades de contexto y léxicas mediante un modelo de Markov.

3.3 Algoritmos de Etiquetado

Los algoritmos usualmente empleados para resolver el problema del etiquetado léxico son el algoritmo de *Viterbi* y el algoritmo *Forward-Backward*.

Mediante el algoritmo de *Viterbi* (Viterbi, 1967) se obtiene la secuencia de estados de mayor probabilidad en un MM, para una determinada frase de entrada $W = w_1, ..., w_T$, compatible con las probabilidades léxicas. Es decir resuelve la ecuación:

$$\widehat{C} = \arg\max_{C} \left(\prod_{i=1...T} P(c_i|c_{i-1}) P(w_i|c_i) \right) \tag{3.7}$$

El algoritmo *Forward-Backward* (Baum, 1972) puede ser utilizado para elegir la etiqueta mas adecuada para cada palabra, calculando sus *probabilidades léxicas* teniendo en cuenta el contexto en el que aparecen.

Dada una frase genérica, $W = \{w_1, w_2, ..., w_T\}$ de talla T y $w_i \in V$ se define:

- *Probabilidad hacia adelante:* $\alpha_t(i) = P(W_{\leq t}; i_t = q_i)$, como la probabilidad de la secuencia $w_1, w_2, ..., w_t$ cuando en el instante i_t estamos en el estado q_i.
- *Probabilidad hacia atrás:* $\beta_t(i) = P(W_{>t}; i_t = q_i)$, como la probabilidad de la secuencia $w_{t+1}, w_{t+2}, ..., w_T$ cuando nos encontramos en el mismo instante i_t.

Estas probabilidades se pueden calcular mediante el algoritmo *Forward-Backward* o *Baum-Welch* utilizando las recursiones definidas en las ecuaciones (3.8) y (3.9)

respectivamente.

$$\alpha_t(1) = \pi_i \cdot b_i(w_1); \quad 1 \leq i \leq N$$
$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) \cdot a_{ij}\right] \cdot b_j(w_{t+1}); \quad 1 \leq j \leq N, t = 1, ..., T-1 \tag{3.8}$$

$$\beta_T(i) = 1$$
$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} \cdot b_j(w_{t+1}) \cdot \beta_{t+1}(j); \quad 1 \leq i \leq N, t = 1, ..., T-1 \tag{3.9}$$

La probabilidad de la frase W se obtiene utilizando la ecuación (3.10):

$$P(W) = \sum_{i=1}^{N} \alpha_T(i) \tag{3.10}$$

En (Merialdo, 1994) se utiliza esta aproximación. El proceso consiste en calcular $P(c_i|w_i)$ para cada categoría c_i , en el contexto anterior $w_1, ..., w_{i-1}$ mediante el algoritmo *Forward* (ecuación 3.11).

$$P(c_i|w_i; w_1, ..., w_{i-1}) = \frac{\alpha_{i(t)}}{\sum\limits_{j=1...N} \alpha_{j(t)}} \tag{3.11}$$

Para obtener la mejor etiqueta para la palabra w_i, se elegirá la de mayor probabilidad mediante la ecuación (3.12)

$$\widehat{c_i} = \arg\max_{c_i} \left(P(c_i|w_i; w_1, ..., w_{i-1})\right) \tag{3.12}$$

La principal diferencia entre las ecuaciones (3.7) y (3.12) es que, en (3.7) se maximiza sobre toda la frase, mientras que en (3.12) la maximización se plantea a nivel de la palabra que se está analizando. Para una cierta palabra w_i se maximiza sobre las posibles etiquetas (c_i) en el contexto de la frase. Este proceso se repite para todas las palabras de la frase con lo que se obtiene el mejor etiquetado de la misma.

También se puede considerar el contexto posterior de la palabra w_i, con lo que su probabilidad léxica se estimaría mediante el algoritmo *Forward-Backward* (ecuación

3.13):

$$P(c_i|w_i; w_1, ..., w_{i-1}, w_i, w_{i+1}, ..., w_T) = \frac{\alpha_{i(t)} \cdot \beta_{i(t)}}{\sum_{j=1...N} (\alpha_{j(t)} \cdot \beta_{j(t)})} \tag{3.13}$$

En la mayor parte de los trabajos sobre etiquetado que utilizan la aproximación de MM, por ejemplo, (Church, 1988), (DeRose, 1988), (Cutting et al., 1992), (Weischedel et al., 1993), (Demartas and Kokkinakis, 1995), etc., se usa el algoritmo de *Viterbi* como etiquetador debido principalmente a que su implementación es más sencilla. Además, la solución obtenida mediante este algoritmo proporciona una mejor interpretación desde el punto de vista lingüístico, ya que, en el proceso de etiquetado, ya que sólo se considera la secuencia de mayor probabilidad, en lugar de considerar todas las posibles secuencias, como se hace en el algoritmo *Forward-Backward*. Así, en este último caso, se tienen en cuenta un conjunto de secuencias de categorías que, generalmente, son lingüísticamente incorrectas.

Para el ejemplo de la figura 2, y para la palabra *"río"* en la etiqueta *"N"*, mediante *Viterbi* se elige un único camino, el de máxima probabilidad (probabilidad acumulada P_{ac} multiplicado por la probabilidad de transición) para llegar a este nodo: *máx{P(N/D)·P_{ac}(D), P(N/P)·P_{ac}(P), P(N/V)·P_{ac}(V), P(N/N)·P_{ac}(V), P(N/A)·P_{ac}(A)}*. Cuando se utiliza el algoritmo *Forward-Backward* se consideran todos los posibles caminos, se suman las probabilidades de todos los caminos que llegan a ese nodo, *sum{P(N/D)·P_{ac}(D), P(N/P)·P_{ac}(P), P(N/V)·P_{ac}(V), P(N/N)·P_{ac}(V), P(N/A)·P_{ac}(A)}*, con lo que se pueden consideran secuencias de categorías lingüísticamente incorrectas.

No obstante, parece que no se observan diferencias significativas a nivel de resultados de etiquetado léxico cuando se utiliza una u otra aproximación, por lo que en esta tesis, igual que en la mayoría de los trabajos publicados sobre el tema, utilizaremos la aproximación de Viterbi.

3.3.1 Algoritmo de Viterbi

La resolución de la ecuación (3.7) es un problema bien conocido de programación dinámica que se puede resolver fácilmente utilizando el mencionado algoritmo de *Viterbi*.

Suponiendo que el número de estados del modelo es N y que la talla de la frase de entrada es T, el coste computacional del proceso es $O(TN^2)$.

Para su resolución se define una función recursiva Φ, como

$$\begin{aligned}\Phi_1(q) &= P(q|q_0) \cdot P(w_1|q) \\ \Phi_{t+1}(q) &= \max_{q' \in Q}[\Phi_t(q') \cdot P(q|q')] \cdot P(w_{t+1}|q), \text{ para } 1 \leq t \leq T-1 \end{aligned} \tag{3.14}$$

Mediante esta recursión, en el instante T, el estado final (q_T) correspondiente al camino de máxima probabilidad viene dado por

$$q_T = \arg\max_{q \in Q} \Phi_T(q).$$

La secuencia de estados, correspondiente al camino de máxima probabilidad para una frase de entrada, se obtiene haciendo vuelta atrás sobre dicho camino mediante la expresión:

$$q_t = \arg\max_{q \in Q} \Phi_t(q) \cdot P(q_{t+1}|q), \text{ para } t = T-1, T-2, ..., 1$$

En la práctica, debido a que el valor de las probabilidades involucradas en los modelos (probabilidades de contexto y léxicas) toman valores muy pequeños, se pueden producir ciertas inestabilidades en el proceso de maximización. Para evitar problemas numéricos se puede plantear el algoritmo de Viterbi como un proceso de maximización de la función logaritmo de probabilidad, en lugar de la probabilidad. Con esta aproximación se define una nueva función Φ' como

$$\Phi'_{t+1}(q) = \max_{q' \in Q} \left[\Phi'_t(q') + \log(P(q|q')\right] + \log(P(w_{t+1}|q)), \text{ para } 1 \leq t \leq T-1$$

Para la implementación de este algoritmo, y con el fin de poder utilizar cualquier modelo de estados finitos para representar el modelo contextual, se ha utilizado una notación basada en autómatas de estados finitos estocástico.

Un autómata finito estocástico, $A_{EF} = (Q, \Sigma, \delta, q_0, q_F, D)$, se define como una 6-tupla en la que Q es un conjunto finito de estados. q_0 y q_F representan el estado inicial y final respectivamente. Σ es el conjunto de símbolos de entrada y $\delta : Qx\Sigma \rightarrow 2^Q$ es una función de transición entre estados a la que D asigna un conjunto de probabilidades.

Aunque un modelo de n-gramas suavizado se puede representar fácilmente como una matriz ($A = \{a_{ij}\}$), también se puede utilizar para su formalización la notación equivalente de autómatas de estados finitos, siendo esta última más general.

La relación entre las dos notaciones presentadas, para MM y para A_{EF}, es la siguiente: Q, representa en ambos casos el conjunto de estados, incluyendo los estados inicial y final (q_0, q_F), las categorías asociadas a los estados $C = \Sigma$, y la matriz $A = \{a_{ij}\}$ viene determinada por el conjunto de transiciones (δ) con sus correspondientes probabilidades definidas por la función D.

Las probabilidades léxicas definidas por $B = \{b_{jk}\}$, independientemente de la estructura elegida para el modelo contextual, muestran la relación entre los símbolos asociados a los estados (categorías) y el léxico.

Con esta representación se pretende utilizar MM en los que sea fácilmente intercambiable la estructura del mismo por cualquier modelo de estados finitos. En concreto, en el siguiente capítulo se sustituirá el modelo de bigramas, por autómatas finitos estocásticos aprendidos utilizando el algoritmo de inferencia gramatical ECGI.

La descripción del algoritmo de *Viterbi* que se ha utilizado en este trabajo se presenta en la figura 3.5. Con esta implementación se reduce el coste computacional del mismo, pasando de $O(TN^2)$ a $O(TN|Actv|)$, donde $|Actv|$ representa el número de estados activos en cada etapa de programación dinámica que siempre cumple que $|Actv| \ll N$. Estos estados activos vienen determinadas por las transiciones permitidas por el modelo contextual, teniendo en cuenta las restriciones impuestas por las probabilidades léxicas.

Entrada: $F = W_1, W_2, ..., W_T$
Datos:
Modelo Contextual
$\mathcal{A} = \{Q, \Sigma, \delta, q_0, q_F, D\}$, donde
$\Sigma = \mathcal{C} = \{C_1, ..., C_N\}$
Prob. de Contexto: $P(j|i) \subset D$
Probabilidades Léxicas
$P(W_j|j)$ $(P(W_j|C_j))$
INICIALIZACIÓN
$i : 1...NumEst; j : 0...NumPal;$
$Sucesores[i] \subset \delta$; /* conjunto de sucesores del estado i */
$Apunta[i][j]$; /* Matriz para guardar los mejores caminos */
$Anterior[i]$, $Actual[i]$; /* Vectores probabilidades acumuladas*/
$YaAlcanzado[i]$; /* Controla si un estado ha sido alcanzado en la etapa t */
$Activos1[i] = q_0, Activos2[i]$; $nactiv1 = 1, nactiv2 = 0$;
/* Est. activos en la etapa actual ($Activos1$) y siguiente ($Activos2$) */
ITERACIÓN
para t := **1** hasta $NumPal$ **hacer**
  $nactiv2 := 0$;
  para $i := 1$ hasta $NumEst$ **hacer** $Actual[i] = \infty$;**fin para**
  para i := **1** hasta $nactiv1$ **hacer**
    $est\_anterior := Activos1[i]$;
    para j ∈ **Sucesores[est_anterior] hacer**
      si $(P(W_t|C_{q_j}) \neq 0)$ **entonces**
        $Coste := Anterior[est\_anterior] + P(j|est\_anterior) + P(W_t|j)$;
        si $(YaAlcanzado[j] < t)$ **entonces**
          $Actual[j] := Coste; YaAlcanzado[j] := t$;
          $nactiv2 := nactiv2 + 1; Activos2[nactiv2] := j$;
          $apunta[j][t] := est\_anterior$;
        sino
          si $(Coste > Actual[j])$ **entonces**
            $Actual[j] := Coste$;
            $apunta[j][t] := est\_anterior$;
          finsi
        finsi
      finsi
    finpara
  finpara
  $Anterior := Actual; Activos1 := Activos2; nactiv1 := nactiv2$;
finpara
MEJOR SECUENCIA DE ESTADOS
$C(T) := i$ que maximice $Actual[j]$ en T
para $i := T - 1$ hasta 1 **hacer**
  $C[i] := apunta[C[i+1]][i+1]$
finpara

Figura 3.5: Algoritmo de Viterbi

3.4 Estimación de las Probabilidades de un MM

Para la resolución del problema planteado, previamente tendremos que calcular o estimar los parámetros que intervienen, es decir, las probabilidades léxicas y de contexto. Para ello se usan generalmente alguna de las alternativas siguientes: la estimación por máxima verosimilitud o basada en las frecuencias relativas (*Métodos Supervisados*) y la que se plantean la reestimación de los parámetros de un Modelo de Markov a partir de datos no etiquetados (*Métodos No Supervisados*).

3.4.1 Métodos Supervisados

Los parámetros del modelo en los métodos supervisados se pueden estimar por máxima verosimilitud (a partir de las frecuencias relativas) utilizando corpus ya etiquetados.

Las *probabilidades de contexto*, $P(c_i|c_{i-1})$, se obtienen mediante un sencillo cálculo consistente en contar cuantas veces aparece en el corpus la *categoría* c_{i-1} seguida de la *categoría* c_i y dividirlo por el número de veces que aparece la *categoría* c_{i-1}. El problema se plantea cuando en el corpus de aprendizaje no aparecen todos los pares de categorías posibles o cuando, aunque aparezcan, no lo hacen con una representación suficiente para que pueda tener relevancia estadística.

Para el cálculo de las *probabilidades léxicas*, $P(w_i|c_i)$, se obtiene la frecuencia de aparición de la palabra w_i en la categoría c_i y se divide por la frecuencia de aparición de la categoría c_i. El problema aparece, como en el caso anterior, cuando en el corpus de aprendizaje no encontramos la palabra w_i en la categoría c_i o bien no aparece dicha palabra w_i (problema de las *palabras desconocidas*[5]), con lo que no podremos calcular su probabilidad.

En ambos casos es necesario hacer uso de lo que se denomina *métodos de suavizado* para poder asignar probabilidades a los eventos no vistos o poco significativos. Este problema y sus soluciones se presentarán en la sección 3.5.

[5]Aquellas palabras que no han sido vistas en el proceso de aprendizaje.

3.4.2 Métodos No Supervisados

En esta aproximación los parámetros del MM se estiman a partir de corpora no etiquetados. Partiendo de un modelo inicial, el sistema toma frases de entrada y estas son consideradas como las emisiones del modelo siendo incapaces de ver la secuencia de estados asociada (puesto que no están etiquetadas).

Dada una frase W perteneciente al conjunto de entrenamiento, se define $\gamma_t(i)$ como la probabilidad de estar en el estado q_i en el instante t y γ_{ij} como el número de transiciones del estado q_i al estado q_j. La formalización de estos parámetros aparece en las ecuaciones (3.15) y (3.16).

$$\gamma_t(i) = P(i_t = q_i|W) = \frac{\alpha_t(i) \cdot \beta_t(i)}{P(W)} \tag{3.15}$$

$$\gamma_{ij} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) \cdot a_{ij} \cdot b_j(w_{t+1}) \cdot \beta_{t+1}(j)}{P(W)} \tag{3.16}$$

Dado un modelo inicial, $\mathcal{M}_I = (\Pi, A, B)$, la reestimación de sus parámetros se puede realizar mediante (3.17), (3.18) y (3.19).

$$\bar{\pi}_i = \gamma_1(i) \tag{3.17}$$

$$\overline{a}_{ij} = \frac{\gamma_{ij}}{\sum_{j=1}^{N} \gamma_{ij}} \tag{3.18}$$

$$\bar{b}_{jk} = \frac{\sum_{t=1, w_k=v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \tag{3.19}$$

El significado intuitivo de estas ecuaciones es el siguiente:

- $\bar{\pi}_i$ representa las veces que en el entrenamiento se ha transitado del estado inicial (q_0) al estado q_i, dividido por las veces que éste ha sido visitado.
- $\overline{a}_{ij}$ es el cociente entre el número de transiciones del estado q_i al estado q_j y el número total de transiciones que parten del estado q_i.

- $\bar{b}_{jk}$ es la relación entre las veces que se ha emitido el símbolo v_k en el estado q_j y el número de veces que se ha emitido un símbolo en el estado q_j.

Los métodos *No Supervisados* resultan muy atractivos ya que es más fácil disponer de grandes corpus de texto no etiquetado. Por otra parte, la elección de las categorías léxicas es más flexible (no es preciso utilizar las que aparecen en el corpus de aprendizaje) y en consecuencia, el cambio de categorías o incluso de una lengua a otra, es mucho más fácil. Sin embargo, el principal inconveniente es la estimación del modelo en si, ya que depende mucho de la inicialización elegida.

Una solución trivial para elegir el modelo inicial puede consistir en definir las probabilidades de contexto iniciales todas iguales y las probabilidades léxicas proporcionales a sus frecuencias de aparición en los corpora de aprendizaje. Esta solución, en general, no da muy buenos resultados y por eso, el modelo inicial se suele estimar a partir de un corpus de texto etiquetado, siguiendo el mismo proceso que en los métodos supervisados (Merialdo, 1994).

Otro aspecto a tener en cuenta es la estimación de las probabilidades léxicas. En aplicaciones reales y para tareas no restringidas, la talla del vocabulario utilizado puede ser de varios cientos de miles de palabras, lo cual dificulta la tarea de aprendizaje. Una solución que intenta paliar este problema la podemos encontrar en (Kupiec, 1992) y (Cutting et al., 1992), donde las palabras se sustituyen por clases de equivalencia de palabras. Si se dispone de un analizador morfológico, que para cada palabra proporcione sus posibles categorías léxicas, podemos sustituir cada palabra por su clase de ambigüedad, es decir, suponer que todas las palabras pertenecientes a una misma clase tienen la misma probabilidad. Con esta aproximación, se pasa de un vocabulario de palabras $V = \{v_1, v_2, ..., v_N\}$, a un nuevo vocabulario de clases de ambigüedad $\widehat{V} = \{A_1, A_2, ..., A_K\}$ (donde $K \ll N$) y las *Probabilidades Léxicas* $P(w_i = v_k|C_i)$ pasan a ser $P(w_i = A_k|C_i)$. Asumiendo estas simplificaciones en (Kupiec, 1992) se pasa de un vocabulario de 226,000 palabras a uno de 202 clases de ambigüedad, sin que las prestaciones del sistema se vean substancialmente afectadas. Una adaptación de esta técnica para construir un etiquetador léxico para el castellano se puede encontrar en (Sánchez and Nieto, 1995).

3.5 Métodos de Suavizado en N-gramas

Como se ha introducido anteriormente, los métodos de suavizado son necesarios para el cálculo de parámetros poco significativos o no contemplados en los corpora de aprendizaje. Cuando la estimación se hace siguiendo el criterio de máxima verosimilitud (probabilidades proporcionales a las frecuencias relativas de los sucesos), si la muestra de aprendizaje no es lo suficientemente rica (muestra insuficiente y/o no completa) se pueden obtener modelos que no describan de manera adecuada las dependencias de contexto o las probabilidades léxicas. En la literatura aparecen diversos métodos para solucionar estos problemas y han sido utilizados en múltiples aplicaciones, principalmente en el campo del reconocimiento automático de habla, para obtener modelos de lenguaje de diferentes niveles: unidades acústicas, palabras, categorías léxicas, categorías semánticas, etc. En este trabajo nos centraremos en los relacionados directamente con el problema del etiquetado léxico, es decir, suavizado de las probabilidades de contexto y léxicas.

Uno de los más sencillos es el conocido como *"Añadir 1"* consistente en incrementar todos los contadores de frecuencias de los sucesos S_i en 1 —*ley de Laplace*— o en una cierta cantidad k —*ley de Lidstone*— donde $(0 < k \leq 1)$.

$$\widehat{P}(S_i) = \frac{f(S_i) + k}{\sum_{\forall S_i}(f(S_i) + k)} \qquad 0 < k \leq 1 \tag{3.20}$$

El problema de esta aproximación es que se sobreestiman los sucesos con baja probabilidad, que probablemente, se corresponden con los de más baja frecuencia.

Otra alternativa es el *"Suavizado Plano"* consistente en reservar una cierta cantidad uniforme, $P^{\,reservada}$, que se descontará a los sucesos vistos, para repartirla entre los no vistos a todos por igual (ecuación 3.21).

$$\widehat{P}(S_i) = \begin{cases} P^{\,modificada}(S_i) & S_i \neq 0 \\ \dfrac{P^{\,reservada}}{\#S_i\ No\_Vistos} & S_i = 0 \end{cases} \tag{3.21}$$

Esto obliga a recalcular la probabilidad de los sucesos vistos para mantener la consistencia del modelo probabilístico, es decir para que se cumpla que

$$\sum_{\forall S_i : S_i \neq 0} P^{\ modificada} = 1 - P^{\ reservada}.$$

Estas aproximaciones las podemos utilizar para la estimación de las dos distribuciones de probabilidad que estamos estudiando (léxicas y de contexto). El inconveniente principal que presentan es que todos los sucesos no vistos se estiman con la misma probabilidad. Para solucionar este problema se suelen utilizar métodos que combinan diferentes distribuciones de probabilidad para dar cuenta de los sucesos no vistos. Estos métodos se pueden clasificar básicamente en dos grandes grupos: *Interpolación Lineal* y *Back-off*. El primero tiene en cuenta todas las distribuciones de probabilidad disponibles para la estimación de un determinado suceso. El segundo, sólo utiliza una, la que se espera que sea la más apropiada de entre las disponibles para ese suceso. Puesto que las distribuciones suavizadoras utilizadas dependen del tipo de suceso a estimar, vamos a hacer un estudio separado distinguiendo las que se usan para las probabilidades de contexto o Modelo de Lenguaje, y las usadas para las probabilidades léxicas o Modelo Léxico, particularizando para el caso de bigramas y refiriéndonos a sus aplicaciones en etiquetado léxico de textos.

3.5.1 Suavizado de las Probabilidades de Contexto

Vamos a plantear el problema sobre un modelo de bigramas; el caso mas general de *n-gramas*, se podría realizar de igual manera considerando las probabilidades $P(c_i|c_{i-n}...c_{i-1})$. El problema consiste en estimar $P(c_i|c_j)$ para cualquier par de categorías $(c_i, c_j) \in \mathcal{C} \times \mathcal{C}$. Supongamos que se dispone de dos distribuciones de probabilidad: *bigramas* ($P(c_i|c_j)$) y *unigramas* ($P(c_i)$) y que se desea combinarlas mediante *Interpolación Lineal* y *Back-off*.

La solución mediante *Interpolación Lineal* queda expresada en la ecuación (3.22), donde se ponderan las distribuciones de probabilidad de los bigramas y los unigramas mediante los parámetros de interpolación λ_i. Éstos pueden ser estimados experimentalmente o utilizando una variante del algoritmo *Forward-Backward* denominado *'deleted interpolation'* (Jelinek and Mercer, 1985; Jelinek, 1991). El número de parámetros de interpolación a estimar se reduce considerablemente si se supone que son independientes de la categoría considerada c_i. Bajo esta aproximación se

pueden calcular de una manera más sencilla como se propone, por ejemplo, en los trabajos de H. Ney (Ney and Kneser, 1991; Ney and Kneser, 1994).

$$P^{Int}(c_i|c_j) = \lambda_{i1} \cdot P(c_i|c_j) + \lambda_{i2} \cdot P(c_i); \quad \lambda_{i1} + \lambda_{i2} = 1 \tag{3.22}$$

La solución mediante *Back-off* consiste en descontar una cierta masa de probabilidad a los sucesos vistos para repartirla entre los no vistos, de manera proporcional a otra distribución de probabilidad. Se puede ver como un caso particular de interpolación lineal en el que para cada suceso, sólo un parámetro λ_i puede ser distinto de cero. Bajo esta asunción, podemos encontrar diversas aproximaciones dependiendo de la función de descuento utilizada y sobre qué sucesos se aplica.

En la ecuación (3.23) presentamos la aproximación introducida en (Katz, 1987), donde r es la frecuencia del suceso, k es un umbral (definido experimentalmente) a partir del cual aplicamos la función de descuento d_r y α es una constante de normalización para garantizar que $\sum_{\forall c_i} P(c_i|c_j) = 1, \forall c_j$

$$P^{\,Suavizada}(c_i|c_j) = \begin{cases} P(c_i|c_j) & r > 0,\ r > k \\ d_r \cdot P(c_i|c_j) & 0 < r \leq k \\ \alpha \cdot P(c_i) & r = 0 \end{cases} \tag{3.23}$$

La función de descuento *'Good Turing Discount'* (Good, 1953), utilizada por (Katz, 1987) (ecuación 3.24), se define en función de unos contadores especiales (n_x: numero de veces que un suceso aparece con frecuencia x) que se obtienen a partir del corpus de aprendizaje. Esta función garantiza que el descuento total aplicado sea igual a $\frac{n_1}{R}$, donde n_1 representa el número de sucesos de frecuencia 1 y R el número total de muestras de aprendizaje.

$$d_r = \frac{\frac{(r+1)n_{r+1}}{rn_r} - \frac{(k+1)n_{k+1}}{n_1}}{1 - \frac{(k+1)n_{k+1}}{n_1}} \tag{3.24}$$

El principal inconveniente de esta aproximación es la determinación experimental del umbral k y el cálculo de d_r, puesto que para algunos casos, la ecuación puede dar valores indefinidos.

En la literatura aparecen otras alternativas que no tienen en cuenta el umbral k (expresión 3.25) y que garantizan que $d_r > 0$ para todo r, cumpliéndose además, que el descuento total aplicado es de $\frac{n_1}{R}$.

$$P^{\,Suavizada}(c_i|c_j) = \begin{cases} d_r \cdot P(c_i|c_j) & r > 0 \\ \alpha \cdot P(c_i) & r = 0 \end{cases} \tag{3.25}$$

El *descuento lineal* (Jelinek, 1991) aplica el mismo descuento a todos los sucesos independiente de su frecuencia (ecuación 3.26)

$$d_r = \frac{n_1}{R} \tag{3.26}$$

El *descuento absoluto* (Ney and Kneser, 1991; Ney and Kneser, 1994) reduce en una constante b las probabilidades de todos los sucesos (3.27).

$$d_r = \frac{r - b}{R} \text{ donde } b = \frac{n_1}{n1+2n_2} \tag{3.27}$$

Una revisión de todos estos métodos, así como su contrastación experimental sobre el corpus en inglés *Wall Street Journal*, se puede encontrar en (Young and Bloothooft (editors), 1997). En (Church, 1988) (DeRose, 1988) (Maltese and Mancini, 1991) (Kupiec, 1992) (Cutting et al., 1992) (Church and Gale, 1991) (Merialdo, 1994) (Weischedel et al., 1993) (Demartas and Kokkinakis, 1995) podemos encontrar su aplicación (en algunos casos con pequeñas variaciones) al problema de etiquetado léxico.

3.5.2 Suavizado de las Probabilidades Léxicas

La estimación de las probabilidades léxicas conlleva una gran dificultad para las palabras de baja frecuencia y sobretodo, para las palabras desconocidas (aquellas que no aparecen en el conjunto de entrenamiento). Cuando aprendemos el modelo de lenguaje contextual, junto con el modelo léxico, se hace con un cierto corpus y éste, generalmente, no cubre todo el espectro léxico de la aplicación. Este hecho hace que ante una frase nueva aparezcan palabras no contempladas en el entrenamiento. El problema se agrava aún más cuando se cambia el dominio de la aplicación y se pretende reutilizar los mismos modelos. En este caso, el léxico puede diferir substancialmente del aprendido inicialmente.

Las aproximaciones utilizadas para el suavizado de las probabilidades léxicas en MM siguen básicamente dos tendencias, que en algunos casos se complementan:

(a) modelar el comportamiento de las palabras desconocidas en cada categoría definiendo una distribución de probabilidad para las mismas.

(b) utilizar información morfológica para determinar la pertenencia de una palabra a una determinada categoría.

En el primer grupo cabe destacar la aproximación presentada en (Demartas and Kokkinakis, 1995). En este trabajo, se realiza un estudio experimental en el que se concluye que *"la distribución de probabilidad de las palabras desconocidas es muy similar al de aquellas palabras que aparecen con frecuencia* 1 *y además, es muy diferente a la distribución de las palabras conocidas"*. Basándose en esta hipótesis, la estimación de estas probabilidades se hace mediante la ecuación (3.28).

$$\begin{aligned} P(w^{\,desconocida}|c_i) &= \frac{P(c_i|w^{\,desconocida}) \cdot P(w^{\,desconocida})}{P(c_i)} \\ &\approx \frac{P(c_i|w^{\,menos\_probable}) \cdot P(w^{\,desconocida})}{P(c_i)} \end{aligned} \tag{3.28}$$

$P(c_i|w^{\,menos\_probable})$ y $P(c_i)$ se calculan a partir del conjunto de entrenamiento utilizado. $P(w^{\,desconocida})$ se estima a partir de textos abiertos, distintos al de entrenamiento, observando las palabras que están fuera del léxico aprendido y viendo a qué categoría pertenecen. Finalmente, todas las probabilidades se normalizan para garantizar la consistencia estocástica del modelo (ecuación 3.29), donde M representa la talla del vocabulario y N el número de categorías léxicas.

$$\sum_{j=1}^{M} P(w_j^{\,conocida}|c_i) + P(w^{\,desconocida}|c_i) = 1 \qquad \forall i = 1...N \tag{3.29}$$

En (Merialdo, 1994) se define una función de distribución de probabilidad uniforme, para cada categoría c_i, que cumple que es inversamente proporcional al número de palabras en cada categoría $T(c_i)$ (ecuación 3.30).

Esta distribución se interpola linealmente con la de las palabras conocidas mediante la ecuación (3.31). Los parámetros de interpolación se pueden estimar experimentalmente o bien utilizando alguna de las técnicas comentadas para el cálculo

de las probabilidades de contexto.

$$P^{\ uniforme}\ (w_j|c_i) = \frac{1}{T(c_i)} \tag{3.30}$$

$$P^{\ Int}(w_j|c_i) = \lambda \cdot P\ (w_j|c_i) + (1-\lambda) \cdot P^{\ uniforme}\ (w_j|c_i) \tag{3.31}$$

En el segundo grupo se tiene en cuenta información morfológica de las palabras. Cabe destacar, por ser una de las primeras, la propuesta de (Weischedel et al., 1993), donde se tiene en cuenta, para el inglés, ciertos finales característicos (*-ing, -ed, -s, -ion, -ly, -able, ...*), palabras en mayúscula, presencia de guiones en las mismas, etc. A partir de un conjunto de entrenamiento supervisado, se estiman las probabilidades de estas características para cada categoría y posteriormente, se combinan para una palabra desconocida (w^{UNK}), en concreto de la siguiente manera:

$$P(w^{UNK}|c_i) = P(W_{unknown}|c_i) \cdot P(W_{Capital}|c_i) \cdot P(W_{ending}|c_i)$$

En trabajos posteriores, también para el inglés, se han utilizado soluciones similares. Por ejemplo, en (Franz, 1997) se combinan un conjunto de características morfológicas mayor, produciendo resultados mejores a nivel de etiquetado para las palabras desconocidas. Para otras lenguas, como por ejemplo el castellano, en el que la problemática de las palabras desconocidas es mucho más compleja, se suelen utilizar analizadores morfológicos. Éstos, utilizan grandes diccionarios, lematizadores, etc, y proporcionan para cada palabra sus posibles etiquetas léxicas. A partir de esta información, se asigna una probabilidad de pertenencia a la categoría. Esta es la solución que se ha adoptado en este trabajo para el castellano y que se describirá con más detalle en el capítulo 5.

3.6 Modelos Contextuales Especializados

Para enriquecer la modelización contextual se han presentado diferentes aproximaciones en la literatura con el objetivo de extender el contexto considerado. Así, el formalismo de MM de primer orden (bigramas) se amplía a ordenes superiores (trigramas, cuatrigramas, ...) para obtener modelizaciones más consistentes. Esta ampliación, al considerar una historia mayor, define un número de parámetros

muy elevado y presenta inconvenientes en la estimación de dichos parámetros. Los modelos ECGI, que se definirán en el siguiente capítulo, también constituyen un ejemplo de extensión del contexto y, en consecuencia, también presentan este mismo problema.

Para optimizar la modelización contextual en problemas de etiquetado léxico y análisis superficial, se pueden seguir, entre otras, básicamente dos tendencias: 1) considerar dependencias de mayor longitud, sólo en ciertos casos preestablecidos, con el fin de mantener tamaños de modelos aceptables y 2) hacer intervenir las palabras en el modelo contextual para establecer nuevas restricciones estructurales que describan relaciones entre las palabras y sus categorías.

Así por ejemplo, en (Brants, 1996) se considera un MM de primer orden que es ampliado, duplicando ciertos estados y fusionando otros, con el fin de poder tener en cuenta mayor información de sus predecesores. Otros autores hacen intervenir las palabras en los modelos contextuales, como en (Kim et al., 1999) (sólo para un cierto número de palabras) o en (Lee et al., 2000) utilizando MM totalmente lexicalizados para el problema de etiquetado léxico, con el consiguiente aumento del tamaño del modelo y la aparición de nuevos problemas asociados a la estimación de sus parámetros.

Dentro del campo de la inferencia gramatical encontramos algunas aproximaciones (incluida la aproximación basada en modelos ECGI que se presentará en el siguiente capítulo) que adoptan ciertos criterios similares para establecer restricciones estructurales más complejas.

La metodología MGGI –*Morphic Generator Grammatical Inference*– (García et al., 1987), también utilizada en (Segarra, 1993), define una función de etiquetado, de acuerdo a un cierto criterio definido por un experto, que produce un reetiquetado del conjunto de muestras de aprendizaje y en consecuencia, un cambio en el alfabeto ($\Sigma \rightarrow \Sigma'$). Con esta función se permite distinguir ciertos símbolos, dependiendo por ejemplo de la posición en la que se encuentren en una determinada cadena, con lo que se consiguen modelizaciones contextuales (bigramas) más complejas. Posteriormente, cuando los autómatas construidos son utilizados en tareas de reconocimiento, se debe definir un homomorfismo o función inversa a la de etiquetado, que deshaga el etiquetado inicial ya que la entrada está formada por símbolos pertenecientes a Σ y el autómata se ha aprendido a partir del alfabeto Σ'.

A continuación se presenta una técnica que permite enriquecer los modelos contextuales presentados (y en general, cualquier modelo regular), con la incorporación de ciertas palabras al mismo, además de las categorías léxicas, y así poder establecer ciertas restricciones de contexto ligadas al léxico.

La especialización de ciertas palabras en sus categorías léxicas, elegidas teniendo en cuenta criterios lingüísticos o de manera automática a partir del conjunto de entrenamiento, redunda en una mejor modelización contextual como se mostrará en los experimentos realizados en los siguientes capítulos. Algunos de los criterios establecidos para elegir el conjunto de palabras objeto de la especialización son los siguientes: las palabras más frecuentes, las palabras con mayor error de etiquetado, las palabras pertenecientes a categorías cerradas, etc. Estos conjunto permiten, en cierto modo, incorporar conocimiento lingüístico a los modelos, ya que pueden ser definidos siguiendo criterios lingüísticos y así poder capturar restricciones estructurales que no se pueden establecer considerando solamente las categorías léxicas.

3.6.1 Formulación del Proceso de Especialización

Sean los conjuntos

- $\Sigma = \{C_1, ..., C_N\}$: conjunto de categorías léxicas.
- $V = \{w_1, ..., w_M\}$: conjunto de palabras o vocabulario de la aplicación.
- $E \subset (V \times \Sigma)^*$: conjunto de aprendizaje.

A partir del conjunto de entrenamiento E, los modelos MM se aprenden considerando frases de pares de palabra y etiqueta ($< w_1, C_1 >, ..., < w_M, C_M >$). Para ello, se utiliza alguna de las técnicas presentadas en este capítulo.

Para construir el modelo especializado se define un conjunto, $\mathcal{W}_e \subset V$, formado por las palabras que se considerarán en el modelo contextual y una función de especialización f_e que determina un nuevo conjunto de entrenamiento.

La función f_e se define sobre el conjunto de entrenamiento original (E) y proporciona un nuevo conjunto de entrenamiento ($\hat{E}$), sobre el que se pueda aprender el modelo especializado. Esta especialización hace que, el conjunto de categorías

original Σ, se incremente con las palabras consideradas en $\mathcal{W}_e$, especializadas en sus diferentes categorías léxicas. El nuevo conjunto $\hat{\Sigma} \subset ((\mathcal{W}_e \cup \lambda) \times \Sigma)$ queda determinado aplicando una función f_e, definida de la siguiente manera:

$$f_e : E \subset (V \times \Sigma)^* \rightarrow \hat{E} \subset \left(V \times \hat{\Sigma}\right)^*$$

$$f_e(< w_i, C_i >) = \begin{cases} < w_i, (w_i, C_i) > & \text{si } w_i \in \mathcal{W}_e \\ < w_i, (\lambda, C_i) > & \text{si } w_i \notin \mathcal{W}_e \end{cases}$$

La función f_e realiza un reetiquetado del conjunto de entrenamiento original (E) consistente en sustituir la etiqueta C_i de la palabra w_i, por la nueva etiqueta (w_i, C_i) si w_i pertenece al conjunto de palabras a especializar o bien, se conserva la etiqueta existente, si no pertenece a dicho conjunto (λ, C_i), donde en este caso λ representa la cadena vacía.

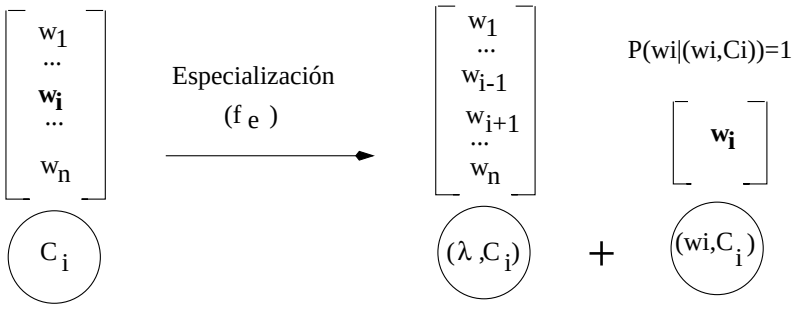

Figura 3.6: Proceso de especialización de una palabra w_i en la cateogoría C_i.

En la figura 3.6 se puede observar el efecto producido sobre un estado del modelo contextual tras aplicar la función de especialización f_e sobre la palabra $w_i \in \mathcal{W}_e$ en la categoría C_i. En el nuevo modelo contextual aparecerá un nuevo estado por cada palabra especializada, en cada una de sus categorías léxicas posibles. Además, en los estados especializados, sólo se puede emitir la palabra asociada, por lo que la probabilidad léxica para la misma debe valer 1.

Una vez construido el modelo especializado, para realizar el proceso de etiquetado se sigue el mismo método (algoritmo de Viterbi) presentado en la sección 3.3.1. Así, se obtiene la secuencia de estados del modelo de máxima probabilidad para un frase de entrada. Una vez obtenida dicha secuencia, como lo que se busca es la secuencia

de categorías, es necesario aplicar una nueva función que deshaga la especialización construida. Esta función, que denotaremos por f_d, deberá proporcionar, a partir de las etiquetas consideradas en el modelo especializado, el conjunto de categorías léxicas (Σ) que serán asignadas finalmente a las palabras de la frase de entrada.

$$f_d : \hat{\Sigma} \rightarrow \Sigma$$

$$f_d\left((w_i, C_i)\right) = C_i \quad \text{donde} \quad w_i \in (\mathcal{W}_e \cup \lambda)$$

3.7 Resumen

En este capítulo se ha presentado la formalización del problema de etiquetado léxico de textos desde el punto de vista estadístico. Los modelos de Markov o n-gramas son el modelo matemático más extendido en los principales trabajos dentro de este paradigma. Se ha formalizado el algoritmo de Viterbi para poder ser utilizado como método de obtención de la mejor secuencia de estados en un modelo de estados finitos de cualquier tipo, que incluyen desde n–gramas, hasta cualquier modelo regular. Como se verá en el siguiente capítulo, el algoritmo es directamente aplicable para los modelos ECGI.

Además, se han presentado las principales técnicas de suavizado utilizadas en la estimación de las probabilidades de contexto y léxicas en un MM. Algunas de estas técnicas se utilizarán en el sistema de etiquetado propuesto en esta tesis y otras serán modificadas para aplicarlas a los citados modelos ECGI.

Finalmente se propone, lo que hemos llamado *Modelos Contextuales Especializados* con el fin de hacer intervenir ciertas palabras, junto con las categorías, en los modelos contextuales. Está lexicalización de los modelos, aunque aumenta el tamaño de los mismos, redunda en una mejor modelización contextual como se mostrará experimentalmente es los capítulos siguientes.

Capítulo 4

Aprendizaje de Modelos Contextuales mediante Inferencia Gramatical

4.1 Introducción

En la aproximación estadística al etiquetado de textos planteada se hace necesario el uso de un modelo contextual o modelo de lenguaje que dé cuenta de las posibles o probables secuencias de categorías léxicas de un determinado lenguaje, es decir, que restrinja las posibles concatenaciones de las unidades lingüísticas consideradas.

El formalismo de n-gramas, como se ha mostrado en el capítulo anterior, es el más extendido para la modelización contextual de secuencias de categorías léxicas así como de otras unidades lingüísticas. Las razones que justifican este hecho son varias. En primer lugar, permite lo que se conoce como aprendizaje basado en corpus; es decir, una vez fijado n (longitud del contexto a considerar), los modelos se aprenden automáticamente a partir de un conjunto de datos de la aplicación. En segundo lugar, permiten una fácil implementación ya que existen algoritmos eficientes para su tratamiento.

Sin embargo, en general, presentan el inconveniente de no reflejar adecuadamente la estructura completa de la frase, lo cual puede redundar en una deficiente

modelización de las relaciones a larga distancia entre términos.

Existen otras aproximaciones, que podríamos llamar gramaticales, en las que se modeliza la estructura de la frase haciendo uso de gramáticas formales, principalmente gramáticas regulares y gramáticas incontextuales, que son capaces de capturar mejor la estructura del lenguaje.

En este capítulo proponemos la utilización de técnicas de Inferencia Gramatical (Fu and Booth, 1975), para abordar el problema de la definición del lenguaje desde un formalismo gramatical (en concreto, gramáticas regulares) que incorpora el aprendizaje basado en datos. Desde este punto de vista, pretendemos aglutinar las siguientes ventajas:

- Aprendizaje automático a partir de un conjunto de datos, característica de los modelos basados en n-gramas.
- Flexibilidad, es decir, tolerancia a construcciones lingüísticas no estrictamente correctas, pero aceptables, mediante la introducción de técnicas de suavizado.
- Representación natural de las restricciones del lenguaje, es decir, de su estructura global, característica de los modelos basados en gramáticas.

En el marco de la Inferencia Gramatical, diferentes métodos son potencialmente aplicables para el aprendizaje de Modelos de Lenguaje. Para un estudio más en profundidad de los fundamentos y diversas aplicaciones de la Inferencia Gramatical, se puede consultar el libro de González y Thomason (González and Thomason, 1978), el libro de Reconocimiento Sintáctico de Formas de Fu (Fu and Booth, 1982), la revisión sobre Inferencia Gramatical de Fu y Booth (Fu and Booth, 1975), el artículo de Angluin y Smith (Angluin and Smith, 1983), el de Miclet (Miclet, 1990), y el de Vidal, García y Casacuberta (Vidal et al., 1993).

A continuación realizaremos una breve revisión de los métodos de inferencia gramatical que han sido aplicados a la modelización del lenguaje:

1. Método de inferencia de lenguajes k-testables en sentido estricto (García and Vidal, 1990). Es la aproximación de n-gramas desde el punto de vista de la Inferencia gramatical (Segarra, 1993). Una muestra de su aplicación al aprendizaje de modelos

de lenguaje, con diferentes métodos de suavizado, para la base de datos BDGEO, la tenemos en (Bordel, 1993; Bordel, 1994).

2. Metodología de inferencia gramatical basada en generadores mórficos (García et al., 1987). Se trata de una metodología general de inferencia gramatical que supone un compromiso entre los métodos heurísticos y caracterizables, ya que identifica una clase específica de lenguajes (los lenguajes locales), pero permite incorporar el conocimiento a priori sobre el problema particular que se intenta abordar. Una aplicación de esta metodología al problema del aprendizaje automático de modelos de lenguaje semánticos se presenta en (Segarra, 1993).

3. Método de inferencia de lenguajes k-reversibles (Angluin, 1982). Este método se basa en la técnica de agrupación de estados y permite inferir lenguajes regulares a partir de muestras positivas. Una aplicación para el aprendizaje de subconjuntos pequeños del inglés la encontramos en (Berwick and Pilato, 1987).

4. Método de inferencia de lenguajes regulares utilizando muestras positivas y negativas (Oncina, 1991). Otro método, también basado en la agrupación de estados, que ha sido aplicado para el aprendizaje de lenguajes, con un comportamiento muy adecuado con respecto a otras aproximaciones (Oncina, 1991; Oncina and García, 1992; Oncina et al., 1993).

5.- Método de inferencia de gramáticas incontextuales Inside-Outside (Baker, 1979) (Lari and Young, 1991). Es una técnica de estimación de las probabilidades de las reglas incontextuales a partir de muestras positivas. Se ha utilizado para el aprendizaje de modelos de lenguaje para un subconjunto pequeño del corpus ATIS (Pereira and Schabes, 1992). Recientemente en (Benedí and Sánchez, 2000) se presenta un modelo híbrido en el que se combinan modelos de n-gramas con gramáticas incontextuales.

En este trabajo estudiaremos la aplicación del algoritmo de Inferencia Gramatical basado en Corrección de Errores (ECGI del inglés *Error Correcting Grammatical Inference*) (Rulot and Vidal, 1987; Rulot et al., 1989; Rulot, 1992), (Vidal et al., 1988) al problema de la modelización de categorías léxicas y sintácticas, para resolver el problema del etiquetado léxico de textos (*POS tagging*) y el análisis sintáctico superficial (*Shalow Parsing*).

4.2 Algoritmo ECGI

El algoritmo de Inferencia Gramatical basado en Análisis Corrector de Errores (*ECGI*) es un heurístico que construye una gramática regular (o el equivalente autómata de estados finitos) de una forma incremental a partir de un conjunto de muestras positivas, consideradas una detrás de otra. Como tal heurístico, incorpora directamente cierto conocimiento sobre el dominio de la aplicación en el proceso de inferencia. En particular, este proceso incide especialmente en la consecución de cierta capacidad de abstracción para capturar la variabilidad relevante que presentan las subestructuras locales de la muestra de aprendizaje en función de sus posiciones en las mismas, sus duraciones y sus concatenaciones.

Inicialmente se construye un autómata (o gramática regular) trivial que sólo reconoce (genera) la primera cadena del conjunto de muestras. A continuación, para cada nueva cadena de la muestra que no pertenece al lenguaje reconocido por el autómata obtenido hasta ese momento, se actualiza dicho autómata añadiendo aquellos estados y transiciones que sean necesarios para que la nueva cadena sea aceptada por el autómata. Con el fin de determinar dichos estados y transiciones, se incorpora un esquema de corrección de errores estándar (inserción, sustitución y borrado) y se utiliza un procedimiento basado en Programación Dinámica, similar al algoritmo de Viterbi (Fourney, 1973), para encontrar el mejor alineamiento entre la cadena de entrada y la cadena más próxima en el lenguaje reconocido por el autómata actual. El resultado de esta fase de análisis sintáctico con corrección de errores se utiliza para modificar el autómata aprovechando al máximo la estructura actual; así pues, sólo las transiciones de error (o secuencias de transiciones de error), conducen a la adición de nuevos estados y transiciones. Este mecanismo de construcción incremental es tal que conduce a la obtención de autómatas sin ciclos, en los que cada estado tiene asignada una etiqueta (terminal). De esta forma, los lenguajes reconocidos por ellos suponen una "generalización conservadora" de la muestra de aprendizaje.

En la figura 4.1 se muestra un ejemplo del proceso de construcción de un autómata ECGI. A partir de la primera muestra de aprendizaje (a), compuesta por secuencias de categorías léxicas del conjunto PAROLE (véase apéndice A), se obtiene un autómata inicial que sólo reconoce esa muestra. Este autómata trivial es

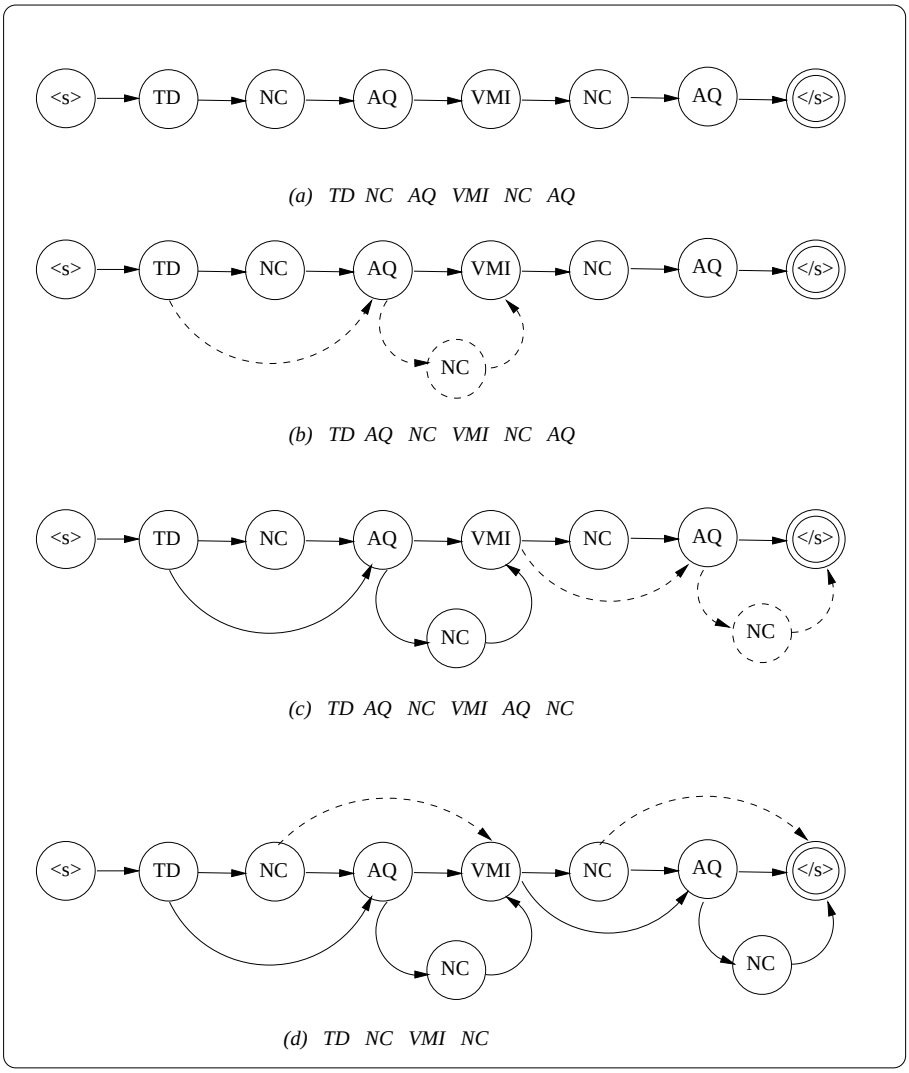

Figura 4.1: Ejemplo de construcción de un modelo de categorías léxicas mediante el algoritmo ECGI.

modificado, añadiendo nuevos estados y transiciones, para reconocer las sucesivas muestras (b), (c), (d). El autómata construido, a parte de aceptar el conjunto de muestras de aprendizaje, produce una generalización del mismo, por lo que también reconoce secuencias de categorías no vistas en el aprendizaje, como por ejemplo: *TD AQ VMI NC AQ*, *TD NC VMI AQ*, *TD NC AQ VMI AQ*, etc.

4.2.1 Descripción y Propiedades del Algoritmo ECGI

El algoritmo ECGI se describe formalmente en la figura 4.2 (Sanchis, 1994). El núcleo principal del mismo es un proceso iterativo sobre el conjunto de datos R^+constituido por dos acciones fundamentales:

- Análisis: proceso de análisis sintáctico con corrección de errores de la cadena. La derivación obtenida incluye reglas de error y de no error, por lo que cada $a_i^{'}$ será o un símbolo de la cadena o bien el símbolo nulo.
- Construcción: actualización del autómata a partir de la información obtenida en la fase anterior.

Las gramáticas obtenidas por el algoritmo ECGI constituyen descripciones estructurales de la muestra de aprendizaje, generalmente muy adecuadas, y de hecho pueden ser utilizadas como modelo de lenguaje de las mismas. Además, estas gramáticas pueden ser ampliadas con información estadística.

Las propiedades de las gramáticas inferidas con este método se estudian en profundidad en (Rulot, 1992). A continuación citaremos algunas de las más relevantes:

- Son no deterministas y generalmente ambiguas.
- Dadas las características del método de construcción, es obvio que los lenguajes generados por estas gramáticas contienen a la muestra de aprendizaje R^+, característica coherente con la propiedad de consistencia de los métodos constructivos de inferencia.

Algoritmo ECGI

Datos: $R^+ = \{\alpha_0, \alpha_1, .., \alpha_n\}$;

Resultado: $G_n = (S, V, N_n, P_n)$;

Inicialización: /* se obtiene la gramática canónica de $\alpha_0 = a_1, a_2, .., a_m$ */

$N_0 = \{A_0, A_1, .., A_m\}$ $S = A_0$; $F = A_m$;

$P_0 = \{A_0 \rightarrow a_1 A_1, A_0 \rightarrow a_2 A_2, .., A_{m-1} \rightarrow a_m A_m\}$;

Iteración:

$\forall k = 1...n$ hacer /* $\alpha_k = a_1, a_2, .., a_T$ */

ANÁLISIS: /* Obtener una derivación óptima con corrección de errores de α_k */

$d^*(\alpha_k|G) \equiv (C_0 \rightarrow a_1^{'} C_1), (C_1 \rightarrow a_2^{'} C_2), ..., (C_{T-1}^{'} \rightarrow a_T^{'} C_T)$

CONSTRUCCIÓN:

$\forall$ subsecuencia $(C_{i-1} \rightarrow a_i^{'} C_i), (\mathbf{C}_i \rightarrow \mathbf{a}_{i+1}^{'} \mathbf{C}_{i+1}), .., (\mathbf{C}_{j-1} \rightarrow \mathbf{a}_j^{'} \mathbf{C}_j), (C_j \rightarrow a_{j+1}^{'} C_{j+1})$
de reglas de error (en negrita), comprendidas entre las dos de no error
$(C_{i-1} \rightarrow a_i^{'} C_i), (C_j \rightarrow a_{j+1}^{'} C_{j+1})$,
Sea $w = b_1 b_2, .., b_l$ la secuencia $a_{i+1}^{'} a_{i+2}^{'}, ..., a_j^{'}$ sin símbolos nulos e.
Añadir:

si $w = \lambda$ entonces /* si es la cadena vacía */

$P_k = P_{k-1} \cup \{(C_i \rightarrow a_{j+1}^{'} C_{j+1})\}$/* Añadir una transición (regla)*/

sino /* Añadir nuevos estados (no terminales) y transiciones (reglas) */

$N_k = N_{k-1} \cup \{C_1^{'}, C_2^{'}, ..., C_l^{'}\}$

$P_k = P_{k-1} \cup \{(C_i \rightarrow b_1 C_1^{'}), (C_1^{'} \rightarrow b_2 C_2^{'}), .., (C_{1-1}^{'} \rightarrow b_l C_l^{'}), (C_1^{'} \rightarrow b_{j+1} C_{j+1}^{'})\}$

finsi

finpara

finpara

fin ECGI

Figura 4.2: Algoritmo ECGI.

- Las gramáticas no presentan ciclos, por lo que los lenguajes que se generan son finitos. Esta propiedad proviene del propio mecanismo de construcción. Obsérvese que no se generan bucles en los estados, ya que los errores de inserción suponen la creación de nuevos estados. Además, el modelo de error utilizado hace que la derivación de corrección de errores corresponda a un camino en la gramática extendida que utiliza los estados correspondientes a un sólo camino de la gramática generada hasta el momento (aunque utilice reglas de error), y por lo tanto, las nuevas reglas añadidas (secuencias de estados y transiciones) comienzan y terminan en estados de este único camino, lo que impide que se generen bucles. Por otra parte, se observa que en aplicaciones de Reconocimiento Sintáctico de Formas (Rulot et al., 1989), aunque la talla del lenguaje que se obtiene tiende a crecer exponencialmente con el tamaño del conjunto de entrada, lo que indica que se produce una generalización, el número de reglas y no terminales (estados) tiende a mantenerse constante a partir de un determinado número de muestras. Esto se debe a que la gramática consigue capturar la variabilidad de las cadenas de entrada, de modo que cuando se han utilizado suficiente número de muestras, el extralenguaje generado contiene, además de las cadenas de aprendizaje, un gran número de cadenas de similares características. Por tanto las nuevas cadenas que se van observando, o ya pertenecen al lenguaje inferido, o sólo requieren de un pequeño número de reglas de error para ser generadas.

- En general, las gramáticas resultantes de la aplicación de este algoritmo, dependen del criterio de presentación de la muestra de aprendizaje; es decir, del orden de presentación. No obstante, se observa que los efectos de esta dependencia son menos significativas si el número de muestras de aprendizaje es suficientemente elevado (Prieto, 1995).

A la vista de estas propiedades y del método constructivo puede destacarse que las gramáticas obtenidas son capaces de describir las diferentes longitudes de las subestructuras que forman los objetos, así como su variabilidad estructural, características que aparecen reflejadas en los extralenguajes que se generan. También se puede observar que las gramáticas representan la variabilidad estructural de las subestructuras en función de su posición relativa en la muestra, de forma que aunque aparezca la misma subestructura repetida en las cadena, éstas generan secuencias

de estados y transiciones en la posición en que aparecen, sin utilizar subsecuencias análogas ya existentes pero en posiciones distintas.

Las gramáticas ECGI pueden ser ampliadas con información estadística referente a las probabilidades de utilización de sus reglas. En concreto, las probabilidades de las reglas de error y de no error pueden ser aproximadas a partir de su frecuencia de utilización durante la fase de análisis del conjunto de muestras de aprendizaje. Ver detalles en (Prieto, 1988), (Rulot, 1992) y (Prieto, 1995).

Las principales dificultades que plantea el aprendizaje de las probabilidades de las reglas de error, es el gran número de éstas que hay que estimar, lo que exigiría un número prohibitivo de muestras. Para evitar estos problemas, se establece una "ligadura" entre las reglas de error, de modo que sus probabilidades no dependan de los no terminales asociados a ellas (es decir, que no dependan de la posición del error en la cadena), sino sólo del tipo de error (sustitución, inserción, borrado). Además, la probabilidad de inserción de un símbolo se considera independiente del símbolo que vaya a continuación. De este modo, el número de probabilidades a estimar se reduce considerablemente.

Una simplificación que también suele introducirse en el modelo de error consiste en definir una gramática expandida sólo con errores de sustitución. Esto facilita la estimación de las probabilidades, y mejora la complejidad computacional de los algoritmos de reconocimiento, dado que el número de reglas es menor. Su viabilidad, comprobada experimentalmente en tareas de Reconocimiento de Palabras Aisladas (Rulot, 1992), aunque con resultados ligeramente inferiores, viene dada por el hecho de que la gramática inferida puede aceptar cadenas de menor y mayor longitud que las analizadas en el aprendizaje, de modo que usando sólo sustituciones se pueden encontrar derivaciones que generen las cadenas que se quieren reconocer. A pesar de está simplificación, el uso de los errores de sustitución como mecanismo de suavizado, no es suficiente para garantizar una adecuada cobertura del lenguaje y se deben combinar con otras funciones de distribución de probabilidad. Un ejemplo se puede encontrar en (Prieto, 1995) donde se combina, mediante interpolación lineal, la matriz de errores de sustitución con la función de distancia lexicográfica entre cadenas.

Debido a esta dificultad, en esta tesis se proponen diferentes técnicas de suavizado, inspiradas en las presentadas en el capítulo 3 para modelos de n-gramas, con

el fin de extender los modelos ECGI y así garantizar la cobertura total del lenguaje usado. Así, los modelos ECGI podrán sustituir al modelo contextual probabilístico utilizado en un MM y ser usado para resolver los problemas de etiquetado léxico de textos y análisis sintáctico superficial abordados en esta tesis.

4.3 Modelos ECGI Extendidos (ECGIE)

Como ya se ha comentado, el modelo contextual inferido mediante el algoritmo ECGI es una gramática regular estocástica o su equivalente autómata finito estocástico (no determinista y ambiguo).

Los autómatas inferidos, $A_{EF} = (Q, \Sigma, \delta, q_0, q_F, D)$, no aseguran una adecuada cobertura del lenguaje que se pretende analizar. Diferentes ideas de tipo práctico, inspiradas en los modelos para n-gramas, se han incorporado a éstos con el fin de conseguir coberturas aceptables y estimaciones fiables.

El método propuesto (Pla, 1999) consiste en una extensión del autómata original, modificando el conjunto de transiciones y el de probabilidades de transición, de tal manera que se garantice que se pueda transitar con cualquier símbolo desde cualquier estado. Así pues, dado un autómata inicial, inferido con el algoritmo ECGI, $A_{EF} = (Q, \Sigma, \delta, q_0, q_F, D)$, se construye un nuevo autómata $A'_{EF} = (Q, \Sigma, \delta', q_0, q_F, D')$, en el que se incrementa el conjunto de transiciones δ' y se define una nueva función D' que asigna las probabilidades a las transiciones. Esta nueva función D', debe cumplir la propiedad de consistencia estocástica, es decir, que la suma de las probabilidades asociadas a todas las transiciones que parten de cualquier estado sea 1.

En la figura 4.3 se presenta parcialmente un conjunto de estados de un autómata ECGI. En trazo continuo se muestran las transiciones vistas en el proceso de aprendizaje para un estado genérico $q_k \in Q$ (p.e. $q_k \rightarrow q_i$, $q_k \rightarrow q_p$), además de una transición no vista (trazo discontinuo), que llamaremos de suavizado, $q_k \dashrightarrow q_j$. Para formalizar los diversos métodos de suavizado que definen los modelos ECGI extendidos, introduciremos la siguiente notación:

- F_k : frecuencia del estado q_k.
- f_{ki} : frecuencia de transición del estado q_k al estado q_i.

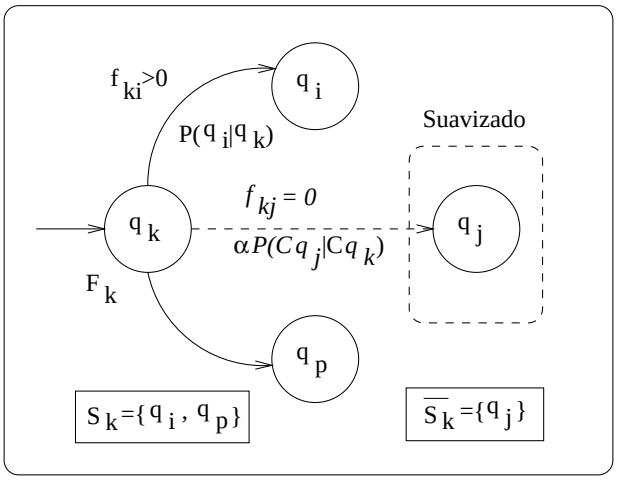

Figura 4.3: Notación utilizada para el suavizado de un modelo ECGI

- $\mathcal{C} = \{C_1, ..., C_N\}$: conjunto de símbolos (categorías léxicas).
- C_{q_i} : símbolo (categoría léxica) asociado al estado q_i.
- $P(q_i|q_k)$: probabilidad de transición del estado q_k al estado q_i (transiciones observadas en el entrenamiento).
- $\alpha P(C_{q_j}|C_{q_k})$: probabilidad de transición del estado q_k al estado q_j, proporcional, α, a la probabilidad de la secuencia de símbolos asociados a estos estados, $C_{q_k}C_{q_i}$(transiciones de suavizado añadidas).

4.4 Suavizado de Modelos ECGI

Los métodos de suavizado para modelos de n-gramas no son directamente aplicables para cualquier autómata de estados finitos. Aunque los modelos de n-gramas se pueden considerar como modelos regulares que llevan asociados los símbolos en los estados, el modelo de autómata que utilizaremos (AEF inferido con el algoritmo ECGI) presenta una estructura diferente, es decir, en un modelo de n-gramas no pueden aparecer dos estados etiquetados con un mismo símbolo, situación que si puede darse en un modelo ECGI (y en general en cualquier autómata).

Sea $\mathcal{C} = \{C_1, ..., C_N\}$ el conjunto de categorías léxicas utilizado (terminales), y q_k un estado cualquiera del autómata inferido, distinto del estado final. Definimos los siguientes conjuntos:

- $\mathcal{S}_k = \{q_i : f_{ki} > 0\}$: conjunto de estados sucesores del estado q_k vistos en el entrenamiento.
- $\mathcal{C}_k = \{C_{qi} : q_i \in \mathcal{S}_k\}$: conjunto de categorías léxicas correspondientes a los estados del conjunto $\mathcal{S}_k$.
- $\overline{\mathcal{C}_k} = \{C_{qj} : C_{qj} \notin \mathcal{C}_k \ \wedge \ q_j \notin \mathcal{S}_k\}$: conjunto de categorías no vistas como sucesores del estado q_k (conjunto complementario de $\mathcal{C}_k$ respecto de $\mathcal{C}$).
- $\overline{\mathcal{S}_k} = \{q_j : q_j \notin \mathcal{S}_k \ \wedge \ C_{q_j} \in \overline{\mathcal{C}_k} \ \wedge \ F_j = \max_{\forall q_i \notin \mathcal{S}_k \ \wedge \ C_{q_i} = C_{q_j}} (F_i)\}$: conjunto de estados sucesores de suavizado para el estado q_k (para garantizar que se pueda transitar del estado q_k a otro estado con cualquier símbolo de entrada perteneciente al conjunto $\mathcal{C}$). Como puede haber varios estados candidatos a sucesores de q_k con el mismo símbolo, elegiremos el de mayor frecuencia en el modelo ECGI.

En nuestro caso se dispone de dos funciones de distribución de probabilidad: la probabilidad de transición entre los estados del autómata (proporcionada por el algoritmo ECGI) y un modelo de bigramas de los símbolos utilizados (categorías pertenecientes a $\mathcal{C}$), ambas funciones estimadas a partir del mismo conjunto de entrenamiento. Nuestro objetivo consiste en combinarlas, mediante *interpolación lineal* o *back-off*, para definir las probabilidades del autómata extendido A'_{EF}.

4.4.1 Interpolación Lineal (IL)

La interpolación lineal es un método de suavizado en el que para estimar las transiciones entre estados se tienen en cuenta todas las funciones de probabilidad disponibles. En nuestro caso combinaremos las dos anteriormente comentadas. El factor de interpolación se puede definir de manera global y única para todos los estados del autómata –ILC– o bien se puede particularizar a cada estado –ILE–.

Interpolación Lineal Constante (ILC)

Para determinar la probabilidad interpolada $P^I(q_i|q_k)$, utilizaremos la expresión (4.1):

$$P^I(q_i|q_k) = (1-\lambda) \cdot P_{ECGI}(q_i|q_k) + \lambda \cdot P_{BIG}(C_{q_i}|C_{q_k}); \forall q_i \in \mathcal{S}_k \cup \overline{\mathcal{S}_k} \qquad (4.1)$$

El factor de interpolación λ, se define como la relación $\frac{n_1}{N}$, cumpliéndose que $0 \leq \lambda \leq 1$. Este valor se considera único para todos los estados del autómata y se calcula a partir de la información que se obtiene de los autómatas inferidos. En esta expresión, n_1 es el número de transiciones en el autómata con frecuencia 1 y N el número total de transiciones.

Esta interpolación también se aplicará al conjunto de transiciones $\overline{\mathcal{S}_k}$ de cada estado q_k (aunque en este caso las probabilidades $P_{ECGI}(q_i|q_k)$ serán igual a cero).

Debido a la naturaleza de los modelos ECGI utilizados (autómatas estocásticos no deterministas), es preciso aplicar un proceso de reescalado o normalización para mantener la consistencia estocástica, es decir, exigir que se cumpla:

$$\sum_{\forall i: q_i \in \mathcal{S}_k \cup \overline{\mathcal{S}_k}} P^I(q_i|q_k) = 1, \forall k$$

En la figura 4.4 se muestra la evolución del factor λ en función de la talla de entrenamiento. Estos valores se han obtenido experimentalmente a partir de un conjunto de muestras de aprendizaje del corpus *Wall Street Journal* (*WSJ*). En el apéndice A, se detallan el conjunto de etiquetas utilizadas en su anotación. También se muestra la evolución del número de estados del modelo inferido en función de la talla del conjunto de entrenamiento. Se observa que cuando aumenta la talla de entrenamiento, el número de estados del modelo se estabiliza (3,000 estados) siguiendo un comportamiento logarítmico. Por otra parte, el factor λ sigue un comportamiento inverso, estabilizándose alrededor de $\lambda = 0.2$. Los valores de λ obtenidos favorecen las probabilidades del modelo ECGI frente al de bigramas (BIG) en todos los casos, aumentando el peso que se les da a éstos a medida que se incrementa el volumen de datos de aprendizaje.

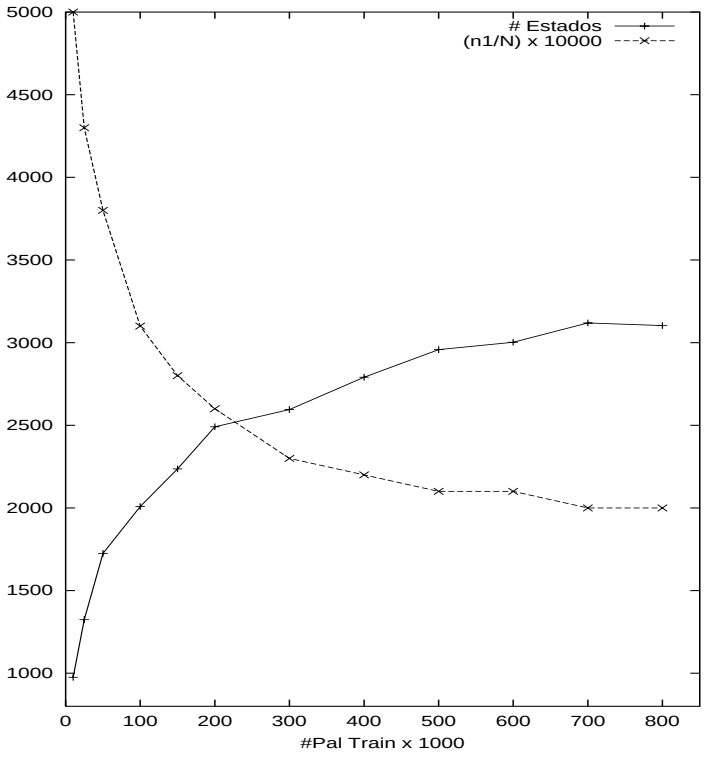

Figura 4.4: Evolución del número de estados del autómata y del valor n1/N en función de la talla de entrenamiento sobre el corpus WSJ.

Interpolación Lineal dependiente del Estado (ILE)

Otra alternativa, que intenta particularizar el valor de λ para cada estado, consiste en definir $\lambda_k = \frac{n_{1k}+1}{F_k+1}, 0 < \lambda_k \leq 1$. En este caso la expresión que se utiliza es (4.2), debiendo aplicar el mismo proceso de normalización que para el caso de ILC.

$$P^I(q_i|q_k) = (1 - \lambda_k) \cdot P_{ECGI}(q_i|q_k) + \lambda_k \cdot P_{BIG}(C_{q_i}|C_{q_k}); \forall q_i \in \mathcal{S}_k \cup \overline{\mathcal{S}_k} \qquad (4.2)$$

A pesar de que con esta aproximación la estimación de los parámetros λ_k tiene más relación con el estado considerado, como se verá más adelante, presenta el inconveniente de que se necesita un número mayor de muestras de aprendizaje, por lo que en los experimentos que hemos realizado no se obtiene ninguna mejora.

4.4.2 Back-off (B)

A diferencia de la Interpolación Lineal, el método de *back-off* utiliza para la estimación de cada evento sólo una función de distribución (no las combina como en el caso anterior). El conjunto de las nuevas probabilidades de transición suavizadas que se obtiene ($\widetilde{P}(q_i|q_k)$) viene dado por la expresión (4.3), donde se observa que las transiciones vistas son penalizadas con un descuento d, y las no vistas se estiman de manera proporcional (α) a la probabilidad del bigrama de los símbolos asociados a los estados considerados ($P(C_{q_i}|C_{q_k})$).

$$\widetilde{P}(q_i|q_k) = \begin{cases} d \cdot P(q_i|q_k) & f_{ki} > 0 \ (q_i \in \mathcal{S}_k) \\ \alpha \cdot P(C_{q_i}|C_{q_k}) & f_{ki} = 0 \ (q_i \in \overline{\mathcal{S}_k}) \end{cases} \tag{4.3}$$

En la sección 3.5.1 se ha descrito el suavizado de *back-off* con algunas funciones de descuento para modelos de n-gramas. A continuación presentamos las funciones de descuento que se han definido para los modelos ECGI.

Descuento Lineal (BDL)

Se aplica el mismo descuento a todas las transiciones de un estado. La cantidad a descontar en cada estado q_k, depende de la frecuencia del mismo, F_k y de las transiciones de frecuencia unidad, n_{1k}.

$$d\,(q_k) = 1 - \frac{n_{1k}+1}{F_k+1}$$

En el autómata inferido sin suavizar se cumple que

$$\sum_{\forall i:q_i \in \mathcal{S}_k} P(q_i|q_k) = 1, \forall k$$

Cuando se extiende el autómata ECGI y se aplica la función de descuento se debe cumplir que

$$\sum_{\forall i:q_i \in \mathcal{S}_k} \widetilde{P}(q_i|q_k) = \sum_{\forall i:q_i \in \mathcal{S}_k} d(q_k) \cdot P(q_i|q_k) = 1 - \frac{n_{1k}+1}{F_k+1}, \forall k$$

Se observa que la cantidad a repartir entre las transiciones no vistas viene dada por la expresión $CR(k) = \frac{n_{1k}+1}{F_k+1}$.

La interpretación del valor $CR(k)$ es la siguiente: cuanto mayor sea el número de transiciones de frecuencia 1 observadas (transiciones poco significativas) mayor será el valor de $CR(k)$, mientras que $CR(k)$ disminuirá cuanto mayor sea la frecuencia del estado considerado F_k (si un estado tiene alta frecuencia, generalmente sus transiciones serán más significativas). Para garantizar que $CR(k) > 0$ (puede haber estados en que $n_{1k} = 0$), añadimos 1 en el numerador y denominador.

La cantidad $CR(k)$ se reparte entre las transiciones de suavizado añadidas al autómata ECGIE de manera proporcional a la probabilidad del bigrama de los símbolos de los estados considerados.

Para que se cumpla que $\sum_{\forall i: q_i \in \mathcal{S}_k \cup \overline{\mathcal{S}_k}} \widetilde{P}(q_i|q_k) = 1, \forall k$, la suma de probabilidades de las transiciones no vistas para un estado genérico q_k debe valer:

$$\sum_{\forall i: q_i \in \overline{\mathcal{S}_k}} \widetilde{P}(q_i|q_k) = \frac{n_{1k}+1}{F_k+1}, \forall k$$

por lo que se debe introducir un factor de normalización α dado por:

$$\alpha = \frac{\frac{n_{1k}+1}{F_k+1}}{\sum_{\forall i: q_i \in \overline{\mathcal{S}_k}} P(C_{q_i}|C_{q_k})}$$

Obsérvese que para los estados que cumplen que $n_{1k} = F_k$ (estados de los que sólo parten transiciones de frecuencia 1), la función de descuento $d(q_k)$ vale 0.

La solución adoptada en estos casos consiste en sustituir la probabilidad de transición del autómata ECGI por la correspondiente probabilidad de transición del bigrama de los símbolos asociados a los mismos, es decir, realizar la siguiente aproximación: $P(q_i|q_k) \approx P(C_{q_i}|C_{q_k})$. En consecuencia, la masa de probabilidad a repartir entre los sucesos no vistos para estos estados vendrá dada por $1 - \sum_{\forall i: f_{ki}=1} P(C_{q_i}|C_{q_k})$.

Descuento en Función de la Frecuencia (BDFF)

El descuento lineal presentado anteriormente aplica el mismo descuento a todas las transiciones de un estado independientemente de su frecuencia. Una solución más adecuada consiste en definir una función de descuento que penalice más a las transiciones de menor frecuencia y menos a las de mayor.

Si se supone, al igual que en el caso anterior, que el descuento total que se desea aplicar en cada estado debe ser $\frac{n_{1k}+1}{F_k+1}$, definimos la función de descuento $d\,(f_{ki})$ como

$$d\,(f_{ki}) = (1 - \frac{C(k)}{f_{ki}});\ \text{donde}\ \ C(k) = \frac{n_{1k}+1}{F_k+1} \cdot \frac{F_k}{N_k},\ (0 < C(k) \leq 1)$$

$C(k)$ es una constante positiva definida para cada estado q_k que depende de la frecuencia del estado(F_k), del número total de transiciones que parten del estado (N_k) y de las transiciones de frecuencia 1 que parten del mismo.

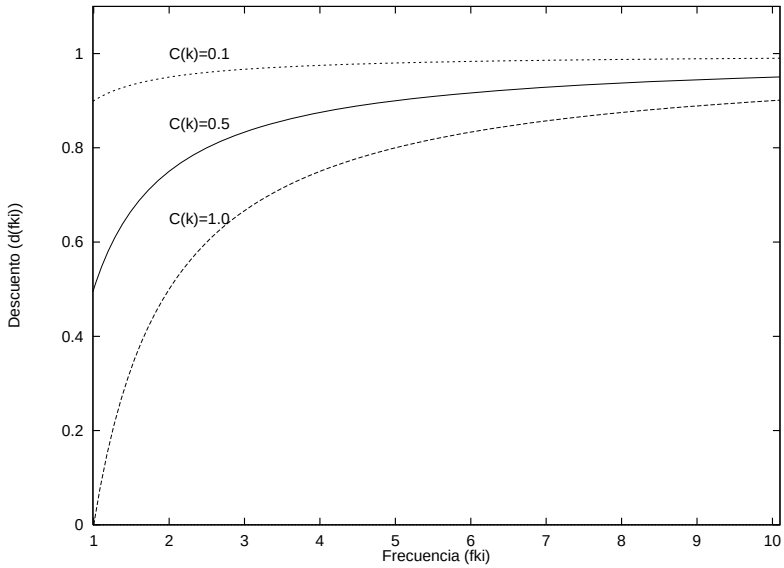

Figura 4.5: Comportamiento de la función de descuento en función de la frecuencia (BDFF) considerando distintos valores de $C(k)$.

En la figura 4.5 se puede observar el comportamiento de la función $d\,(f_{ki})$ para distintos valores de $C(k)$ en función de la frecuencia de transición. El valor $C(k) = 0$, se corresponde con la función constante $d\,(f_{ki}) = 1$. Esta situación no se da en nuestra aproximación puesto que siempre se aplica un descuento a todas las transiciones. El valor $C(k) = 1$ se representa en la curva inferior. En nuestra aproximación sólo se puede dar este caso para estados que cumplan que $n_{1k} = F_k$ (y en consecuencia igual a N_k). Esto obliga a que todas las transiciones que parten del estado sean de frecuencia 1 (se corresponde con el punto $f_{ki} = 1$ de la curva). En este caso la función de descuento $d\,(f_{ki}) = 0$ para todas las transiciones que parten del estado. La solución adoptada es la misma que para el caso de BDL, es decir,

aproximar la probabilidad de transición mediante la expresión $P(q_i|q_k) \approx P(C_{q_i}|C_{q_k})$ y repartir la cantidad $1 - \sum_{\forall i: f_{ki}=1} P(C_{q_i}|C_{q_k})$ entre las transiciones no vistas en ese estado.

Se puede comprobar que el comportamiento de la función de descuento $d\,(f_{ki})$ es el esperado. Las probabilidades con frecuencia menor, tienen valores de $d\,(f_{ki})$ menores, con lo que al multiplicarlos por las probabilidades asociadas que se obtienen en el proceso de aprendizaje, se ven más penalizadas.

En la figura 4.6 se muestra la distribución de los valores de $C(k)$, obtenida experimentalmente a partir de una muestra del corpus WSJ formada por 800,000 palabras. Aunque la distribución es bastante homogénea, los valores más frecuentes aparecen en los intervalos $]0.1, 0.5[(53\%)$ y $]0.9, 1[(12\%)$.

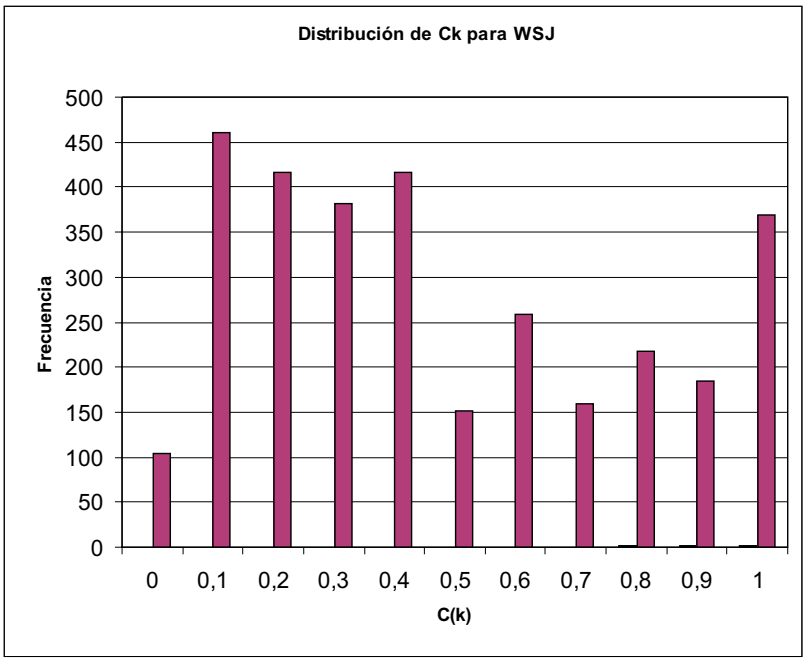

Figura 4.6: Distribución de $C(k)$ sobre el corpus WSJ con un conjunto de entrenamiento de 800,000 palabras.

Al igual que en el caso de BDL, se debe cumplir que para las transiciones vistas

en el proceso de aprendizaje:

$$\sum_{\forall i:q_i \in \mathcal{S}_k} d\left(f_{ki}\right) \cdot P(q_i|q_k) = 1 - \frac{n_{1k}+1}{F_k+1}, \forall k \tag{4.4}$$

y para las transiciones no vistas:

$$\sum_{\forall i:q_i \in \overline{\mathcal{S}_k}} \widetilde{P}(q_i|q_k) = \sum_{\forall i:q_i \in \overline{\mathcal{S}_k}} \alpha \cdot P(C_{q_i}|C_{q_k}) = \frac{n_{1k}+1}{F_k+1},\ \forall k;$$

donde el factor de normalización α viene dado por:

$$\alpha = \frac{\frac{n_{1k}+1}{F_k+1}}{\sum_{\forall i:q_i \in \overline{\mathcal{S}_k}} P(C_{q_i}|C_{q_k})}$$

4.5 Evaluación Experimental de los Modelos Contextuales ECGIE

Para la evaluación de los diferentes métodos de suavizado de los modelos ECGIE se presenta un conjunto de experimentos de etiquetado léxico sobre el corpus WSJ utilizando conjuntos de entrenamiento de diferente talla.

En un MM la información contextual se modeliza mediante un n-grama, que como ya se ha comentado, puede ser visto como un autómata de estados finitos. Asociado a cada estado se dispone de un conjunto de probabilidades léxicas que dan cuenta de la probabilidad de emisión de las diferentes palabras del vocabulario en ese estado. En la aproximación planteada en este trabajo, el modelo contextual se puede representar indistintamente por un modelo de bigramas o por un modelo ECGIE.

Para que la evaluación del modelo contextual sea independientemente de las probabilidades léxicas, se ha asumido la siguiente hipótesis: *se supone una distribución de probabilidad léxica equiprobable para todos las palabras en cada uno de los estados.* Esta asunción, evidentemente, disminuye las prestaciones de etiquetado, pero hace que sólo se tenga en cuenta el Modelo Contextual, que es lo que se pretende contrastar. En el siguiente capítulo se presentarán resultados más completos en tareas de etiquetado léxico.

En la tabla 4.1 se muestra el resultado de precisión de etiquetado con diferentes modelos contextuales, bigramas (BIG) y modelos ECGIE con diversos suavizados (BDFF, DBDL, ILE, ILC). La evaluación se ha realizado sobre el corpus WSJ tomando conjuntos de entrenamiento de diferente talla y eligiendo un conjunto de prueba común de 100,000 palabras aproximadamente.

En la figura 4.7 se muestra también de manera gráfica el resultado de la evaluación. Se puede observar, que cuando se utiliza un número reducido de muestras de aprendizaje (menos de 200,000 palabras), los resultados de precisión utilizando modelos de bigramas (BIG) son superiores a los obtenidos con modelos ECGIE. A partir de este valor, la tendencia se invierte y con los modelos ECGIE se obtiene una mayor precisión, principalmente con aquellos suavizados que tienen en cuenta las propiedades del estado considerado (BDFF , BDL e ILE). Además, mientras que

la precisión usando un modelo BIG tiende a estabilizarse a partir de 300,000 muestras de aprendizaje, los modelos ECGIE presentan un comportamiento que mejora en cada partición.

Entrenamiento	% Precisión de Etiquetado				
#Palabras x10^3	BIG	BDFF	BDL	ILE	ILC
10	92.40	91.99	91.96	91.84	91.96
25	92.59	92.42	92.42	92.42	92.49
50	92.84	92.67	92.69	92.53	92.60
100	92.95	92.92	92.90	92.84	92.82
150	92.95	92.93	92.97	92.94	92.86
200	92.94	93.13	93.13	93.1	92.93
300	92.98	93.13	93.10	93.08	92.95
400	92.98	93.16	93.16	93.16	92.95
500	92.99	93.18	93.20	93.16	93.00
600	92.99	93.26	93.27	93.25	93.03
700	92.96	93.33	93.33	93.28	93.09

Tabla 4.1: Evaluación de la precisión de etiquetado sobre el WSJ para los modelos BIG y ECGIE considerando un modelo léxico equiprobable.

En la figura 4.8 se presenta la precisión de etiquetado, para la última partición, usando los diferentes modelos contextuales y considerando un intervalo de confianza del 95%. Como se puede observar, con los métodos de *back-off* (BDL y BDFF) se obtienen diferencias significativas, dentro de este intervalo, con respecto a los modelos de bigramas (BIG).

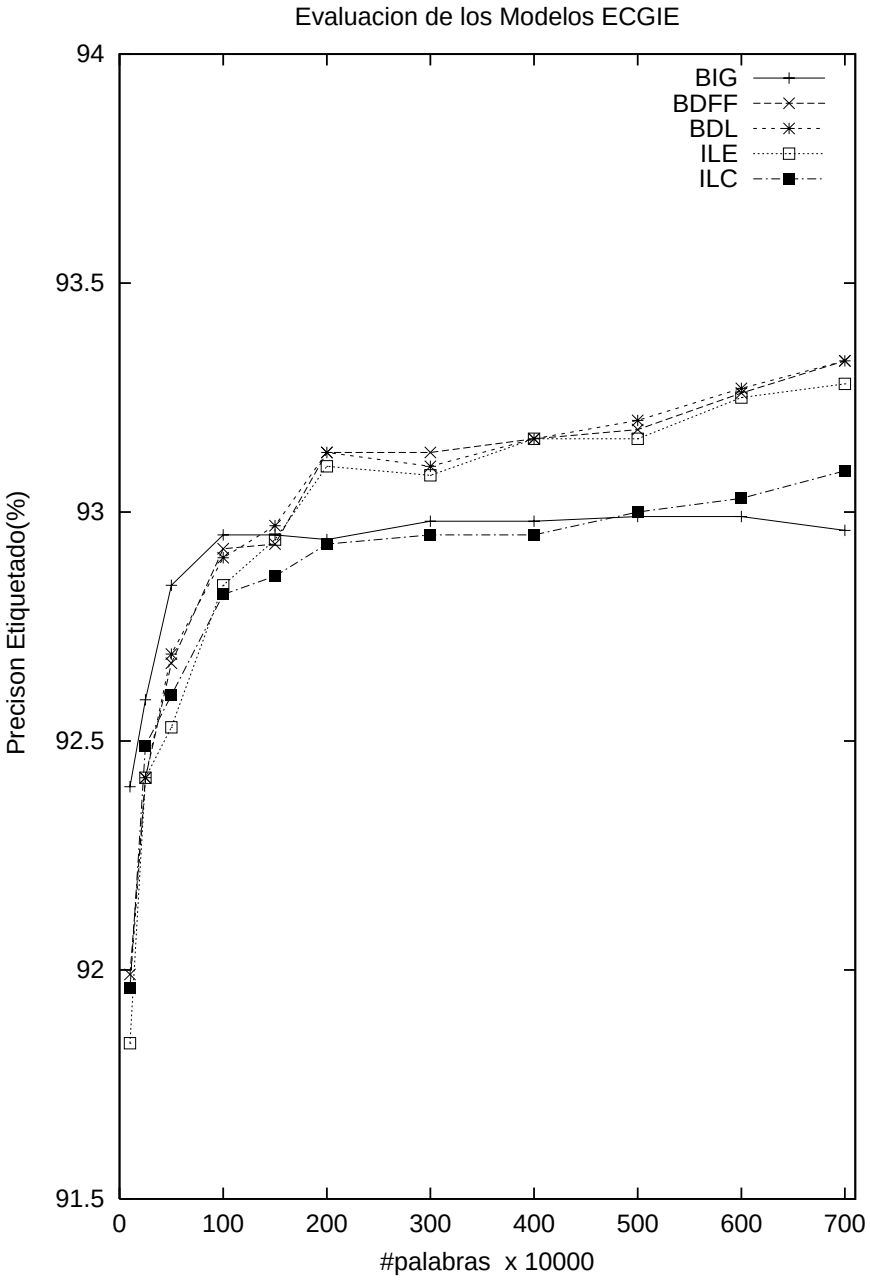

Figura 4.7: Evaluación de los métodos de suavizado en función de la talla de entrenamiento considerando un modelo léxico equiprobable.

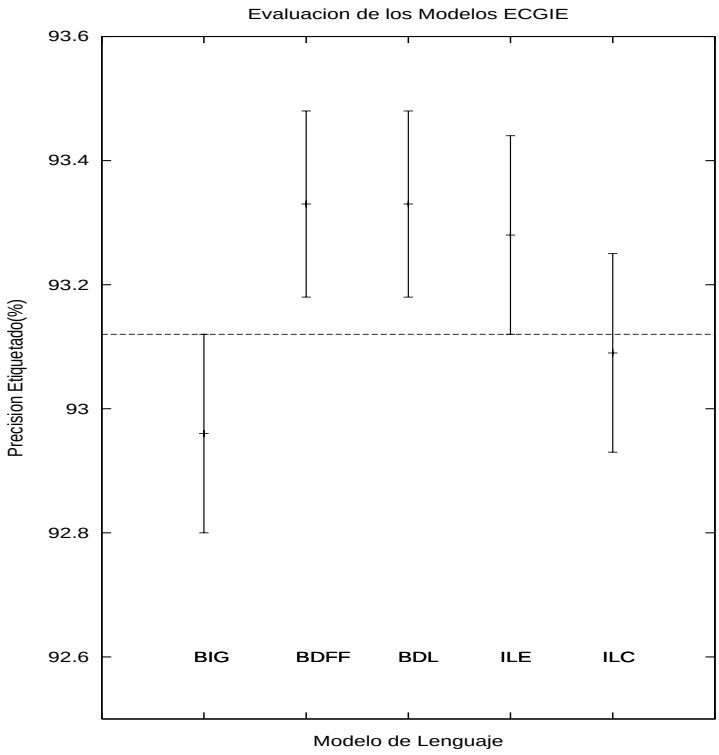

Figura 4.8: Evaluación de los métodos de suavizado considerando un modelo léxico equiprobable para un conjunto de entrenamiento de 700,000 palabras y uno de prueba de 100,000 palabras.

4.6 Evaluación de los Modelos Especializados

En esta sección se presenta un estudio experimental para comprobar el efecto de la especialización de los modelos contextuales (presentados en la sección 3.6) en las prestaciones de etiquetado. El experimento se ha realizado utilizando los mismos conjuntos de aprendizaje definidos anteriormente sobre el corpus *WSJ* (tabla 4.1 y figura 4.8), así como el mismo conjunto de evaluación (100,000 palabras). De la misma forma se ha considerado un Modelo Léxico equiprobable.

El conjunto $\mathcal{W}_e$, formado por las palabras a especializar, se ha determinado experimentalmente a partir del conjunto de aprendizaje, considerando aquellas palabras de mayor frecuencia, salvo símbolos de puntuación, números, nombres propios y algunas palabras que no aportaban ninguna mejora. Con este criterio el conjunto $\mathcal{W}_e$

está formado por 485 palabras (véase apéndice C.1).

En la tabla 4.2 se presenta una comparación de la precisión de etiquetado entre modelos de bigramas especializados (BIG_{esp}) y modelos de bigramas (BIG) correspondientes a los presentados en la sección 4.5. Se observa que al incorporar el léxico al modelo contextual se obtienen diferencias de etiquetado significativas. Para la última partición, se obtienen diferencias significativas de etiquetado, 94.56 ± 0.14 frente a 92.96 ± 0.16, considerando un intervalo de confianza del 95%. Como contrapartida a la mejora, la talla de los modelos contextuales también aumenta. Al realizar la especialización pasamos de un modelo con 47 estados (BIG) a uno de 690 estados (BIG_{esp}) para la última partición. Este aumento, aunque es considerable, no ralentiza de manera importante para nuestros propósitos el proceso de etiquetado léxico.

Entrenamiento	Precisión (%)	
#Palabras x10^3	BIG	BIG$_{esp}$
50	92.84	93.99
100	92.95	94.30
150	92.95	94.39
200	92.94	94.44
300	92.98	94.50
400	92.98	94.54
500	92.99	94.53
600	92.99	94.60
700	92.96	94.56

Tabla 4.2: Evaluación de la precisión de etiquetado sobre el WSJ para los modelos de bigramas (BIG) y bigramas especializados (BIGesp) considerando un modelo léxico equiprobable.

La especialización también se puede aplicar a los modelos *ECGIE*, pero en este caso, si que se ven afectadas las prestaciones del sistema. En la tabla 4.3 se presentan los resultados de la especialización sobre las dos primeras particiones del corpus WSJ. Se puede observar que el tamaño de los modelos contextuales (modelos ECGIE

Entrenamiento	#Estados		Precisión (%)	
#Palabras x10^3	ECGIE	ECGIE$_{esp}$	ECGIE	ECGIE$_{esp}$
50	1700	4300	92.67	93.55
100	2000	5600	92.92	93.92

Tabla 4.3: Evaluación de la precisión de etiquetado sobre el corpus WSJ para modelos ECGIE y ECGIEesp considerando un modelo léxico equiprobable.

con suavizado BDFF) crece de manera considerable cuando se realiza la especialización. Este hecho aumenta la complejidad espacial y temporal, tanto del proceso de aprendizaje, como del proceso de etiquetado. También se observa, al igual que en el caso de bigramas, que la precisión de etiquetado se incrementa cuando se realiza la especialización.

Debido a que nuestro interés en el uso de etiquetadores va encaminado principalmente a su incorporación en sistemas de PLN, pensamos que la mayor prioridad a exigir a nuestro sistema debe ser su eficiencia computacional, a parte evidentemente, de unas precisiones aceptables. Aunque esta eficiencia, se podría incrementar introduciendo técnicas de búsqueda en haz, para acelerar el proceso de etiquetado, en lo sucesivo, para los experimentos que se presentan, utilizaremos la especialización sólo para los modelos de bigramas. No obstante, debido a que los resultados de etiquetado obtenidos con modelos ECGI$_{esp}$, en algunos casos, son complementarios a los obtenidos con modelos BIG y BIG$_{esp}$, se estudiará en un futuro la posibilidad de combinar estos resultados para aumentar las prestaciones del sistema.

4.7 Resumen

En este capítulo se ha estudiado la viabilidad de utilizar técnicas de inferencia gramatical para obtener modelos contextuales y su aplicación al problema del etiquetado léxico de textos. Se ha presentado brevemente el algoritmo ECGI para inferir autómatas de estados finitos a partir de secuencias de categorías léxicas.

Se ha propuesto un mecanismo de extensión de los mismos (modelos ECGIE) adaptando los métodos de suavizado más usuales en modelos de bigramas: interpolación lineal y *back-off*. En ese sentido, se han propuesto cuatro mecanismos de

suavizado (ILC, ILE, BDL. BDFF).

La evaluación experimental se ha realizado sobre el corpus WSJ, mediante un experimento de etiquetado léxico, en el que se ha considerado un modelo léxico equiprobable. Bajo estas condiciones, para todos los suavizado propuestos, se obtienen valores de precisión superiores a los obtenidos con modelos de bigramas. Para los modelos BDFF y BDL se obtienen valores de precisión ($93.33\% \pm 0.15\%$) que son superiores a los de bigramas ($92.96\% \pm 0.16\%$) con unas diferencias que son significativas considerando un intervalo de confianza del 95%.

Se han utilizado, lo que hemos llamado *Modelos Contextuales Especializados*, con el fin de hacer intervenir ciertas palabras, junto con las categorías, en los modelos contextuales. Está lexicalización de los modelos, aunque aumenta el tamaño de los mismos, supone un incremento considerable de la precisión de etiquetado. La aproximación se puede aplicar a todos los modelos estudiados: bigramas y ECGIE. Sin embargo, para el problema de etiquetado léxico, se considera más adecuado su uso para los modelos de bigramas, ya que para los modelos ECGI, se obtienen modelos cuya talla ralentiza de manera considerable el proceso de aprendizaje y de etiquetado. Cuando se aplica la lexicalización a los modelos de bigramas el valor de la precisión se incrementa con diferencias significativas al 95%, pasando de $92.96\% \pm 0.16\%$ a $94.56\% \pm 0.14\%$.

Capítulo 5

Descripción y Evaluación del Sistema de Etiquetado Léxico

En este capítulo se describe el sistema de desambiguación léxica propuesto en esta tesis basado en la utilización de modelos de estados finitos estocásticos. Se describen los dos procesos involucrados: aprendizaje de los modelos a partir de corpora y etiquetado léxico de textos. A continuación se describen los corpora sobre los que se realizará la evaluación del sistema: WSJ para el inglés, *LexEsp* y *BDGEO* para el castellano. Además, se describe la tarea de etiquetado léxico y supervisión del corpus *BDGEO*.

5.1 Descripción del Sistema de Etiquetado

En la figura 5.1 se presenta un esquema del sistema de desambiguación léxica que proponemos en este trabajo. En él se distinguen dos fases: *Aprendizaje* y *Etiquetado*. En la fase de *Aprendizaje* se obtienen los Modelos de Lenguaje (contextuales) y el Modelo Léxico (que define las probabilidades léxicas). En la fase de *Etiquetado*, a partir del análisis morfológico de las palabras de la frase de entrada, se realiza la desambiguación de las palabras en sus diferentes categorías léxicas teniendo en cuenta los modelos contextuales y léxicos aprendidos.

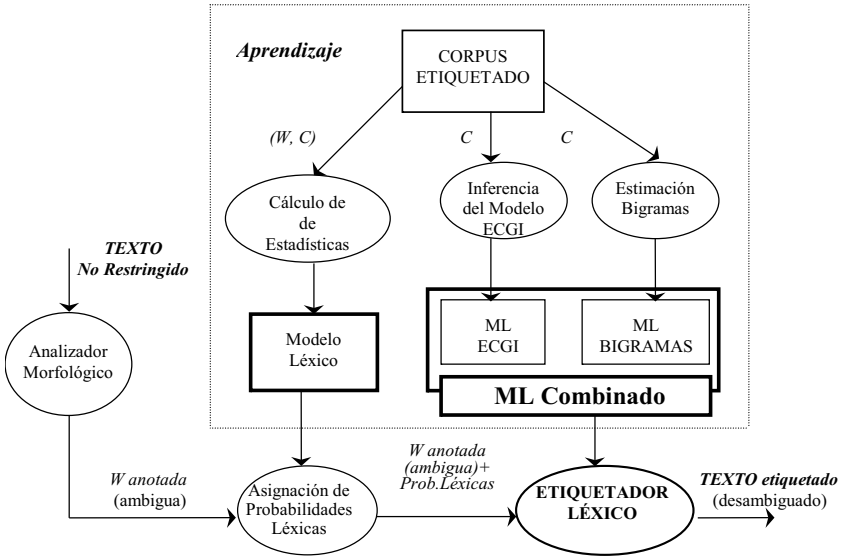

Figura 5.1: Descripción del sistema de etiquetado léxico.

5.1.1 Fase de Aprendizaje

La fase de *aprendizaje* de los modelos se realiza a partir de un corpus etiquetado formado por pares de secuencias de palabras y de categorías léxicas <W,C>. Los corpora, generalmente revisados por expertos, se suponen libre de errores de etiquetado, aunque en la práctica no es lo habitual y suelen presentar alrededor de un 3% de error. A partir de las secuencias de categorías (C) se aprenden los modelos de lenguaje contextuales. Estos pueden ser cualquier modelo regular de los presentados en los capítulos anteriores: bigramas (BIG), modelos $ECGIE$ suavizados, utilizando alguno de los métodos propuestos (ILC, ILE, BDL, $BDFF$) o bien, cualquiera de los anteriores modelos especializados en un determinado número de palabras.

El modelo de bigramas (BIG) se puede estimar y suavizar utilizando cualquiera de las técnicas utilizadas para los MM. Nuestro sistema utiliza directamente la herramienta *SLM Toolkit*, descrita en (Rosenfeld, 1994; Clarksond and Ronsenfeld, 1997). Ésta se compone de un conjunto de utilidades que permiten construir modelos de lenguaje de n-gramas y evaluarlos por medio de la perplejidad del conjunto de prueba. En nuestros experimentos se ha utilizado para construir modelos contextuales de bigramas de categorías léxicas (o categorías especializadas), suavizados con *back-off* con la función de descuento *Good-Turing*. Una vez construidos se representan como

autómatas de estados finitos tal y como se ha descrito en capítulo 3.

Los modelos *ECGI* se infieren mediante el algoritmo presentado en el capítulo 4 y se extienden por medio de alguna de las técnicas de suavizado propuestas (*ILC, ILE, BDL, BDFF*).

El *Modelo Léxico* se estima de la manera usual, teniendo en cuenta los pares formados por cada palabra y su correspondiente categoría <W,C>extraidos a partir de textos etiquetados. Se calculan las frecuencias de las palabras, categorías y de cada palabra en cada categoría. Estos datos son necesarios para estimar las probabilidades léxicas $P(w_i|c_i)$. Debido al hecho de que éstas probabilidades son, por un lado, más difíciles de estimar, y por otro, presentan dificultad para aplicar los métodos de suavizado directamente, se han calculado las simétricas, $P(c_i|w_i)$, y se ha realizado el correspondiente suavizado. Para las palabras desconocidas (no contempladas en el conjunto de entrenamiento), su probabilidad se aproxima por la probabilidad de la clase de ambigüedad correspondiente de la palabra, y si ésta es cero, por la probabilidad de la categoría. Alternativamente, para estos casos también se ha aplicado el suavizado conocido como '*añadir uno*', obteniéndose resultados similares. A continuación, se realiza un proceso de renormalización y se aplica la regla de Bayes para calcular las probabilidades simétricas buscadas ($P(w_i|c_i)$).

5.1.2 Fase de Etiquetado

La fase de *etiquetado* o desambiguación léxica toma como entrada un texto no restringido el cual es convenientemente separado en unidades o *'tokens'*. Para cada *'token'* detectado, haciendo uso de un analizador morfológico, o en su defecto de un diccionario, se obtiene para cada uno de ellos sus posibles categorías léxicas. Nuestro sistema puede tomar textos de entrada, tanto en inglés, como en castellano. Dependiendo de la lengua utilizada el proceso de extracción de unidades lingüísticas (*tokenización*) y análisis morfológico es distinto.

Para el inglés se utiliza un diccionario construido en la fase de aprendizaje a partir de todo el corpus de la tarea que se esté considerando. El diccionario recoge para cada palabra las posibles categorías vistas en el corpus. Para cada unidad de entrada ('token') se consulta el mismo y éste proporciona las posibles categorías. En estas condiciones, el diccionario hace el papel de un analizador morfológico ideal

que es capaz de proporcionar todas las etiquetas en que se ha visto una determinada palabra, pero restringido al corpus utilizado en la aplicación.

Para el castellano se utiliza un analizador morfológico, en concreto el analizador *MACO* (Carmona et al., 1998). Inicialmente, el analizador segmenta los textos de entrada a etiquetar en unidades (*'tokens'*). Es capaz de identificar signos de puntuación, abreviaciones, números, nombres propios, agrupaciones léxicas (locuciones adverbiales, fechas, concatenaciones de nombres propios ...), etc. Como salida proporciona para cada *'token'* detectado, todas sus posibles categorías léxicas.

Independientemente de cómo se realice el preproceso de los textos de entrada, se toman los diferentes *'tokens'* con sus posibles categorías léxicas y teniendo en cuenta el modelo léxico, se asignan las respectivas probabilidades léxicas ($P(w_i|c_i)$). Éstas probabilidades, junto con el Modelo de Lenguaje elegido, que en cualquier caso viene representado utilizando el formalismo homogéneo de autómatas de estados finitos, componen la entrada al etiquetador.

El proceso de etiquetado se realiza como se ha presentado en el capítulo 4 mediante el algoritmo de Viterbi (Viterbi, 1967). El proceso consiste en encontrar la secuencia de estados de mayor probabilidad en el modelo contextual elegido para un texto de entrada, teniendo en cuenta las probabilidades léxicas. Una vez obtenida dicha secuencia, como cada estado tiene asociada una única categoría léxica, se dispone de la mejor secuencia de categorías (etiquetado léxico) para la secuencia de palabras (*'tokens'*) de entrada.

5.2 Descripción de los Corpora

En esta sección se describen los corpora utilizados para la contrastación experimental de los métodos de etiquetado propuestos. Para el inglés se ha utilizado una porción del corpus *Wall Street Journal* etiquetado de acuerdo al conjunto de etiquetas definido en el *Penn Treebank*. Para el castellano se ha usado el corpus *LexEsp* etiquetado con el conjunto de etiquetas *PAROLE*.

Con el fin de estudiar el comportamiento del sistema para una tarea de etiquetado de corpora compuestos por frases de estructura diferente a las del conjunto de aprendizaje, se ha utilizado el corpus *BDGEO*. Debido a que este corpus no estaba

anotado con categorías léxicas, se ha utilizado el corpus LesExp como conjunto de aprendizaje para realizar un etiquetado léxico inicial que, después, se ha refinado utilizando una técnica de '*bootstrapping*' para adaptar los modelos de lenguaje. El conjunto de etiquetas léxicas utilizado en estos corpora se presenta en el apéndice A.

5.2.1 Wall Street Journal (WSJ)

El corpus *Wall Street Journal* es, sin lugar a dudas, el conjunto de datos en inglés más ampliamente utilizado como conjunto de aprendizaje y como banco de prueba de las diferentes aproximaciones propuestas en la literatura para resolver el problema de etiquetado léxico. El subconjunto del mismo, *Penn Treebank*, consta de 1,170,000 palabras etiquetas léxicamente utilizando el conjunto de etiquetas descrito en el apéndice A.3. En los experimentos que se presentan se ha utilizado el *Penn Treebank* para construir un diccionario de etiquetas. Este consta de unas 49,000 entradas que proporciona las posibles categorías léxicas asociadas a cada palabra. El diccionario se ha utilizado en los experimentos realizados para simular un analizador morfológico para el inglés, como ya se ha comentado anteriormente. Se han definido, de manera aleatoria, una parte de entrenamiento (700,000 palabras) y una de prueba (100,000 palabras) que son las que se han utilizado en todos los experimentos de etiquetado. Estas particiones coinciden con las usadas en los experimentos del capítulo anterior. No se ha utilizado todo el corpus, con el fin de disponer en un futuro de una parte que se pueda usar para refinar ciertos parámetros de los modelos contextuales o para estudiar la distribución de las palabras desconocidas por categoría en un texto nuevo, etc. Estas mismas particiones del corpus están analizadas sintácticamente de manera global, por lo que también se han utilizado para contrastar nuestra aproximación al análisis sintáctico superficial que se presentará en el siguiente capítulo.

5.2.2 LexEsp

LexEsp es un conjunto de textos no restringidos en castellano, etiquetado léxicamente y que aborda diferentes temáticas como noticias, literatura, artículos científicos, etc. Consta aproximadamente de 5.5 millones de palabras. Se dispone de una parte supervisada manualmente (100.000 palabras aproximadamente), que hemos divido en dos partes, una para aprendizaje de los modelos y otra para la evaluación del

etiquetado.

Las etiquetas utilizadas en este corpus son las definidas en el proyecto *PAROLE*. Este conjunto consta de unas 230 etiquetas, estructuradas en categorías y subcategorías, y que contemplan diferentes aspectos como género, número, tiempo verbal, persona, etc. El conjunto de etiquetas completo se describe en el apéndice A.1. Debido al reducido número de muestras de aprendizaje supervisadas disponibles se ha reducido el conjunto completo de etiquetas considerando solamente información referente a la categoría y la subcategoría, con lo que el número de etiquetas se reduce considerablemente, pasando a ser de 62 etiquetas (véase apéndice A.2). Así, por ejemplo, la etiqueta *NCFS000*, que significa Nombre Común Femenino Singular, pasa a ser *NC*, la etiqueta *VMIP3S* (Verbo Principal de Indicativo tercera Persona del Singular) a *VMI*, y así para todas las demás. A pesar de que con esta reducción se pierde cierta información estructural en los modelos de lenguaje aprendidos, es una simplificación que resulta útil adoptar debido a las limitaciones en el número de muestras de aprendizaje. No obstante, una vez resuelta la desambiguación con el conjunto reducido de etiquetas, se puede recuperar la etiqueta completa, utilizando la información proporcionada por el analizador morfológico. En algunas categorías, principalmente en las verbales, este proceso puede resultar ambiguo, con lo que en estos casos se puede optar por proporcionar las diferentes etiquetas posibles para esa palabra o aplicar nuevos métodos que permitan la resolución de este tipo de ambigüedad. Para todos los experimentos que se presentan en este capítulo se utilizará el conjunto reducido de etiquetas formado por 62 unidades, por lo que desambiguación en todos los experimentos ha sido total.

5.2.3 BDGEO

BDGEO es un corpus desarrollado dentro del proyecto *ALBAYZIN* (Díaz et al., 1998) y está compuesto por un conjunto de frases en castellano de consulta a una base de datos geográfica. Los tipos de oraciones son, principalmente, interrogativas directas, indirectas e imperativas, restringidas al dominio semántico de la tarea. Consta de 9,292 frases (103,880 palabras) y de un vocabulario de unas 1,340 palabras diferentes.

El hecho de que este corpus se haya definido sobre un universo semánticamente

restringido, lo hace especialmente útil para abordar tareas de comprensión sobre dominios particulares. Además, como la estructura sintáctica de las frase es bastante homogénea, se ha utilizado para mostrar que en estos casos, la aproximación basada en modelos *ECGIE*, frente a la basada en modelos de bigramas, presenta unos resultados de etiquetado ligeramente superiores.

El corpus *BDGEO* recoge un conjunto de frases restringidas a un dominio concreto (consultas a una base de datos geográfica) pero no está anotado ni léxica ni sintácticamente. A continuación se describe el proceso seguido para el etiquetado léxico de este corpus.

Descripción del proceso de etiquetado léxico del corpus BDGEO.

El objetivo de los experimentos que hemos realizado van encaminados, por una parte, a estudiar cómo se comporta el etiquetador ante un corpus diferente al utilizado en la fase de aprendizaje (distinta estructura sintáctica y distinto dominio semántico) y por otra, cómo conseguir corpus etiquetados y supervisados con un mínimo esfuerzo humano.

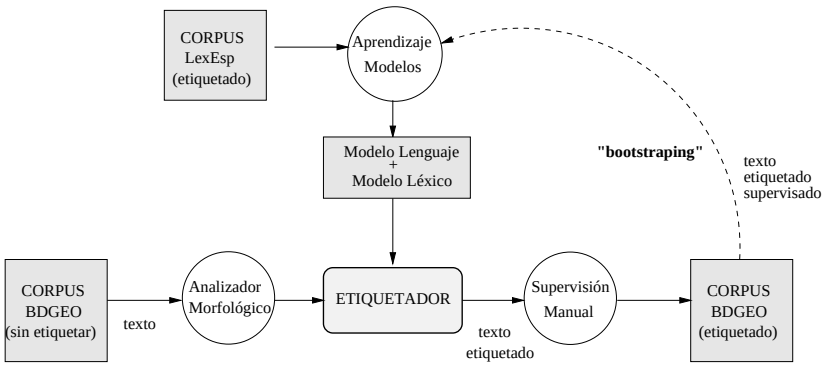

Figura 5.2: Descripción del proceso de etiquetado léxico del corpus BDGEO.

Para el etiquetado del corpus *BDGEO* (Pla and Molina, 1999) se ha seguido la técnica de '*bootstrapping*' que se muestra en la figura 5.2. Los modelos contextuales y léxicos aprendidos a partir de corpus *LexEsp* (modelos *ECGIE* con suavizado *BDFF*, aunque se podrían haber considerado cualquiera de los descritos) se han utilizado para etiquetar una pequeña partición del corpus *BDGEO*. Posteriormente

este conjunto se ha revisado manualmente y se ha incorporado al proceso de aprendizaje para adaptar los modelos iniciales. Este proceso se ha repetido para diferentes particiones.

Para obtener los modelos iniciales se ha utilizado una partición del corpus *LexEsp* etiquetada y supervisada manualmente. Sobre el corpus *BDGEO* se han definido tres particiones al azar (*BD1, BD2* y *BD3*), de talla predeterminada y cuyas características se muestran en la tabla 5.1. Para cada una de ellas se presenta el número de frases y de palabras, así como el índice de ambigüedad (sobre las palabras ambiguas y sobre el total).

Particiones	# Frases	# Palabras	# Palabras Ambiguas	Ambigüedad Media
LexEsp	3,360	65,854	26,183(39.76%)	1.62 etiquetas/palabra
BD1	850	9,716	4,071(41.90%)	1.61 etiquetas/palabra
BD2	854	10,133	4,311(42.54%)	1.62 etiquetas/palabra
BD3	7,588	84,031	33,700(40.10%)	1.59 etiquetas/palabra

Tabla 5.1: Particiones utilizadas en el etiquetado léxico del corpus BDGEO.

En la tabla 5.2 se muestra el resultado de los experimentos realizados. Para cada uno de ellos se indica el conjunto de entrenamiento (supervisado) y de prueba usados así como el error de etiquetado (palabras mal etiquetadas respecto a la referencia). Este proceso de '*bootstrapping*' seguido sirve para simplificar la tarea de supervisión del etiquetado automático, por lo que se puede obtener un conjunto de entrenamiento cada vez mayor y adaptado a la tarea. En *E1* el entrenamiento se ha realizado con un conjunto de datos no pertenecientes a la tarea. Para evaluar el error de etiquetado sobre la partición *BD1*, previamente se ha realizado un proceso de supervisión manual. En *E2* se ha incorporado *BD1* al entrenamiento y se ha procedido de la misma manera para evaluar el etiquetado sobre la partición *BD2*. Como se puede observar, al incorporar frases de entrenamiento específicas de la tarea, la precisión de etiquetado aumenta de manera significativa considerando un intervalo de confianza del 95%. Se pasa de una precisión del 97.61 ± 0.30 a 99.21 ± 0.17 (de 2.39% a 0.79% en términos de error de etiquetado).

Para comprobar que las mejoras obtenidas son independientes de las particiones elegidas, se han realizado los experimentos *E1'* y *E2'* en los que se observa resultados similares. Finalmente, en el experimento *E3* se ha incorporado un mayor número de frases de la tarea (*BD1* ∪ *BD2*). Para evaluar el error de etiquetado sobre *BD3*, se han elegido un conjunto de frases al azar de esta partición (10%), se ha supervisado y evaluado el error de etiquetado y se ha extrapolado al resto, con lo que se estima un error total del orden del 0.5%.

Experimento	Cjto.Entrenamiento	Cjto.Prueba	Error
E1	LexEsp	BD1	2.39% (233 palabras)
E2	LexEsp ∪BD1	BD2	0.79% (81 palabras)
E3	LexEsp ∪BD1 ∪BD2	BD3	0.5 % –estimado–
E1'	LexEsp	BD2	2.67% (271 palabras)
E2'	LexEsp ∪BD2	BD1	0.68% (67 palabras)

Tabla 5.2: Resultados de etiquetado sobre el corpus BDGEO.

En el apéndice B se muestra un conjunto de frases pertenecientes al corpus BDGEO etiquetadas léxicamente usando el proceso descrito en esta sección. También se muestra un ejemplo del proceso seguido para la recuperación de la etiqueta léxica completa suministrada por el analizador morfológico.

5.3 Evaluación del sistema de Etiquetado Léxico

En este apartado se presentan los diferentes experimentos de etiquetado diseñados sobre los corpora *WSJ*, *LexEsp* y *BDGEO* con el fin de medir las prestaciones del sistema propuesto cuando se utilizan los diferentes modelos contextuales presentados en los capítulos anteriores. La evaluación se cuantifica mediante el parámetro *precisión de etiquetado,* calculando el porcentaje de palabras bien etiquetadas respecto a un conjunto de referencia.

Además, en la sección 5.2.3 se ha presentado la utilidad del sistema descrito para realizar el etiquetado léxico de nuevos corpus, siguiendo una técnica de *'bootstrap-*

ping', mediante la que se obtiene una precisón de etiquetado mayor cuando se tienen en cuenta en la fase de aprendizaje frases de la tarea.

5.3.1 Evaluación sobre el Corpus WSJ

En los experimentos que se presentan se ha utilizado el *Penn Treebank* (1,170,000 palabras) para construir un diccionario de etiquetas de unas 49,000 entradas. En este corpus el 34% de la palabras son ambiguas, presentando una ambigüedad media de 1.47 etiquetas/palabra sobre el total o de 2.40 etiquetas/palabra sobre las ambiguas.

Se ha utilizado la parte de entrenamiento (700,000 palabras) y de prueba (100,000 palabras) definida en el capítulo anterior con todas sus particiones. A partir de estos conjuntos de entrenamiento se han aprendido los diferentes modelos de lenguaje y se han estimado las probabilidades léxicas. Aunque el conjunto de prueba se ha utilizado para la construcción del diccionario, en ningún caso se ha tenido en cuenta para el cálculo de las probabilidades léxicas.

Entrenamiento	% Precisión de Etiquetado					
#Palabras x10^3	**LEX**	**BIG**	**BDFF**	**BDL**	**ILE**	**ILC**
10	89.69	95.44	94.47	94.55	94.34	94.95
25	91.48	96.00	95.19	95.22	95.13	95.53
50	92.78	96.32	95.54	95.73	95.67	95.99
100	93.48	96.57	96.20	96.17	96.17	96.29
150	93.82	96.72	96.37	96.37	96.32	96.50
200	93.92	96.77	96.45	96.42	96.39	96.48
300	94.13	96.84	96.56	96.58	96.56	96.62
400	94.28	96.87	96.61	96.58	96.59	96.68
500	94.29	96.88	96.62	96.63	96.62	96.74
600	94.28	96.91	96.76	96.76	96.72	96.79
700	94.30	**96.91**	96.77	96.75	96.73	**96.82**

Tabla 5.3: Comparación de etiquetado léxico sobre el corpus WSJ entre un modelo BIG y un modelo ECGI con diferentes suavizados.

En la tabla 5.3 (y también de forma gráfica en la figura 5.3) se presentan los resultados de precisión obtenidos utilizando diferentes modelos contextuales. *LEX* se corresponde con un etiquetador sin modelo de lenguaje, es decir, se elige la etiqueta con mayor probabilidad léxica. Los diferentes modelos de lenguaje utilizados son: un modelo de bigramas (*BIG)* y modelos *ECGIE* con los diferentes suavizados (*BDFF, BDL, ILE, ILC).*

A la vista de los resultados se observa el siguiente comportamiento:

- Los modelos de bigramas (*BIG*) presentan una precisión mayor que la obtenida con cualquier modelo *ECGIE* para todos los conjuntos de aprendizaje ensayados. La diferencia de precisión entre *BIG* y los modelos *ECGIE* tiende a igualarse a medida que se aumenta la talla de entrenamiento (0.5 de diferencia en la primera partición frente a 0.1 en la última entre *BIG* y ILC). Para este último caso, las diferencias no son significativas considerando un nivel de confianza del 95% (*BIG:* 96.91 ± 0.11, *ILC:* 96.82 ± 0.11).

- El mejor resultado de los modelos *ECGIE* se obtiene utilizando el suavizado ILC aunque la tendencia observada indica, que cuando se incrementa la talla de entrenamiento, la precisión de etiquetado tiende a igualarse entre los diferentes suavizados.

- Si se comparan los resultados con los obtenidos considerando modelo léxico equiprobable (presentados en el capítulo 4) se observa un cambio en las prestaciones de etiquetado. Con los modelos *ECGIE* se obtenían valores de precisión superiores a *BIG*. Los mejores suavizados correspondían a *BDFF* y a *BDL*, mientras que como se puede observar, cuando se considera un modelo léxico estimado por máxima verosimilitud, el mejor resultado corresponde al suavizado ILC, y es inferior al obtenido con modelos BIG.

Pensamos que este comportamiento es debido a los siguientes factores:

- El número de parámetros a estimar en los modelos *ECGIE* es mucho mayor que para los modelos *BIG*. Este hecho hace que sólo se obtengan valores de precisión comparables a partir de una cierta talla del conjunto de aprendizaje. Así, por ejemplo, para la última partición ($700,000$ palabras) las diferencias

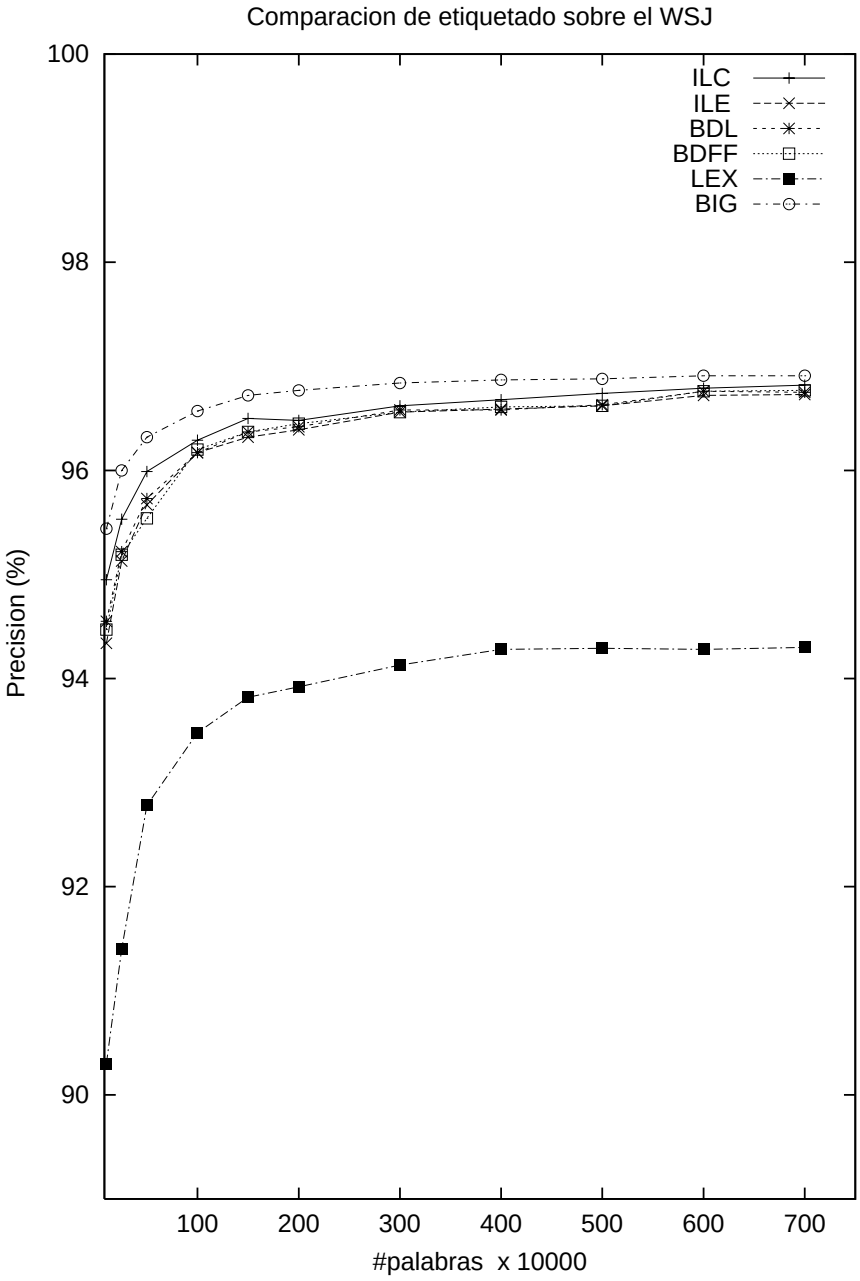

Figura 5.3: Comparación de etiquetado léxico entre modelos ECGI con diferentes suavizados y los modelos BIG y LEX.

de precisión entre *ECGIE* con suavizado *ILC* (96.82%) y *BIG* (96.91%) no son significativas considerando un intervalo de confianza del 95%. El hecho de que el suavizado *ILC* sea con el que se consiguen mejores valores de etiquetado puede ser debido a que con está técnica siempre se combinan las probabilidades del modelo contextual *ECGI* con el *BIG* lo cual puede determinar que algunas probabilidades mal estimadas en el modelo *ECGI* se vean compensadas por las del modelo *BIG*.

- Las probabilidades léxicas es otro factor que influye notablemente en el empeoramiento de la precisión de etiquetado. El cálculo de la probabilidad léxica se obtiene computando las palabras que aparecen en cada categoría. En un modelo *BIG* este cálculo es equivalente a calcular las palabras que aparecen en cada estado, puesto que hay tantos estados como categorías. En los modelos *ECGIE* se ha asumido la misma hipótesis. Se ha supuesto que todos los estados etiquetados con la misma categoría léxica tienen el mismo conjunto de probabilidades léxicas. Esto supone una simplificación motivada por la gran cantidad de muestras de aprendizaje que sería necesario utilizar para particularizar estas probabilidades a cada estado. No obstante, se debería realizar un estudio más profundo para decidir si la particularización de las probabilidades léxicas para ciertos estados (probablemente aquellos caracterizados por categorías cerradas) conllevaría un incremento de las prestaciones de etiquetado.

- También se puede deducir que en muchos casos de ambigüedad léxica, el papel que juegan las probabilidades léxicas frente a las de contexto es mucho mayor. Con el modelo *LEX* se obtiene una precisión de etiquetado de 94.30% que es superior a la obtenida considerando sólo el modelo contextual: 93.33% con modelos *ECGIE* y 92.96% con modelos *BIG*.

5.3.2 Evaluación sobre el Corpus LexEsp

Para los experimentos realizados sobre el corpus *LexEsp* se ha definido un conjunto de entrenamiento formado por 65,000 palabras (unas 3,510 frases) y uno de prueba de unas 25,000 palabras (1,335 frases) obtenido del conjunto de datos supervisados manualmente. En este conjunto de frases el 39.8% de la palabras son ambiguas, presentando una ambigüedad media de 2.6 categorías/palabra (1.6 categorías/palabra sobre el total).

La evaluación se realizará calculando la *precisión de etiquetado,* o en algunos casos mediante el *número de errores*, comparando el etiquetado obtenido con el conjunto de referencia. En el proceso de etiquetado con modelos *ECGIE* se computarán el número de transiciones de suavizado usadas (TNV) frente a las vistas (TV) en el proceso de inferencia de los modelos ECGI. Además se presentará la disminución observada en el número de errores de etiquetado en algunas de las palabras consideradas en los modelos especializados.

MÉTODO	PRECISIÓN	#ERRORES	TNV/TV
LEX	95.55%	1,119(4.45%)	–
BIG	**96.97**%	762(3.03%)	–
ILC	**96.78**%	809(3.22%)	1924(7.7%)/23204(92.3%)
ILE	96.31%	928(3.69%)	1429(5.7%)/23699(94.3%)
BDL	96.61%	852(3.39%)	1543(6.1%)/23585(93.2%)
BDFF	96.61%	852(3.39%)	1590(6.3%)/23538(93.7%)

Tabla 5.4: Resultados de etiquetado léxico sobre el corpus LexEsp con un conjunto de aprendizaje de 65,864 palabras.

En la tabla 5.4 se presentan los resultados de etiquetado con las diferentes aproximaciones propuestas. Se observa que las prestaciones del etiquetador con modelos de *BIG* son mejores que con modelos *ECGI* extendidos con cualquiera de los suavizados. Este comportamiento es similar al observado en los experimentos realizados sobre el corpus *WSJ*.

Respecto a las características de los modelos *ECGI* utilizados destacar que constan de 1481 estados. El número de transiciones de frecuencia 1 es muy elevado (35% aproximadamente del total), lo cual indica que las probabilidades de transición de los modelos *ECGI* presentan poca relevancia estadística. Pensamos que se necesitarían muchas más muestras de aprendizaje, ya que el número de parámetros a estimar es muy superior al de los modelos *BIG*.

Para el mejor resultado obtenido con modelos ECGI y suavizado *ILC* (96.78 ± 0.22) se observa una diferencia negativa respecto a los modelos *BIG* (96.97 ± 0.21) de un 0.2% (47 palabras). Esta diferencia no es estadísticamente significativa si se considera un nivel de confianza del 95%.

Además del número de transiciones de suavizado en valor absoluto (*TNV*) también se ha comprobado su distribución a nivel de frase. Se ha observado que prácticamente todas las frases utilizan alguna transición de suavizado. Por lo tanto el método de suavizado elegido influye en las prestaciones de etiquetado del sistema, sobre todo cuando el conjunto de aprendizaje es reducido. Como se ha visto en los experimentos realizados sobre el corpus WSJ esta diferencia tiende a igualarse cuando se incrementa el tamaño del conjunto de aprendizaje.

También se ha calculado cuántas frases del conjunto de prueba (secuencias de categorías) están contenidas en el conjunto de entrenamiento. Tan solo 29 frases –68 frases (5.1%) considerando repeticiones– del conjunto de prueba coinciden en el conjunto de aprendizaje. En estas condiciones, la modelización a nivel de frase que se realiza mediante el método *ECGI* no aporta grandes ventajas al problema de etiquetado, puesto que es muy poco probable que una secuencia de categorías vista en el conjunto de aprendizaje se vuelva a ver en el conjunto de prueba.

Para comprobar que los modelos *ECGI* proporcionan mejores resultados en situaciones en que el conjunto de prueba presenta estructuras (frases) que han sido vistas en la fase de entrenamiento, se ha diseñado un experimento en el que se ha incluido el conjunto de prueba en el aprendizaje (sólo para la inferencia del modelo contextual) sin que éste intervenga en la estimación de las probabilidades léxicas. Para este experimento, que llamaremos cerrado[1], se presentan los resultados en la tabla 5.5. Se observa como las prestaciones del sistema con modelos ECGIE son

[1]En realidad no se trata de un experimento cerrado, ya que para que realmente lo fuera, el conjunto de prueba y el de entrenamiento debería ser el mismo.

mucho mejores que si se utilizan modelos de bigramas. Para el mejor de los casos (ILC) existe una diferencia bastante significativa, 187 palabras, respecto a BIG, que en términos de precisión de etiquetado significa un incremento del 97.19 ± 0.20 al 97.93 ± 0.18.

MÉTODO	#ERRORES (%)	TNV/TV
BIG	708(2.81%)	—
ILC	521(2.07%)	280(1.1%)/24848(98.9%)
ILE	568(2.26%)	1087(4.3%)/24041(95.7%)
BDL	563(2.24%)	350(1.4%)/24778(98.6%)
BDFF	558(2.22%)	402(1.6%)/24726(98.4%)

Tabla 5.5: Resultados de etiquetado léxico (test cerrado) sobre el corpus LexEsp con un conjunto de aprendizaje de 90,978 palabras.

Influencia del incremento del conjunto de aprendizaje en el proceso de etiquetado (LexEsp).

En las tablas 5.6, 5.7 y 5.8 se presentan los resultados de etiquetado cuando se utilizan conjuntos de aprendizaje[2] de tamaño creciente ($400,000 - 800,000 - 1,000,000$ palabras). Se presentan los resultados obtenidos con modelos *BIG* y *ECGIE* con los suavizados que proporcionan mejores resultados (*ILC, BDFF*). Se observa que la tendencia es la misma que en los experimentos de la tabla 5.4. Los mejores resultados se obtienen para bigramas, con una diferencia que tiende a disminuir ligeramente a medida que aumenta la talla de entrenamiento. Además, los métodos *ILC* y *BDFF* tienden a igualar su precisión cuando aumenta la talla de aprendizaje.

MÉTODO	*#ERROR (%)*	*TNV/TV*
BIG	709(2.82%)	—
ILC	785(3.12%)	1141(4.5%)/23987(95.5%)
BDFF	779(3.10%)	476(1.9%)/24652(98.1%)

Tabla 5.6: Etiquetado sobre LexEsp con un conjunto de entrenamiento de 402,908 palabras

MÉTODO	*#ERROR (%)*	*TNV/TV*
BIG	704(2.80%)	—
ILC	785(3.12%)	1165(4.6%)/23963(95.4%)
BDFF	779(3.10%)	325(1.3%)/24803(98.7%)

Tabla 5.7: Etiquetado sobre LexEsp con un conjunto de entrenamiento de 823,041 palabras

MÉTODO	*#ERROR (%)*	*TNV/TV*
BIG	702(2.79%)	—
ILC	765(3.04%)	885(3.5%)/24243(96.5%)
BDFF	754(3.00%)	246(1.0%)/24882(99.0%)

Tabla 5.8: Etiquetado sobre LexEsp con un conjunto de entrenamiento de 1,077,641 palabras

[2]Estos conjuntos de entrenamiento adicionales no han sido supervisados manualmente. Se estima que tienen un 3% de error de etiquetado.

5.3.3 Evaluación sobre el Corpus BDGEO

Para evaluar el comportamiento de los modelos de bigramas (*BIG*) y los modelos *ECGIE* sobre el corpus *BDGEO* se han tenido en cuenta las particiones definidas en la tabla 5.1. Se ha elegido un conjunto de aprendizaje constituido por la partición *BD3* (84,000 palabras) y uno de prueba formado *BD1* $\cup$ *BD2* (19,000 palabras). A partir de la partición *BD3* se ha inferido un modelo BIG y modelos *ECGIE* con suavizado *BDFF* e *ILC*.

MÉTODO	*#ERROR (%)*	*TNV/TV*
BIG	182(0.91%)	—
ILC	160(0.80%)	1798(9%)/18050(91.0%)
BDFF	161(0.80%)	175(0.8%)/19673(99.2%)

Tabla 5.9: Evaluación del etiquetado léxico sobre el corpus BDGEO

Los resultados de etiquetado léxico se muestran en la tabla 5.9. Como se puede comprobar las prestaciones son muy similares, con una ligera diferencia positiva para los modelos *ECGIE*.

5.4 Etiquetado Léxico usando Modelos Especializados

En esta sección se presentan resultados de etiquetado léxico usando los modelos contextuales especializados introducidos en el capítulo 3 (sección 3.6). Los mejores resultados se obtienen con modelos especializados de bigramas (BIG_{esp}).

5.4.1 Resultados sobre el Corpus WSJ

Para el entrenamiento de los modelos se ha especializado el conjunto de entrenamiento escogiendo el mismo criterio presentado en el capítulo anterior, es decir, se han considerado las palabras del conjunto de aprendizaje con mayor frecuencia y eliminando los signos de puntuación, nombres propios, números, etc. Con esta reducción el conjunto de palabras a especializar se ha reducido a 485 (véase apéndice C.1).

Entrenamiento	% Precisión de Etiquetado	
#Palabras x10^3	BIG	BIG_{esp}
50	96.32	96.49
100	96.57	96.82
150	96.72	96.99
200	96.77	97.08
300	96.84	97.18
400	96.87	97.23
500	96.88	97.24
600	96.91	97.25
700	**96.91**	**97.30**

Tabla 5.10: Comparación de etiquetado léxico sobre el corpus WSJ entre modelos BIG y BIGesp.

En la tabla 5.10 se observa que con esta especialización, para todas la particiones, se obtiene una mejora en las prestaciones de etiquetado. Para la última partición

(700,000 palabras) la diferencia entre un modelo BIG y uno especializado BIG_{esp} es del 0.4%, por lo que se obtienen diferencias significativas (97.30 ± 0.10 frente a $96.91\% \pm 0.11$) con un intevalo de confianza del 95%.

5.4.2 Resultados sobre el Corpus LexESP

Debido a que el conjunto de entrenamiento es más reducido el criterio para elegir las palabras a especializar se ha modificado. Se han especializado las palabras más frecuentes encontradas en el conjunto de entrenamiento. De éstas, se han eliminando las pertenecientes a clases abiertas (nombre, adjetivos, verbos, etc.), además de los signos de puntuación, nombres propios, etc. Básicamente sólo se han considerado preposiciones, pronombres, relativos y determinantes. Con este criterio el conjunto de palabras se reduce a 94 palabras (véase apéndice C.2).

Método	Precisión(#errores)
BIG	96.97% (762 errores)
BIG_{esp}	97.42% (648 errores)

Tabla 5.11: Comparación de la precisión de etiquetado léxico sobre el corpus LexEsp entre modelos BIG y BIGesp.

Con la especialización propuesta se obtienen los resultados de la tabla 5.11. Se ha pasado de un 96.97% de precisión con **BIG** a un 97.42% con $\mathbf{BIG_{esp}}$. En términos absolutos supone una disminución de 114 errores, que considerando un intervalo de confianza del 95%, se obtiene una mejora significativa (97.42%±0.21% frente a 96.97%±0.21%).

También se ha realizado un estudio de la precisión de etiquetado, sobre alguna de las palabras ambiguas especializadas. Esta comparación se presenta en la tabla 5.12. Se puede observar que aparte de mejorar la precisión de etiquetado total, también se incrementa la precisión sobre las palabras que se han especializado en el modelo contextual. Las preposiciones no se incluyen puesto que, al no ser ambiguas, no presentan error de etiquetado. No obstante, al especializarlas, se resuelven favorablemente ciertos casos de ambigüedad.

	Frecuencia		**N^o Errores**		
Palabra	**Train**	**Test**	**BIG**	**BIG$_{esp}$**	**Mejora**
que	1680	709	109	83	26 (23.9%)
los	1188	371	17	11	6 (35.3%)
lo	268	140	53	13	40 (28.6%)
la	2530	869	12	7	7 (41.6%)
las	743	270	4	3	1 (25.0%)

Tabla 5.12: Mejora de etiquetado léxico obtenida sobre el corpus LexEsp usando modelos BIGesp sobre alguna de las palabras especializadas.

La especialización también se ha realizado sobre los modelos ECGIE con los diferentes suavizados sin observar en estos casos mejoras apreciables (ver tabla 5.13). Creemos que esto es debido a que el número de muestras de aprendizaje es insuficiente en relación al número tan elevado de parámetros que hay que estimar.

Suavizado	Precisión (ECGIE)	Precisión (ECGIE$_{esp}$)
ILC	96.78%	96.80%
ILE	96.31%	96.45%
BDL	96.61%	96.66%
BDFF	96.61%	96.68%

Tabla 5.13: Comparación de la precisión de etiquetado léxico sobre el corpus LexEsp usando modelos ECGIE y ECGIEesp.

5.5 Comparación Experimental de las Prestaciones de Etiquetado Léxico con otras Aproximaciones

A continuación presentamos un estudio comparativo con otras aproximaciones de etiquetado desarrolladas por otros investigadores. La contrastación se hace sobre

los mismos corpus y bajo las mismas condiciones, es decir, se consideran los mismos conjuntos de aprendizaje y prueba y las mismas restricciones.

Los trabajos con los que ha sido posible efectuar la comparación son los siguientes:

- **TT**: corresponde a un etiquetador basado en árboles de decisión (Màrquez, 1999).
- **R**: es un etiquetador basado en técnicas de relajación (Padró, 1998), que combina diferentes fuentes de conocimiento como, Bigramas (**RB)**, Trigramas (**RT)**, o los dos (**RBT**).

Comparación sobre el Corpus LexEsp

Método	**Precisión (%)**
TT	97.00%
RB	97.30%
RT	97.20%
RBT	97.40%
BIG	96.97%
BIGesp	97.42%

Tabla 5.14: Comparación de las prestaciones de etiquetado léxico de nuestro sistema con otras aproximaciones sobre el corpus LexEsp.

La comparación se establece con los resultados de nuestro sistema, presentados en la sección 5.4.1, usando modelos BIG y BIG_{esp}. En la tabla 5.14 se observa que con modelos BIG_{esp} se obtienen resultados de precisión iguales o superiores a los obtenidos con las aproximaciones comparadas.

Comparación sobre el Corpus WSJ

En este caso la comparación se realiza considerando las mismas restricciones. Es decir, se supone la existencia de un analizador morfológico perfecto que proporciona

todas las posibles categorías posibles para las palabras. Esta asunción, que se detalla en el apartado 5.1.2, es la misma que se realiza en (Màrquez, 1999). Respecto a los conjuntos de aprendizaje y de prueba remarcar que no son los mismos. En nuestro caso se utiliza una muestra de aprendizaje de 700,000 palabras y uno de prueba de 100,000 palabras, mientras que en el trabajo comparado (TT), se utilizan conjuntos cuyas tallas respectivas son 1,120,000 palabras para el aprendizaje y 50,000 palabras para el de prueba. En ambos casos, el diccionario construido para realizar el análisis morfológico es el mismo.

Método	**Precisión(%)**
TT	97.29%
BIGesp	97.30%

Tabla 5.15: Comparación de la precisión de etiquetado léxico sobre el corpus WSJ entre un modelo BIGesp y un modelo basado en árboles de decisión (TT).

Se observa (ver tabla 5.15) que las prestaciones son las mismas, a pesar de que nuestro conjunto de aprendizaje es más reducido.

5.6 Resumen

En el presente capítulo se ha descrito el sistema de etiquetado léxico de textos desarrollado. Se decribe el proceso de aprendizaje de los distintos modelos utilizados así como el proceso de etiquetado en sí.

Se han abordado diferentes tareas encaminadas a demostrar la validez del mismo sobre diferentes corpus: WSJ y LexEsp. Se ha abordado la tarea de etiquetado del corpus BDGEO siguiendo la técnica de *'bootstrapping'*. Se han realizado comparaciones entre los resultados obtenidos con los diferentes modelos contextuales considerados en esta tesis y los obtenidos con sistemas de etiquetado desarrollados por otros investigadores bajo las mismas condiciones.

Los mejores resultados se han logrado con modelos de bigramas especializados (BIG_{esp}) alcanzándose valores de precisión iguales o superiores a los presentados por

otros investigadores sobre las mismas tareas: 97.3% sobre el corpus WSJ y 97.42% sobre el corpus LexEsp.

Aunque los resultados obtenidos con modelos ECGIE no superan estos valores, se observa la habilidad que presentan estos modelos en tareas restringidas –experimentos cerrados, experimentos sobre BDGEO– para capturar una mayor información contextual. Además, se ha comprobado que en muchos casos los resultados obtenidos son complementarios, por lo que se abre una vía para obtener resultados de precisión superiores combinando diferentes modelos.

Finalmente, se debería incorporar un analizador morfológico para el inglés con el fin de poder establecer comparaciones más fiables con otros sistemas de etiquetado disponibles en la actualidad.

Capítulo 6

Análisis Sintáctico Superficial

6.1 Introducción

El análisis sintáctico superficial consiste en dividir una oración en segmentos no solapados que se corresponden con ciertas estructuras sintácticas, pero sin establecer relaciones funcionales entre las palabras que componen estas estructuras o entre las propias estructuras, es decir, sin construir el árbol sintáctico. Mediante el análisis superficial se identifican frases o sintagmas no recursivos ('*chunks*' en inglés), como por ejemplo sintagmas nominales, preposicionales, verbales, adverbiales, etc. El análisis sintáctico superficial se puede utilizar como paso previo al análisis global o bien en tareas que no requieran un análisis completo de la oración, como por ejemplo, extracción de información, generación de índices, resúmenes, etc.

En el capítulo 2 se han presentado las principales aportaciones al análisis superficial incidiendo especialmente en las que utilizan métodos de aprendizaje automático para la definición de los modelos de lenguaje. Todas estas aproximaciones tienen en común que toman como entrada oraciones etiquetadas léxicamente. Es decir, el etiquetado léxico se suele usar como un paso previo al análisis sintáctico.

En este capítulo se presenta un sistema que permite realizar de manera conjunta el etiquetado léxico y el análisis sintáctico superficial de textos mediante el uso de modelos contextuales estructurados. Mediante esta aproximación se resuelven dos problemas, en un proceso integrado, con el mismo coste computacional y obteniendo prestaciones similares. Estos modelos combinan información contextual de

categorías léxicas y unidades sintácticas, junto con las probabilidades léxicas, en un único modelo estructurado en dos niveles. En el nivel superior se representa el ML contextual de la frases en términos de categorías léxicas y de descriptores de unidades sintácticas. En el nivel inferior, se modeliza la estructura de las diferentes unidades sintácticas consideradas en el nivel superior en términos de categorías léxicas. Los modelos involucrados en cada nivel se representan mediante modelos regulares estocásticos; en concreto, se han utilizado modelos de *bigramas (BIG)* y los autómatas *ECGIE* presentados, combinándolos indistintamente en cada uno de los niveles. Además, estos modelos se pueden especializar en ambos niveles.

6.2 Aproximación Unificada al Etiquetado y Análisis Superficial

En la figura 6.1 se describe esquemáticamente nuestro sistema para el etiquetado léxico y análisis superficial de textos.

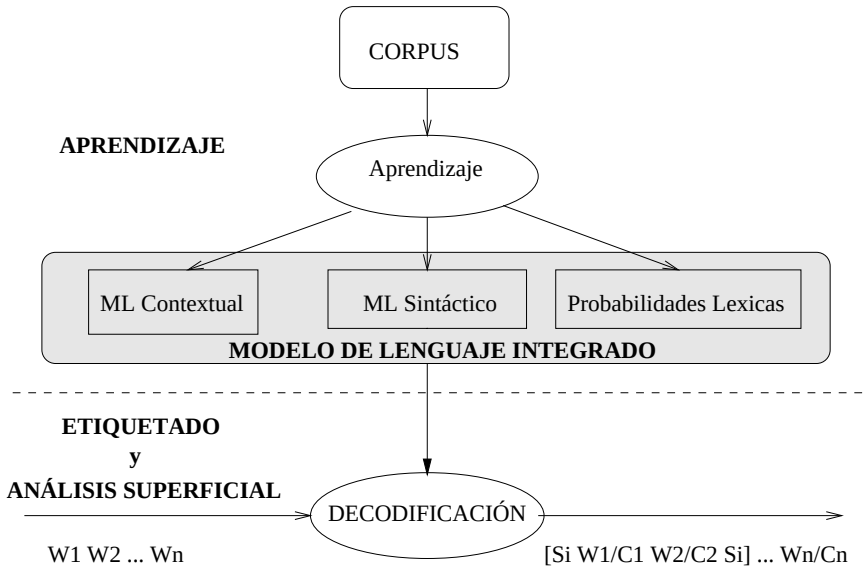

Figura 6.1: Esquema del sistema integrado de etiquetado léxico y análisis sintáctico superficial.

El sistema toma como entrada frases escritas en lenguaje natural $(W_1, W_2, ..., W_n)$

y obtiene la correspondiente secuencia de categorías léxicas y la segmentación en unidades sintácticas ($[S_i\ W_1/C_1\ W_2/C_2\ S_i]$ ... W_n/C_n), teniendo en cuenta el Modelo de Lenguaje Integrado (Pla et al., 2000a) cuyas fuentes de información son las siguientes:

- **ML Contextual**. Es un modelo de estados finitos obtenido a partir de la secuencia de categorías léxicas C_i y de descriptores de unidades sintácticas, S_i, vistas en el corpus de aprendizaje.
- **ML Sintáctico.** Es un modelo de estados finitos, similar al ML Contextual, que representa las secuencias de categorías léxicas que componen cada unidad sintáctica S_i.
- **Probabilidades Léxicas.** Se obtiene siguiendo el mismo proceso descrito en el sistema de etiquetado. A partir del conjunto de estadísticas de la frecuencia de las palabras, categorías y palabras en cada categoría obtenidas del conjunto de entrenamiento, se asignan las probabilidades léxicas para cada palabra en sus posibles categorías léxicas. Debido a que la palabra puede que no haya sido vista en el entrenamiento, o que haya sido vista sólo en algunas de las categorías posibles, se aplica un suavizado.

En la figura 6.2 se presenta más detalladamente el proceso de construcción del modelo integrado así como sus principales características. En la figura 6.2(a) se muestra un ejemplo del autómata correspondiente al ML Contextual de las oraciones. Los símbolos asociados a los estados son categorías léxicas (C_i) y descriptores de unidades sintácticas (S_i). La figura 6.2(b) muestra el modelo de la unidad sintáctica genérica (S_i) donde los símbolos asociados a los estados son ya únicamente categorías léxicas (C_i). Estos modelos deben ser convenientemente suavizados para garantizar la completa cobertura del lenguaje (obsérvese que en la figura no están suavizados).

A continuación, como se muestra en la figura 6.2(c), se realiza una sustitución regular de el/los modelo/s inferior/es en el superior, obteniendo un único modelo integrado que contempla las posibles concatenaciones de categorías léxicas y unidades sintácticas, con sus respectivas probabilidades léxicas. También se realiza un reetiquetado de los estados indicando, además de la categoría léxica asociada, a qué unidad sintáctica pertenece.

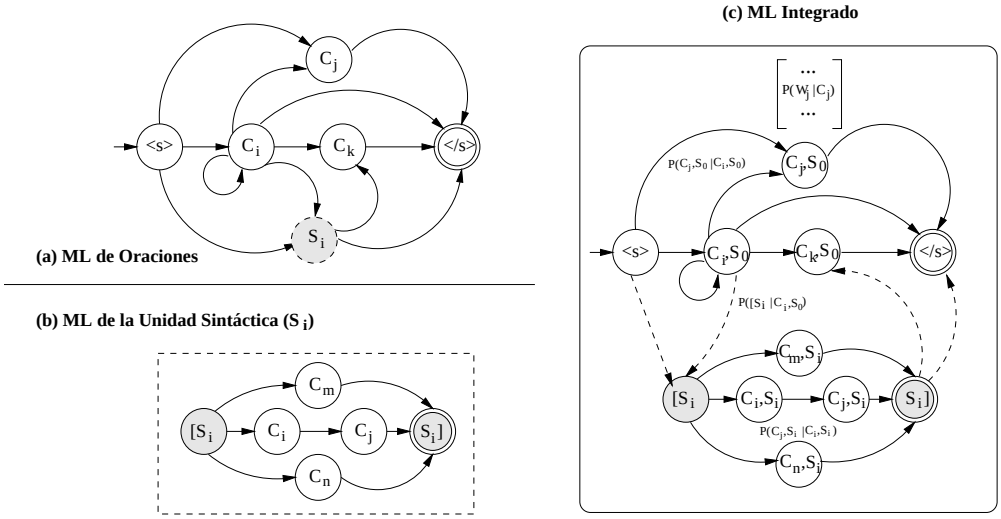

Figura 6.2: Proceso de construcción de un modelo de lenguaje integrado.

Tras el etiquetado, todo estado se caracteriza por un par $< C_i, S_i >$, donde $C_i \in \mathcal{C}$ y $S_i \in \mathcal{S}$. Los conjuntos $\mathcal{C}$ y $\mathcal{S}$ se definen como:

- $\mathcal{C} = \{C_1, ..., C_N\}$ el conjunto de etiquetas léxicas considerado.
- $\mathcal{S} = \{[S_i, S_i], S_i, S_0\}$ el conjunto de etiquetas sintácticas utilizado. $[S_i$ y $S_i]$ representan, respectivamente, el estado inicial y final de la unidad sintáctica de descriptor S_i. La etiqueta S_i indica que un estado pertenece a la unidad sintáctica S_i. Por el contrario, S_0 indica que el estado se encuentra fuera de cualquier unidad sintáctica.

Con este etiquetado se permite caracterizar las propiedades de un estado sin perder la información de la estructura a la que pertenece. Así, por ejemplo, podemos distinguir la probabilidad de la secuencia de categorías C_iC_j, cuando nos encontramos dentro de la unidad sintáctica o fuera de ella, ya que éstas son distintas $P(< C_jS_0 > | < C_i, S_0 >) \neq P(< C_j, S_i > | < C_i, S_i >), \forall S_i \in \mathcal{S}$.

Las probabilidades léxicas también se podrían particularizar para los estados que se encuentren fuera o dentro de una unidad sintáctica. Esto evidentemente requeriría una mayor cantidad de datos de aprendizaje, por lo que la solución que se ha adoptado consiste en considerar la misma distribución de probabilidad léxica

para todo estado que venga caracterizado por la misma categoría léxica, es decir, asumir la siguiente aproximación $P(W_i| < S_i, C_i >) \approx P(W_i|C_i)\ \forall S_i \in \mathcal{S}$.

6.3 Formulación Probabilística del Problema

El problema integrado del etiquetado léxico y del análisis sintáctico superficial consiste en encontrar, para una frase de entrada ($W = W_1, ..., W_T$), la mejor secuencia de categorías léxicas ($C = C_1, ..., C_T$) y la mejor segmentación en unidades sintácticas ($S = S_1, ..., S_K; K \leq T$), teniendo en cuenta el modelo integrado definido.

$$\hat{S}\ \hat{C} = \arg\max_{S,C} P(S, C|W) \tag{6.1}$$

La resolución de la ecuación 6.1 se puede plantear como un problema de etiquetado doble si, como se ha formulado anteriormente, se exige que a todas las palabras se les asigne una etiqueta léxica y una sintáctica, con lo que en este caso el número de etiquetas léxicas y sintácticas para una frase coincide ($K = T$).

Si se aplica la regla de Bayes, y se tiene en cuenta que la maximización es independiente de la secuencia de palabras W considerada, se obtiene:

$$\begin{aligned}\hat{S}\ \hat{C} &= \arg\max_{S,C} P(S, C|W) = \arg\max_{S,C} \frac{P(W|S, C) \cdot P(S, C)}{P(W)} \\ &= \arg\max_{S,C} P(W|S, C) \cdot P(S, C)\end{aligned}$$

donde $P(S, C)$ representa el Modelo Contextual de Lenguaje y $P(W|S, C)$ el Modelo Léxico.

Asumiendo las simplificaciones usuales en un modelo de Markov, presentadas en el capítulo 2, y particularizando para el caso de bigramas, se obtiene:

- Modelo de Lenguaje

$$P(S, C) = P(S_1, ..., S_T, C_1, ..., C_T) \approx \prod_{i=1..T} P(< S_i, C_i > | < S_{i-1}, C_{i-1} >)$$

- Modelo Léxico

$$P(W|S,C) = \prod_{i=1...T} P(W_i| < S_i, C_i >) \approx \prod_{i=1...T} P(W_i|C_i)$$

Con estas simplificaciones el problema se traduce en resolver:

$$\hat{S}\ \hat{C} = \arg\max_{S,C} P(S,C|W) = \arg\max_{S,C} \prod_{i=1...T} P(< S_i, C_i > | < S_{i-1}, C_{i-1} >) \cdot P(W_i|C_i) \tag{6.2}$$

6.4 Proceso de Decodificación: Etiquetado y Análisis Superficial

El proceso de etiquetado y análisis conjunto (ecuación 6.2) se puede llevar a cabo mediante el algoritmo de Viterbi presentado en el capítulo 3, el cual debe ser convenientemente modificado para su aplicación sobre los modelos integrados definidos. A continuación se presenta un ejemplo en el que se ilustra la modificación adoptada.

En la figura 6.3 se muestra parcialmente el '*trellis*' de programación dinámica asociado a una frase de entrada $W_1, W_2, ..., W_n$ cuando se utiliza el modelo integrado de la Figura 6.2(c). Mediante un círculo negro se resaltan aquellos estados a los que se puede transitar desde la etapa anterior de programación dinámica, y que son compatibles con las restricciones impuestas por el modelo léxico. Así, por ejemplo, se puede transitar desde el estado caracterizado por la categoría $< C_i, S_0 >$ al estado cuya categoría es $< C_k, S_0 >$, con la palabra W_2, porque está permitido por el ML integrado y la probabilidad léxica $P(W_2|C_k)$ es distinta de cero.

También se permiten transiciones a estados que son principios o finales de unidades sintácticas. Este tipo de transiciones, que llamaremos ε, no consumen símbolos de la frase de entrada y su tratamiento requiere una modificación del algoritmo de Viterbi. Se ha representado mediante un círculo en blanco los estados a los cuales llegan transiciones ε. Cuando se da esta situación, dentro de la misma etapa de programación dinámica, se consideran transiciones a los sucesores de estos estados (transiciones en trazo discontinuo), donde ahora si que se debe consumir símbolo.

En la figura 6.4 se muestra el pseudocódigo correspondiente a la modificación del algoritmo de Viterbi presentado en el capítulo 3. El proceso que se sigue es

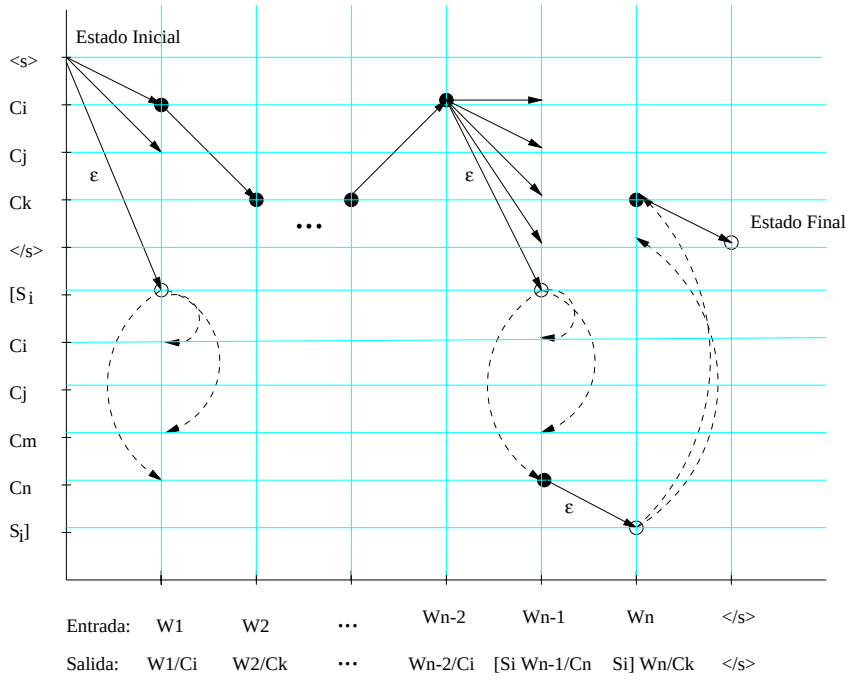

Figura 6.3: Trellis parcial de la programación dinámica para la frase W usando el modelo integrado de la figura 6.2 (c).

esencialmente el mismo, pero se debe tener en cuenta el conjunto $\mathcal{S}_{IoF}$, correspondiente a los estados que son inicios o finales de autómatas de unidades sintácticas ($\mathcal{S}_{IoF} = \{[S_i, S_i]\}$ en la Figura 6.3). Durante el proceso de programación dinámica, cuando encontramos transiciones a estados pertenecientes a $\mathcal{S}_{IoF}$, se expanden nuevas transiciones hacia los sucesores del estado, siempre dentro de la misma etapa de programación dinámica.

Una vez construido el 'trellis' de programación dinámica, haciendo vuelta atrás y deshaciendo el camino de máxima probabilidad, se obtiene la mejor secuencia de estados correspondiente a la frase de entrada. Como cada estado lleva asociado un par $< C_i, S_0 >$, disponemos del mejor etiquetado léxico, e interpretando las etiquetas sintácticas, la mejor segmentación en unidades sintácticas de la frase.

En (Ramshaw and Marcus, 1995), como se ha presentado en el capítulo 2, se utiliza un conjunto de etiquetas sintácticas, que se asocian a las palabras y que permiten obtener la segmentación de la frase en sus constituyentes sintácticos. Este conjunto está formado por $IOB1 = \{I\_S_i, B\_S_i, O\}$, donde I_S_i denota que la

```
/* Modificación del Algoritmo de Viterbi para contemplar transiciones ε. */
S_IoF = {[S_i, S_i]} : conjunto de estados iniciales o finales de un subautómata correspondiente a
cualquier unidad sintáctica.
para i := 1 hasta nactiv1 hacer
    est_anterior := Activos1[i];
    para j ∈ Sucesores[est_anterior] hacer
      si j ∈ Estados_IoF entonces
        est_anterior = j;
        para jj ∈ Sucesores[est_anterior] hacer
          si (P(W_t|jj) ≠ 0) entonces
            Coste := Actual[est_anterior] * P(jj|est_anterior) * P(W_t|jj);
            si (YaAlcanzado[jj] < t) entonces
              Actual[jj] := Coste;
              YaAlcanzado[jj] := t;
              nactiv2 := nactiv2 + 1;
              Activos2[nactiv2] := jj;
              apunta[jj][t] := est_anterior;
            sino
              si (Coste > Actual[jj]) entonces
                Actual[jj] := Coste;
                apunta[jj][t] := est_anterior;
              finsi
            finsi
          finsi
        finpara
      finsi
    finpara
finpara
```

Figura 6.4: Modificación del algoritmo de Viterbi para contemplar transiciones ε

palabra es principio o está dentro de la unidad sintáctica S_i, B_S_i indica principio de S_i siendo la palabra anterior final de S_i, y por último O indica que está fuera de cualquier unidad sintáctica. El cambio de nuestro sistema de etiquetas $\mathcal{S} = \{[S_i, S_i], S_i, S_0\}$ al conjunto $IOB1$ se puede hacer de manera inmediata.

6.5 Evaluación del Sistema Integrado

En el capítulo 2 se han presentado las principales aproximaciones inductivas al análisis sintáctico superficial, así como, las prestaciones obtenidas en términos de precisión y cobertura. Estas aproximaciones utilizan todo o parte del corpus *WSJ* como base de aprendizaje y evaluación. La tarea de detección de sintagmas nominales no recursivos y no solapados (*baseNP* en inglés) constituye una de las subtareas más ampliamente estudiadas en este campo. La contrastación experimental más rigurosa entre los diferentes métodos es la definida por (Ramshaw and Marcus, 1995), ya que se utiliza el mismo conjunto de aprendizaje y de prueba y la misma definición de NP.

A continuación se presenta la evaluación experimental realizada con el sistema integrado de etiquetado léxico y análisis sintáctico superficial. Las tareas abordadas son las siguientes:

- Detección de sintagmas nominales no recursivos y no solapados (baseNP en inglés). Se ha utilizado el mismo conjunto de datos del WSJ presentado en la sección 5.3.1 para los experimentos de etiquetado léxico. En esta tarea se utilizan y comparan diferentes combinaciones de modelos de estados finitos para construir el modelo de lenguaje integrado.

- Detección de un conjunto de unidades sintácticas más extenso. A parte de la detección de NP, se construye un modelo integrado que es capaz de detectar otras unidades (PP, SV, ADJP, ...). La contrastación de los resultados se ha realizado utilizando un conjunto de datos de entrenamiento y de prueba extraído del corpus WSJ, perteneciente a una tarea compartida definida sobre el conjunto de datos usado en (Ramshaw and Marcus, 1995). Este conjunto ha sido segmentado en una serie de sintagmas y sobre éstos, se ha propues-

to una tarea compartida que se puede consultar en la dirección *http://lcg-www.uia.ac.be/conll2000/chunking/* .

- Detección de SN para el castellano sobre el corpus LexEsp. Se ha utilizado las mismas particiones usadas en la evaluación del sistema de etiquetado del capítulo 5. A partir de éstas se ha usado el analizador parcial de oraciones *APOLN* (Molina et al., 1999a; Molina et al., 1999b) para realizar la segmentación en SN. Estos datos, sin realizar ninguna supervisión manual, se han utilizado como conjunto de entrenamiento y de prueba de la tarea.

La evaluación experimental se realiza mediante las medidas presentadas en el capítulo 2: *precisión(P)*, *cobertura(C)* y $F_{\beta=1}$.

- $P = \frac{\#\, constituyentes\, correctos\, en\, el\, análisis\, propuesto}{\#\, constituyentes\, en\, el\, análisis\, propuesto}$
- $C = \frac{\#\, constituyentes\, correctos\, en\, el\, análisis\, propuesto}{\#\, constituyentes\, en\, el\, análisis\, de\, referencia}$
- $F_{\beta=1} = \frac{2 \times P \times C}{P + C}$

6.6 Detección de NP sobre el WSJ

Muchos son los trabajos (ver capítulo 2) que abordan la tarea de detección de sintagmas nominales no recursivos (*baseNP*) tomando todo o parte del corpus WSJ como base de aprendizaje y evaluación.

En esta sección se presenta un conjunto de experimentos que utilizan las mismas particiones del corpus WSJ (800,000 palabras) usadas en la evaluación del sistema de etiquetado presentado en el capítulo 5 (700,000 palabras para entrenamiento y 100,000 palabras para evaluación). Para esta tarea, a parte de la información del etiquetado léxico del texto, se ha utilizado la segmentación por marcas de NP del mismo.

Queremos destacar que la definición de los NP utilizada es la definida por (Church, 1988) y que también ha sido utilizada en (Ejerhed, 1988). Esta definición, como se apunta en (Ramshaw and Marcus, 1995), es bastante simple para

algunos casos. Por ejemplo, ciertos NP coordinados por conjunciones (*y*, *o*) o por *comas*, son considerados como NP distintos, en vez de como un único NP. No obstante, el principal objetivo de los experimentos que se muestran es evaluar el sistema propuesto y comprobar la capacidad inductiva del mismo en el aprendizaje de unidades sintácticas. Posteriormente, la evaluación se realizará sobre los *NP* definidos en (Ramshaw and Marcus, 1995), que en algunos caso es más compleja.

Los modelos integrados que se utilizan se pueden construir eligiendo cualquiera de los autómatas de estados finitos presentados en este trabajo, BIG y ECGIE (con diversos suavizados), indistintamente en cada uno de los niveles. Los resultados que se presentan consideran diferentes combinaciones de estos modelos. La notación empleada indica los modelos utilizados para representar el ML de las oraciones y el ML de las unidades sintácticas: BIG-BIG, BIG-EGGI, ECGI-BIG y ECGI-ECGI. Por ejemplo BIG-ECGI significa que el ML integrado utiliza en el nivel superior (ML contextual) un modelo de bigramas (BIG) y en el inferior (ML para un NP) un modelo ECGI. Para todos los modelos ECGI se ha elegido el suavizado ILC.

Respecto al análisis morfológico de las frases de entrada se sigue la misma aproximación presentada en el capítulo 5. Es decir, se construye un diccionario de categorías a partir del conjunto de entrenamiento y de prueba, pero sin utilizar en ningún caso el conjunto de prueba para estimar las probabilidades léxicas.

6.6.1 Integración de Modelos de Bigramas (BIG)

Siguiendo el esquema de aprendizaje presentado en la sección 6.3 se ha aprendido un ML contextual de las frases y un ML para los NP. Ambos se han formalizado como bigramas suavizados utilizando la técnica de *Back-off* y se han integrado en un único modelo BIG-BIG. Con estos modelos se obtienen los resultados que se detallan a continuación (Pla et al., 2000a).

En la figura 6.5 se muestran los resultados de etiquetado léxico utilizando conjuntos de entrenamiento de diferente talla y tres aproximaciones distintas para el etiquetado. La más simple (*LEX*) consiste en un proceso de etiquetado que no tiene en cuenta información contextual (sin ML); en este caso, la etiqueta léxica asociada a cada palabra será la de mayor frecuencia observada en el conjunto de entrenamiento. El segundo método corresponde a un etiquetador basado en un modelo de bigramas

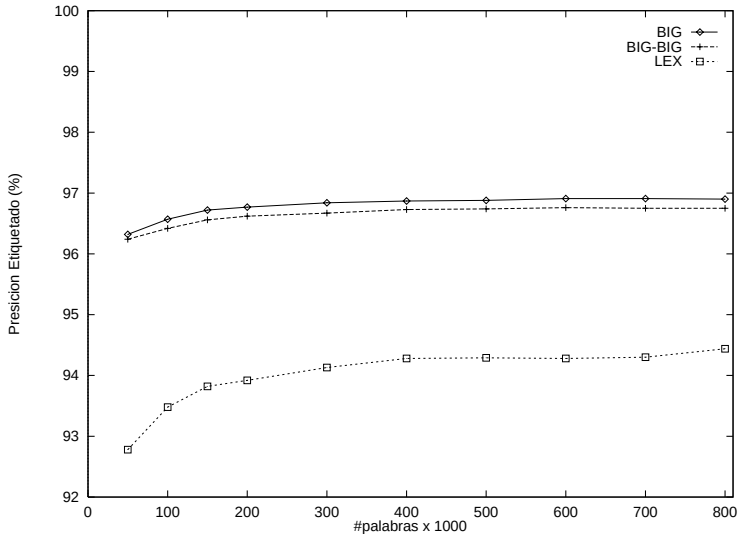

Figura 6.5: Evolución de la precisión de etiquetado usando modelos BIG, LEX y BIG-BIG.

(*BIG*). Por último, se usa un ML integrado (*BIG-BIG*) cuyas características se han descrito anteriormente.

La precisión de etiquetado para los métodos *BIG* y *BIG-BIG* son similares, 96.9% y 96.8% respectivamente, sin embargo si no se usa modelo de lenguaje (*LEX*), la precisión es 2.5 menor (94.3%). La tendencia observada en todos los casos es que un incremento en la talla de entrenamiento redunda en un incremento de la precisión de etiquetado. A partir de $300,000$ palabras de entrenamiento, los resultados tienden a estabilizarse.

En la figura 6.6, se muestra la precisión y cobertura obtenidas en la detección de NP. Los resultados con modelos BIG-BIG son satisfactorios, obteniéndose una precisión de 94.6% y una cobertura del 93.6% con el máximo número de muestras de entrenamiento. También se puede observar cómo los valores de estos parámetros se incrementan con el aumento de la talla de entrenamiento.

También se han realizado experimentos considerando modelos integrados de bigramas especializados. Para cada una de las particiones consideradas, se han obtenido prestaciones superiores a los no especializados. En la tabla 6.1 se comparan los resultados obtenidos para la última partición utilizando modelos integrados de bigramas (BIG) con y sin especialización.

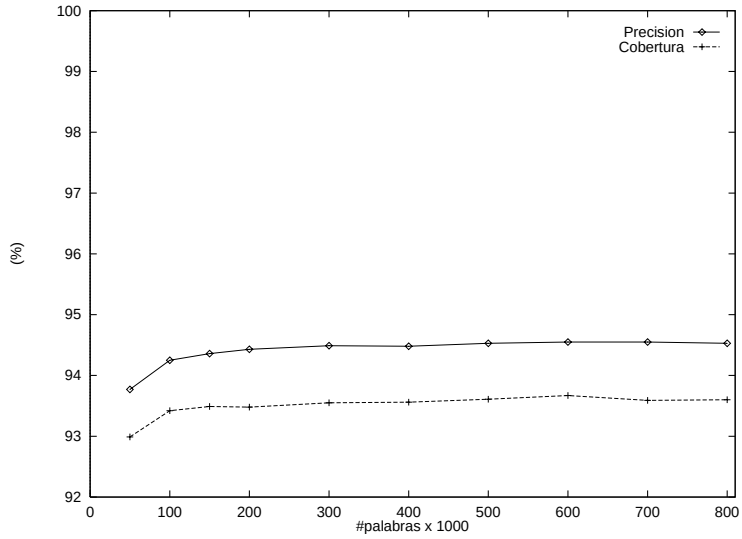

Figura 6.6: Evolución de la precisión y la cobertura en la detección de NP usando modelos BIG-BIG.

Proceso Integrado	Etiquetado	Detección de NP	
	Precisión	Precisión	Cobertura
BIG-BIG	96.8%	94.6%	93.6%
BIG_{esp}-BIG_{esp}	97.1%	94.9%	94.0%

Tabla 6.1: Resultados de etiquetado léxico y detección de NP obtenidos mediante el proceso integrado usando modelos BIG-BIG, con y sin especialización, y un conjunto de entrenamiento de 700,000 palabras.

Los modelos integrados también se puede utilizar para realizar únicamente la tarea de detección de NP. En ese sentido se debe seguir un proceso secuencial. Esto es, primero se realiza un etiquetado de la frase de entrada y a continuación, se usa el modelo integrado (por ejemplo BIG-BIG) para realizar la detección de NP. En este caso, el proceso consiste en encontrar la mejor secuencia de estados en el modelo para una frase de entrada que ya está etiquetada. Por lo tanto, puesto que la desambiguación léxica ya se ha realizado, en el proceso sólo se tienen en cuenta las probabilidades contextuales del modelo integrado, sin considerar las probabilidades léxicas.

En la tabla 6.2 se muestran los resultados cuando se realiza la detección de NP de manera secuencial utilizando diferentes etiquetadores.

- LEX: sólo se tienen en cuenta las probabilidades léxicas. En este caso la precisión de etiquetado es del 94.3%.
- BIG: se basa en un modelo de bigramas con una precisión del 96.9%.
- BIG_{esp}: se basa en un modelo de bigramas especializados con una precisión del 97.3%.
- IDEAL: se simula un etiquetador con precisión 100%. Para hacer esto, se usan las frases etiquetas tomadas directamente del corpus *WSJ*.

Proceso Secuencial				
Modelo Integrado	Etiquetador Léxico	Etiquetado	Detección de NP	
		Precisión	Precisión	Cobertura
BIG-BIG	LEX	94.3	90.8	91.3
	BIG	96.9	94.9	94.1
	IDEAL	100	95.5	94.7
BIG_{esp}-BIG_{esp}	LEX	94.3	91.4	92.0
	BIG_{esp}	97.3	95.1	94.4
	IDEAL	100	95.7	94.9

Tabla 6.2: Resultados de etiquetado léxico y detección de NP obtenidos mediante un proceso secuencial usando diferentes etiquetadores y modelos integrados usando un conjunto de entrenamiento de 700,000 palabras.

Estos resultados confirman que la precisión y cobertura en la detección de NP tiene una relación directa con la precisión de etiquetado, cuando ésta aumenta (BIG, BIG_{esp} y IDEAL), también se incrementa la detección de NP.

La precisión de etiquetado del proceso secuencial (usando el etiquetador BIG) es ligeramente superior a la obtenida mediante el proceso integrado (BIG-BIG),

$96.9\% \pm 0.1$ frente a $96.8\% \pm 0.1$. Aunque la diferencia no es significativa considerando un intervalo de confianza al 95%, no deja de ser sorprendente que la mayor información que se recoge en un modelo integrado, no repercuta en unos resultados mejores de etiquetado. Esta diferencia es todavía mayor entre los resultados obtenidos con BIG_{esp} ($97.3\% \pm 0.1\%$) y BIG_{esp}-BIG_{esp} ($97.1\% \pm 0.1\%$).

Pensamos que esto puede ser debido básicamente a las siguientes razones:

- El modelo contextual contempla un número mayor de parámetros a estimar. Al estimar un número de parámetros mayor utilizando el mismo número de muestras de aprendizaje, el problema de la insuficiencia de datos se ve más agravado.

- Las probabilidades léxicas no se particularizan a los estados. Se ha considerado la misma distribución de probabilidad léxica para las palabras, independientemente de su aparición dentro o fuera de una unidad sintáctica. La particularización de éstas dentro de cada unidad quizá mejoraría los resultados. Sin embargo, esta asunción necesitaría más datos para su correcta estimación.

- Combinación de las probabilidades de los diferentes modelos involucrados en el ML integrado. Se podría estudiar métodos alternativos de combinación de los mismos. Así, por ejemplo, se podrían establecer ciertos pesos que primaran unos modelos frente a los otros.

6.6.2 Integración de Modelos ECGI y BIG

En la tabla 6.3, se muestran los resultados de etiquetado y detección de NP para todas las combinaciones de modelos ECGI con BIG (Pla et al., 2000b).

Se observa que de todas las combinaciones mostradas, los mejores resultados de etiquetado léxico y detección de NP se obtienen con modelos BIG-BIG. Se han realizado experimentos adicionales tendentes a determinar qué diferencias de etiquetado existen entre estas combinaciones. En ese sentido, se ha estimado que la diferencia es de un 2% a un 3%, incluso entre aquellas combinaciones con las que se obtiene la misma precisión de etiquetado. Este hecho nos hace pensar en la posibilidad de hacer un estudio más profundo para determinar en qué situaciones (si es que se

pudieran caracterizar) un modelo se comporta mejor que otro, y así poder combinar esta información para obtener prestaciones mayores. Está misma técnica, en caso de que fuera fructífera, se podría extender para mejorar también la detección de NP, como por ejemplo se hace en el trabajo de (Tjong-Kim-Sang, 2000b) en el que se combinan diferentes analizadores para aumentar la precisión y la cobertura.

Método	Precisión NP	Cobertura NP	Precisión Etiq.
BIG-BIG	94.6%	93.6%	96.8%
BIG-ECGI	93.8%	93.1%	96.7%
ECGI-ECGI	93.2%	91.4%	96.6%
ECGI-BIG	93.8%	91.8%	96.6%

Tabla 6.3: Resultados de etiquetado léxico y detección de NP sobre el corpus WSJ usando diferentes combinaciones de modelos contextuales.

6.7 Detección de Unidades Sintácticas sobre WSJ

Los experimentos presentados hasta el momento han tenido por objetivo principal mostrar la capacidad de los modelos integrados para realizar la tarea conjunta de etiquetado y análisis sintáctico. Para ello, se ha elegido una tarea (detección de NP), que aunque es de suma importancia, no deja de ser incompleta debido a que no se consideran otros constituyentes de la oración (PP, VP, PRT, ..., etc).

En ese sentido, algunos autores han extendido sus trabajos de detección de NP a otras unidades. Un intento de evaluar diferentes aproximaciones inductivas sobre un conjunto extenso de unidades sintácticas y sobre un mismo conjunto de datos, lo constituye el Workshop *CoNLL-2000* (*http://lcg-www.uia.ac.be/conll2000/*) en el que se plantea una tarea compartida consistente en obtener la segmentación de frases, en un número superior de segmentos. Un ejemplo de frase de la tarea se muestra a continuación.

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow] [PP to] [NP only # 1.8 billion] [PP in] [NP September] .

6.7.1 Descripción de la Tarea

Se ha escogido un conjunto de datos extraído del corpus WSJ que coincide con el utilizado en la tarea de detección de NP definida en (Ramshaw and Marcus, 1995). Consta de las secciones 15-18 del WSJ como conjunto de aprendizaje (211,727 palabras) y la sección 20 como conjunto de prueba (47,377 palabras). Las unidades sintácticas no solapadas o *chunks* de estos textos, se han obtenido a partir del corpus WSJ mediante un programa escrito por *Sabine Buchholz* de la Universidad de Tilburg (The Netherlands) que es capaz de derivar el análisis superficial a partir del análisis total de las oraciones (ver el ejemplo que se presenta en el apéndice A). Además, para simular una situación más realista, en la que normalmente un analizador sintáctico toma como entrada la salida de un etiquetador, el etiquetado léxico de las frases se ha obtenido con el etiquetador de Brill (Brill, 1994), sin realizar ninguna supervisión, por lo que se observan ciertos errores de etiquetado (aproximadamente del 3% respecto al WSJ).

La tarea consiste en obtener la segmentación, en todas las unidades sintácticas propuestas (ADJP, ADVP, CONJP, INTJ, NP, PP, PRT, SBAR, VP), mediante una aproximación inductiva que sólo utilice el conjunto de aprendizaje propuesto. La evaluación se debe realizar en términos del factor $F_{\beta=1}$. Una descripción de la tarea y de las características de las unidades sintácticas consideradas se puede encontrar en (Tjong-Kim-Sang and Buchholz, 2000).

6.7.2 Características de las Unidades Sintácticas

Los tipos de unidades lingüísticas utilizadas en la tarea se basan en parte de las categorías sintácticas definidas en el *WSJ*. En ciertos casos, existen dificultades para convertir la notación de árbol, utilizada en el *WSJ*, a unidades sintácticas no solapadas (*chunk*). Por eso, se han tenido que realizar ciertas simplificaciones, que se enuncian brevemente, para cada una de las unidades consideradas.

NP

La definición de los NP es muy similar a la utilizada en (Ramshaw and Marcus, 1995). Algunas asunciones tomadas son, por ejemplo:

- Dividir las construcciones de NP posesivas en dos (*[$_{NP}$ Eastern Airlines] [$_{NP}$ ' creditors]*)
- Los constituyentes ADJP dentro de NP se toman como parte de este último (*(NP The (ADJP most volatile) form* $\rightarrow$ *[$_{NP}$ The most volatile form]*).

VP

- En la notación del WSJ encontramos ciertos casos de anidamiento de sintagmas verbales, que en nuestro caso, se toman como uno solo: *((S (NP-SBJ-3 Mr. Icahn) (VP may not (VP want (S (NP-SBJ*-3) (VP to (VP sell ...))))) .))* $\rightarrow$ *[$_{NP}$ Mr. Icahn] [$_{VP}$ may not want to sell]* ... Esto no impide que en algunos casos tengamos VP consecutivos: *[$_{NP}$ The impression] [$_{NP}$ I] [$_{VP}$ have got] [$_{VP}$ is] [they]* ...
- Adverbios y frases adverbiales permanecen como parte del VP: *(VP could (ADVP very well) (VP show ...))* $\rightarrow$ *[$_{VP}$ could very well show]* ... Sin embargo, los adjetivos predicativos de los verbos no forman parte del VP: *[$_{NP}$ they] [$_{VP}$ are] [$_{ADJP}$ unhappy]* ...
- En frases invertidas, el verbo auxiliar no forma parte de un VP en el WSJ. En consecuencia, este caso, no aparece dentro de ningún VP: *((S (SINV (CONJP Not only) does (NP-SBJ-1 your product) (VP have (S (NP-SBJ *-1) (VP be (ADJP-PRD excellent))))))), but ...* $\rightarrow$ *[$_{CONJP}$ Not only] does [$_{NP}$ your product] [$_{VP}$ have to be] [$_{ADJP}$ ecellent] , but ...*

ADVP y ADJP

La mayor parte de los ADVP se corresponden con la definición del WSJ. Sin embargo, ciertos ADVP, dentro de ADJP o dentro de VP, se toman como parte de estos constituyentes.

- ADVP que contienen NP se desdoblan en dos: *(ADJP-TMP (NP a year) earlier)* $\rightarrow$ *[$_{NP}$ a year][$_{ADVP}$ earlier]*
- ADJP que contienen NP también se desdoblan en dos: *(ADJP-PRD (NP 68 years) old)* $\rightarrow$ *[$_{NP}$ 68 years][$_{ADVP}$ old]*

PP y SBAR

- La mayor parte de los PP están formados por sólo una palabra (una preposición cuya etiqueta léxica es IN). Esto no significa que sea una tarea trivial, puesto que con esta categoría, aparecen otros constituyentes (SBAR) y además, ciertos PP, se componen de más de una palabra, como por ejemplo: *such as*, *because of*, *due to*, *even in*, etc.

- Los SBAR principalmente están formados por una palabra con la categoría IN, pero, también pueden incluir preposiciones de más de una palabra, por ejemplo: *so that*, *as if*, *only if*, etc.

CONJP, PRT, INTJ, LST, UCP

- Las conjunciones pueden estar formadas por más de una palabra como: *as well as*, *instead of*, *but also,* ... Algunas palabras, que son conjunciones (*and*, *or*) no están anotadas como CONJP en el WSJ, por lo que estos *chunks* no se han considerado en la tarea.

- En el WSJ se usa la etiqueta PRT para marcar los participios de los verbos y esta es la definición que se ha usado en la tarea. Aunque en principio son fáciles de reconocer, puesto que llevan asociada la etiqueta léxica RP, ciertos errores de etiquetado léxico (aparecen con etiquetas IN y RB), dificultan su detección.

- INTJ representa interjecciones como *oh*, *hello*, *good grief!*, ...

- LST y UCP son unidades poco frecuentes. La primera representa marcas de listas (*1.*, *2.*, *a)*, *b)*, ...) mientras que la segunda se corresponde con unidades poco comunes caracterizadas por conjunciones como *and* y *or*.

6.7.3 Evaluación Experimental

Para resolver esta tarea (Pla et al., 2000c) se ha utilizado la aproximación propuesta en este capítulo. Se ha obtenido un ML integrado (BIG-BIG), que contempla todas las unidades sintácticas propuestas. Se ha construido un diccionario de etiquetas (a

partir del conjunto entrenamiento y de prueba) para simular un analizador morfológico, al igual que se hizo en la tarea de detección de NP y en los experimentos de etiquetado léxico sobre el corpus WSJ.

	precision	cobertura	$F_{\beta=1}$
ADJP	65.53 %	63.01%	64.26
ADVP	74.61%	71.25%	72.89
CONJP	0.00%	0.00%	0.00
INTJ	50.00%	100.00%	66.67
NP	89.08%	87.38%	88.23
PP	83.92%	94.60%	88.94
PRT	51.56%	31.13%	38.82
SBAR	25.00%	0.19%	0.37
VP	91.01%	90.0%	90.50
TOTAL	87.22%	86.06%	86.64

Tabla 6.4: Resultados con modelos integrados sin especialización (Precisión de etiquetado sobre etiquetas IOB1 = 91.87%)

Una vez aprendido el modelo se ha utilizado para etiquetar léxicamente y detectar las unidades sintácticas del conjunto de prueba en un proceso conjunto. Los resultados obtenidos se presentan en la tabla 6.4.

A la vista de los resultados se observa el siguiente comportamiento:

- La precisión y cobertura en la detección de NP ha bajado de manera considerable, si se compara con los experimentos mostrados en la sección 6.6. La precisión desciende de 94.6% a 89.8% y la cobertura de 93.6% a 87.38%. Pensamos que esto es debido a varios factores:
  - El número de muestras de aprendizaje es mucho menor (700,000 frente a 200,000 palabras).
  - Al considerar más unidades sintácticas, el modelo integrado está peor estimado. Además, como ya se ha comentado, la definición de los NP es más compleja.

  - En los experimentos de la sección 6.6 el conjunto de aprendizaje no presentaba errores de etiquetado (extraído del etiquetado del WSJ), mientras que el etiquetado de la tarea es la salida del etiquetador de Brill.

- Para ciertas unidades CONJP, INTJ, PRT, SBAR, PP se obtienen prestaciones muy bajas. Los factores que influyen son:

  - El número de muestras para las unidades sintácticas CONJP, INTJ es muy pequeño.
  - Las unidades PRT, SBAR, PP vienen caracterizadas por las mismas categorías léxicas (básicamente la categoría IN), lo cual hace que las restriciones contextuales sean difíciles de captar.

Como alternativa a algunos de los problemas detectados en la tarea se propone utilizar los modelos BIG_{esp} con el objetivo de estudiar si la particularización de ciertas palabras en el modelo contextual conlleva un aumento de la prestaciones.

Para ello se define una función de especialización del conjunto de entrenamiento que particularice ciertas palabras (en sus categorías léxicas) dentro del modelo contextual.

Definimos el conjunto de la palabras a especializar ($\mathcal{W}_e$) como el conjunto de aquellas pertenecientes al conjunto de entrenamiento ($\mathcal{T}$) cuya frecuencia sea mayor que un determinado umbral U.

$$\mathcal{W}_e = \{w_i \in \mathcal{T} : f(w_i) > U\}$$

En la figura 6.7 se observa la evolución del factor F_β, cuando se utiliza para la tarea modelos BIG_{esp}-BIG_{esp} en función del número de palabras consideradas en el modelo contextual. El valor del umbral U se ha tomado en un rango que va de 2,000 hasta 80, considerando los intervalos que se muestran en la figura. Los mejores resultados se obtiene con $U = 80$, que se traduce en una especialización de unas 450 palabras. Para valores de U inferiores a 80 las prestaciones disminuyen.

Destacar que se han eliminado del conjunto $\mathcal{W}_e$ ciertas palabras, como por ejemplo, signos de puntuación, nombres propios, números, etc. La eliminación de estas

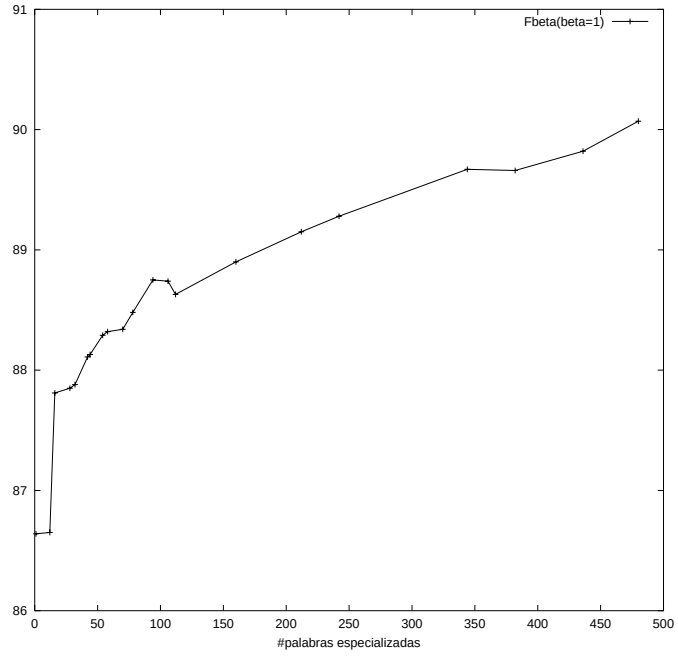

Figura 6.7: Evolución del factor F_β en función del número de palabras especializadas en el modelo contextual.

palabras del modelo contextual no repercute de manera negativa en los valores de F_β, sino que por el contrario, se reduce el tamaño de los modelos utilizados.

En la tabla 6.5 se presentan los resultados obtenidos para las diferentes unidades sintácticas consideradas con modelos BIG$_{esp}$-BIG$_{esp}$. Se observa que con la especialización propuesta se obtiene una mejora substancial del factor F_β (de 86.64 a 90.14). Las unidades SBAR, PP y PRT, caracterizadas por palabras como (*that, about, as, if, out, while, wether, for, to, from, in., out, ...*), han experimentado un aumento considerable. Otras unidades con aumentos resaltables, aunque el número de muestras en el conjunto de prueba es menor en estos casos, son CONJP y INTJ. Sobre los NP y VP, aunque el aumento no es tan notable, también se ha producido un incremento en su detección.

A partir de estos resultados se deduce que la especialización en los modelos integrados, redunda en un aumento considerable en las prestaciones del análisis superficial al igual que ocurría en los experimentos de etiquetado léxico presentados.

Se han realizado experimentos adicionales, utilizando otros criterios de espe-

	precision	cobertura	$F_{\beta=1}$
ADJP	72.89 %	66.89%	69.76
ADVP	79.75%	74.13%	76.84
CONJP	40.00%	66.67%	50.00
INTJ	100.00%	100.00%	100.00
NP	90.28%	89.41%	89.84
PP	95.89%	95.14%	95.51
PRT	60.31%	74.53%	66.67
SBAR	82.07%	77.01%	79.46
VP	91.58%	91.58%	91.58
TOTAL	90.63%	89.65%	90.14

Tabla 6.5: Resultados con modelos integrados especializados (Precisión de etiquetado sobre etiquetas IOB1= 93.79%).

cialización, obteniéndose resultados similares; por ejemplo, tomando las palabras pertenecientes a categorías cerradas en el conjunto de entrenamiento (preposiciones, determinantes, ...etc.). También se ha observado que algunas palabras influyen de manera más directa en la mejora, mientras que por el contrario, otras reducen las prestaciones. Este efecto se puede observar en las fluctuaciones que aparecen en la gráfica de la figura 6.7. Pensamos que las prestaciones del sistema sobre la tarea se podrían incrementar considerando criterios lingüísticos que ayudaran a refinar el conjunto de palabras especializadas. Además, un estudio más profundo de la tarea, también podría ayudar a refinar nuestros modelos.

6.7.4 Comparación con otras Aproximaciones

Para la resolución de la tarea compartida planteada en el *WorkShop ConLL-2000* se han aplicado 11 sistemas. Los resultados obtenidos, medidos en términos de precisión, cobertura y factor $F_{\beta=1}$ se muestran en la tabla 6.6.

Estos sistemas[1] se pueden clasificar en cuatro grandes grupos:

[1]Las características de estas aproximaciones se publicarán en los *Proceedings* de *CoNLL-2000* y *LLL-2000*. Septiembre 2000. Lisboa, Portugal.

- Sistemas basados en reglas: [5], [10] y [11]
- Sistemas basados en memoria: [8]
- Sistemas estadísticos: [4], [6], [7], [9]
- Sistemas combinados: [1], [2], [3]

Las principales características de cada uno de ellos son las siguientes:

- En [11] se usa un analizador sintáctico *Alembic* basado en reglas de transformación. [10] utiliza reglas incontextuales y sensibles al contexto para transformar secuencias de etiquetas léxicas en *chunks*. En [5] se aplica el sistema ALLiS.
- En [8] se evalúa el ajuste de diferentes parámetros de un algoritmo de aprendizaje basado en memoria.
- Nuestro sistema [9] utiliza la aproximación detallada en la sección 6.7.3, considerando información léxico-contextual. En [4] se implementa un analizador basado en MM, cuya salida es corregida mediante un método basado en reglas de transformación y teniendo en cuenta la información obtenida mediante un método basado en memoria para estimar probabilidades de *chunks*. En [6] y [7] se usan métodos basados en Máxima Entropía, obteniéndose resultados muy similares entre ellos.
- En [3] se combinan 5 analizadores distintos utilizando la técnica de votación y analizando diferentes representaciones de los *chunks*. En [2] también se combinan 5 analizadores estableciendo diferentes pesos para cada uno de ellos. Finalmente, en [1] se obtiene el mejor de los resultados mediante una combinación que se realiza utilizando un algoritmo de programación dinámica.

Los resultados obtenidos con nuestra aproximación son inferiores a la mayoría de aproximaciones. Pensamos que esto es debido principalmente a los siguientes factores:

- Los métodos combinados utilizan mucha más información, ya que se basan en elegir los mejores resultados obtenidos con diferentes analizadores para ciertos casos.

Aproximaciones	Precisión	Cobertura	$F_{\beta=1}$
[1] Kudoh and Matsumoro	93.45%	93.51%	93.48
[2] Van Halteren	93.13%	93.51%	93.32
[3] Tjong Kim Sang	94.04%	91.00%	92.50
[4] Zhou, Tey and Su	91.99%	92.25%	92.12
[5] Déjean	91.87%	91.31%	92.09
[6] Koeling	92.08%	91.86%	91.97
[7] Osborne	91.65%	92.23%	91.94
[8] Veenstra and Van den Bosch	91.05%	92.03%	91.54
[9] Pla, Molina and Prieto	90.63%	89.65%	90.14
[10] Johansson	86.24%	88.25%	87.23
[11] Vilain and Day	88.82%	82.91%	85.76
Sistema Base	72.58%	82.14%	77.07

Tabla 6.6: Resultados de diferentes aproximaciones en la tarea de detección de un conjunto de unidades sintácticas definidas en CoNLL-2000.

- Los métodos basados en máxima entropía y basados en memoria pueden contemplar mayor información y de muy diversa naturaleza.

- Los sistemas basados en reglas, aunque pueden modelizar mejor ciertos problemas, para la tarea propuesta, excepto con el trabajo [5], presenta resultados inferiores a los nuestros.

- La aproximación más similar a la nuesta es la presentada en [4]. Cuando se utiliza el sistema básico, sin amplicar reglas de transformación ni aprendizaje basado en memoria, los resultados son inferiores a los nuestros ($F_{\beta=1} = 89.57$)

- Si se hace un análisis de los distintos sintagmas, se observa que para PRT, PP y SBAR nuestra aproximación obtiene resultados muy elevados, que en algunos casos, son superiores a las demás aproximaciones. Pensamos que esto es debido a que la lexicalización de los modelos considerada favorece a estas unidades. Sin embargo, para otras unidades como NP, que constituyen aproximadamente el 50% de las unidades del conjunto de prueba, no se observa un incremento tan espectacular. Un estudio más detallado del léxico de estas unidades o quizá el uso de criterios lingüísticos para definir el conjunto de palabras a

especializar, podría repercutir en una mejora de las prestaciones para los NP y consecuentemente un incremento de las prestaciones globales.

Respecto a las ventajas de nuestro sistema consideramos que las principales son:

- Sencillez del proceso de aprendizaje y análisis.
- Sencillez para establecer dependencias léxico-contextuales.
- Integración de los procesos de etiquetado léxico y análisis superficial con unas prestaciones[2] aceptables. Con esta integración se mejora la eficiencia computacional de los procesos, con lo que se puede incorporar fácilmente a un sistema de PLN, sin disminuir seriamente las prestaciones del mismo.

6.8 Detección de SN sobre LexEsp

Se ha utilizado las mismas particiones usadas en la evaluación del sistema de etiquetado presentadas en el capítulo 5. A partir de éstas, mediante el analizador parcial de oraciones (*APOLN)* se ha realizado la segmentación en SN del conjunto de prueba y aprendizaje, sin realizar ningún tipo de supervisión manual.

APOLN, Analizador Parcial de Oraciones en Lenguaje Natural (Molina et al., 1999a; Molina et al., 1999b), permite realizar el análisis parcial de oraciones escritas en lenguaje natural no restringido. Está construido utilizando técnicas de máquinas de estados finitos. Su funcionamiento es incremental, es decir, analiza las oraciones aplicando una secuencia de pasos o niveles de procesamiento, de manera que la entrada a un determinado nivel es la salida del nivel inmediatamente anterior. En cada uno de estos niveles se reconoce un conjunto de estructuras sintáctica o patrones que se describen mediante definiciones regulares. Estos patrones se definen utilizando como alfabeto el conjunto de etiquetas léxicas y de patrones. Además, son patrones no recursivos, es decir, en la definición de un nivel determinado sólo pueden aparecer patrones definidos en niveles previos. APOLN toma como entrada

[2]No se proporcionan la precisión del etiquetado léxico debido a que la referencia no esta supervisada. Además, el conjunto de entrenamiento para esta tarea es escaso para que se puede comparar con otras aproximaciones.

```
Nivel 1 // núcleos nominales y verbales
 NSN ->(NC |NP)+
 NSV ->(VM |VA VMP)
Nivel 2 // sintagmas nominales
 SN ->TD? AQ* NSN AQ*
Nivel 3 // sintagmas preposicionales
 SPR ->SP SN
Nivel 4 // sintagmas verbales
 SV ->NSV (SN |SPR)*
```

Tabla 6.7: Definición de patrones utilizados en el sistema APOLN

oraciones etiquetadas léxicamente. En cada nivel se parentiza la entrada agrupando las secuencias de símbolos reconocidas por alguno de los patrones definidos en ese nivel y esto constituye la entrada para el siguiente nivel. Un ejemplo de definiciones de patrones sencillas para el castellano, utilizando las etiquetas léxicas *PAROLE*, se muestra en la tabla 6.7.

Método	Precisión SN	Cobertura SN	Prec.Etiq
BIG-BIG	93.2%	92.7%	96.9%
BIG-ECGI	92.5%	92.0%	96.9%
ECGI-ECGI	91.8%	91.4%	96.8%
ECGI-BIG	91.8%	91.5%	96.7%

Tabla 6.8: Resultados de etiquetado léxico y detección de NP sobre el corpus LexEsp usando diferentes combinaciones de modelos contextuales.

Una vez obtenida la segmentación en SN, a partir del conjunto de entrenamiento se han aprendido los modelos integrados combinado diferentes modelos contextuales. Los resultados de etiquetado léxico y detección de SN (Pla et al., 2000b), usando estos modelos, se muestran en la tabla 6.8. En este caso se obtiene una precisión de etiquetado de 96.9% usando los modelos integrados BIG-BIG o BIG-ECGI. Con modelos BIG se obtenía una precisión de 97.0% o 96.8% con modelos ECGI con suavizado ILC.

Los resultados de precisión y cobertura para los SN se han determinado tomando como referencia los SN salida de APOLN, sin realizar ninguna supervisión, por lo que su validez es relativa, ya que no se conoce el error asociado, por haberse segmentado automáticamente.

6.9 Resumen

Se ha propuesto una extensión del sistema de etiquetado léxico para abordar diferentes tareas de análisis sintáctico superficial. Para ello se han definido una serie de modelos estructurados que permiten realizar, mediante un proceso integrado, dos tareas que tradicionalmente se han abordado de manera separada: el etiquetado léxico y el análisis sintáctico superficial.

Mediante la integración de tareas, a pesar de que se obtienen resultados ligeramente inferiores, se consigue mejorar la eficiencia del proceso. Además, aplicando los modelos especializados se obtiene una mejora de los resultados, ya que pueden establecer restricciones contextuales adicionales teniendo en cuenta un conjunto de palabras preestablecido

Se ha aplicado con éxito sobre diferentes corpus, en inglés y en castellano, usando diferentes definiciones de unidades sintácticas. Además, se ha realizado una constrastación experimental rigurosa –tarea compartida– con otras aproximaciones utilizando el mismo conjunto de aprendizaje y de prueba. Se ha obtenido una precisión del 90.63%, una cobertura del 89.65% y un factor $F_{\beta=1} = 90.14$ sobre el total de las unidades sintácticas consideradas.

Capítulo 7

Entorno Gráfico para la Desambigüación de Textos

Es indudable el interés mostrado en los últimos años en el desarrollo de técnicas de análisis en los distintos niveles del procesamiento del lenguaje natural (morfológico, léxico, sintáctico y semántico) que se basan en métodos de aprendizaje. Dichas técnicas necesitan un corpus de aprendizaje anotado con la información que se debe aprender (etiquetas léxicas, parentizado sintáctico, roles sintácticos, etiquetas semánticas, etc). La construcción de grandes corpora anotados y supervisados es una tarea costosa y requiere un alto coste humano. Por ello, sería deseable realizar esta tarea lo más automáticamente posible. Además, aquellos aspectos que no son formalizables, deberían simplicarse, proporcionando un conjunto de facilidades que ayuden a la supervisión por parte de expertos lingüistas.

En este capítulo se describe una herramienta gráfica (Ribera et al., 2000) útil para realizar el etiquetado léxico y el análisis sintáctico. Se integra mediante un entorno gráfico: 1) El analizador morfológico *MACO* b) El etiquetador léxico basado en modelos regulares estocásticos: bigramas, modelos ECGIE o modelos especializados 3) El analizador parcial *APOLN* basado en expresiones regulares 4) El sistema integrado que realiza etiquetado léxico y análisis superficial y en el que se puede elegir cualquier combinación de los modelos presentados. Se proporciona un entorno gráfico que facilita la supervisión del proceso de desambiguación, permitiendo que el usuario corrija cualquier salida antes de continuar con la siguiente.

La aplicación se puede ejecutar en un ordenador personal que utilice el sistema operativo *Linux*. Los programas y facilidades desarrolladas se han implementado con el lenguaje de programación *C*, habiendo usado también diferentes herramientas del sistema operativo y otros lenguajes como *gawk*, *perl* y *tcl-tk* para el entorno gráfico.

7.1 Funcionalidad de la Aplicación

Las principales funciones que se pueden realizar utilizando el entorno gráfico desarrollado son:

- Edición de etiquetas
- Edición de gramáticas
- Visualización y corrección del etiquetado léxico y el análisis sintáctico
- Evaluación de prestaciones

7.1.1 Edición de Etiquetas

La aplicación integra editores de etiquetas léxicas y sintácticas. En principio se han incorporado las etiquetas léxicas definidas en el proyecto *PAROLE* y las utilizadas en el proyecto *Penn Treebank*. Usando estos editores se pueden definir nuevas etiquetas, así como modificar o eliminar las existentes.

Figura 7.1: Ventana de representación de una etiqueta léxica

Las etiquetas léxicas se presentan mediante una ventana donde se reflejan sus rasgos y los distintos valores definidos para cada rasgo (figura 7.1).

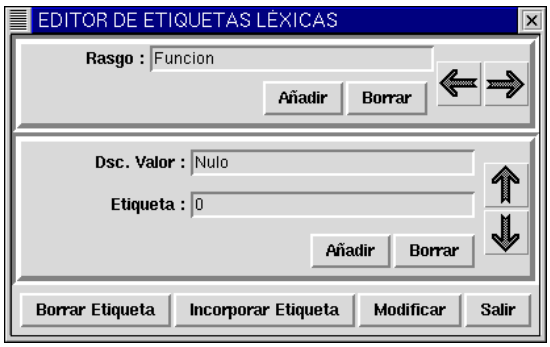

Figura 7.2: Ventana del manipulador de etiquetas léxicas

La manipulación de las etiquetas se realiza a través de una ventana que nos permite añadir, modificar o eliminar rasgos y valores asociados (figura 7.2).

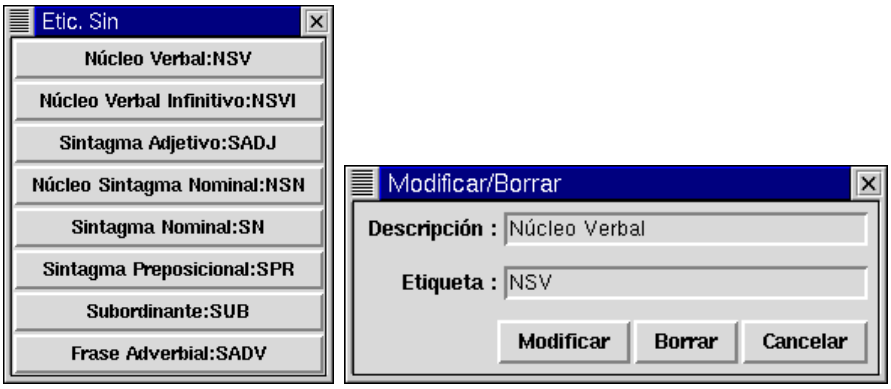

Figura 7.3: Lista de etiquetas sintácticas y ventana de manipulación

El editor de etiquetas sintácticas posee una funcionalidad similar al de etiquetas léxicas. Las etiquetas sintácticas se definen mediante su nombre (NSV, SADJ, SN, ...) y su descripción (Núcleo Verbal, Sintagma Adjetivo, Sintagma Nominal, ...) (figura 7.3).

7.1.2 Edición de Gramáticas

La aplicación incluye un sencillo editor que ofrece una serie de facilidades para la construcción y modificación de las gramáticas (basadas en definiciones regulares).

En la figura 7.4 se muestra la ventana de edición de las gramáticas. Los terminales de estas gramáticas se deben seleccionar de la ventana de edición y están formados por el conjunto de etiquetas léxicas y sintácticas previamente definido. Con esto se evita la construcción de expresiones regulares con símbolos incorrectos. Además, la validación sintáctica de las mismas se puede realizar mediante un proceso de compilación.

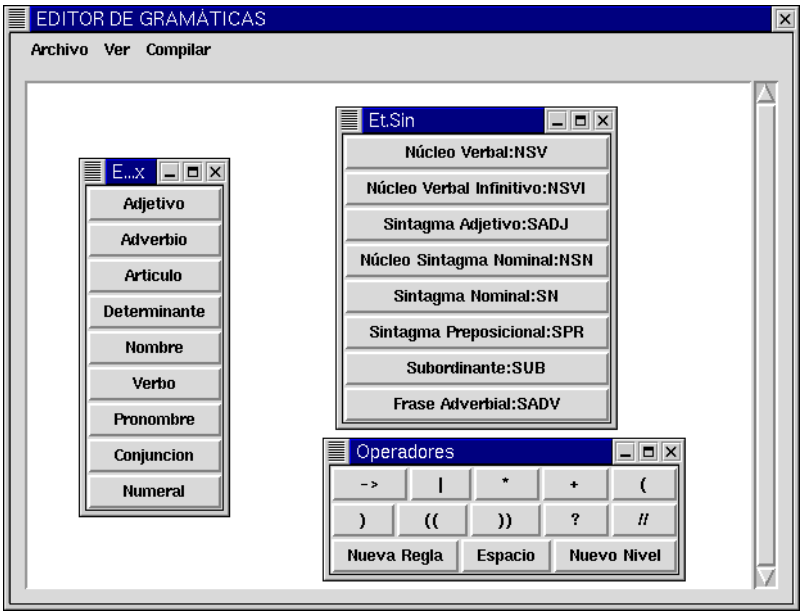

Figura 7.4: Editor de gramáticas

7.1.3 Visualización y Corrección del Etiquetado Léxico y el Análisis Sintáctico

La herramienta permite llevar a cabo el proceso de análisis de forma secuencial: análisis morfológico, etiquetado léxico y análisis sintáctico, a partir de un corpus de frases no etiquetadas o bien introduciendo interactivamente un conjunto de frases. En cada uno de los pasos se puede elegir el modelo o modelos a utilizar.

El usuario puede corregir la salida de cualquiera de estas fases, mediante el uso de un "navegador de frases" (figura 7.5) que facilita el acceso directo a cualquiera de las oraciones del corpus.

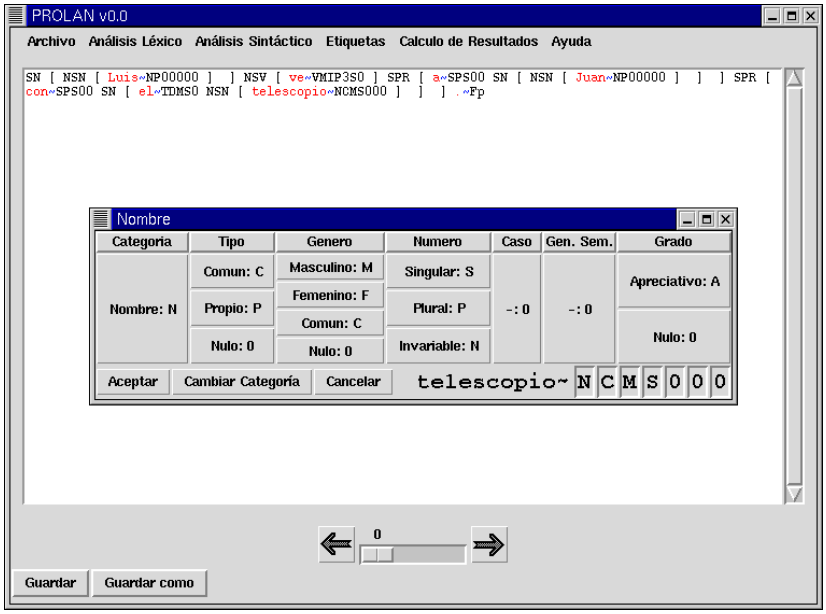

Figura 7.5: Resultado del análisis con parentizado a izquierdas

La visualización y edición de oraciones puede realizarse en "modo texto" y en "modo gráfico". Por ejemplo, la figura 7.5 muestra en "modo texto" el resultado del proceso de análisis. Las etiquetas léxicas y sintácticas, se resaltan en color facilitando su lectura. Además son sensibles al puntero del ratón permitiendo su modificación utilizando las ventanas de edición comentadas anteriormente.

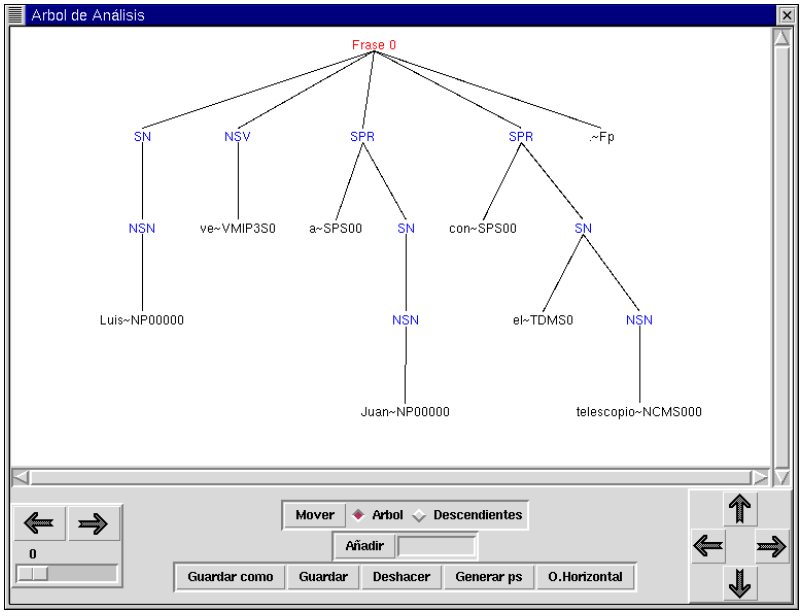

Figura 7.6: Árbol sintáctico en "modo gráfico" salida de APOLN.

En las figuras 7.6 y 7.7 puede verse el mismo ejemplo con la presentación en "modo gráfico" del árbol de análisis. De esta forma se facilita tanto la visualización del análisis como la corrección del mismo. Con este modo cualquier etiqueta es susceptible de modificación, al igual como ocurría en "modo texto". Es posible añadir, eliminar, modificar o mover cualquier nodo del árbol sintáctico. En la figura 7.7 se muestra además, cómo se ha completado el análisis sintáctico, obtenido de manera automática mediante el analizador parcial APOLN (figura 7.6), añadiendo un nuevo nodo (SV) en el árbol.

Por otra parte, se ofrecen otras facilidades que permiten modificar la distancia entre ramas y nodos, adaptando la longitud de la frase a la pantalla así como la posibilidad de generar la imagen del árbol sintáctico construido en formato *'postscript'* para su impresión.

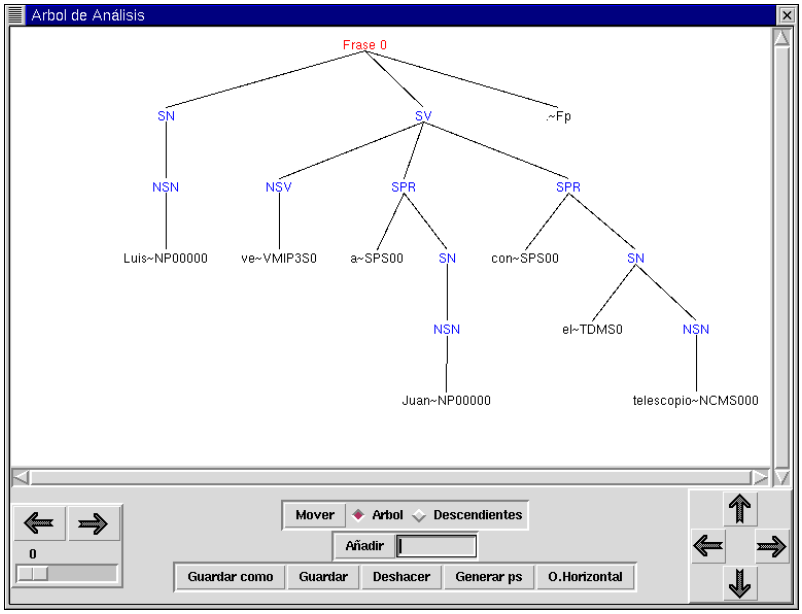

Figura 7.7: Árbol sintáctico en "modo gráfico" completado con un SV.

7.1.4 Evaluación de Prestaciones

La herramienta permite la evaluación automática de las prestaciones de cada uno de los procesos de desambiguación que se contemplan en la aplicación. Si se dispone de un corpus de referencia supervisado, se pueden presentar resultados de precisión de los distintos etiquetadores integrados, así como de la precisión y cobertura del análisis sintáctico en lo que respecta a la detección de estructuras no recursivas ('*chunks*'), tanto de manera global, como particularizada para cada estructura.

7.2 Ventajas de la Herramienta Gráfica

Las principales ventajas de la herramienta presentada son las siguientes.

- Facilita la tarea creación y de supervisión de corpora anotados.
- Es útil en procesos de '*bootstraping*' de manera que se facil la supervisión de textos para su posterior incorporación al conjunto de entrenamiento.
- Es una herramienta flexible que incorpora los conjuntos de etiquetas léxicas utilizadas en este trabajo para el castellano y para el inglés, pero que permite que el usuario las configure según sus necesidades. Incluso puede definirse otros tipos de etiquetas (roles gramaticales, etiquetas semánticas, etc.) con las cuales anotar los nodos del árbol de análisis.
- El modo gráfico simplifica la tarea de construcción del análisis sintáctico completo de las frases.
- La posibilidad de integrar cualquier etiquetador léxico o analizador sintáctico que respete los formatos de etiquetado y parentizado sintáctico utilizados.

Respecto a las mejoras, se está trabajando en los siguientes aspectos:

- La integración de la fase de entrenamiento de los modelos necesarios para el etiquetado léxico y el análisis integrado.
- Posibilitar al usuario facilidades para la integración de sus propias herramientas de etiquetado y análisis morfológico y sintáctico.

Capítulo 8

Conclusiones y Trabajos Futuros

8.1 Conclusiones

En este trabajo se ha desarrollado un sistema capaz de abordar las tareas de etiquetado léxico y el análisis sintáctico superficial de textos no restringidos. El sistema representa los diferentes niveles de conocimiento del lenguaje considerados como modelos de estados finitos que se establecen de manera automática a partir de grandes cantidades de datos anotados con información lingüística.

Para ello se han utilizado métodos estadísticos, métodos de aprendizaje automático y técnicas de inferencia gramatical con el fin de obtener ML de diferentes unidades lingüísticas que sean capaces de resolver los distintos casos de ambigüedad planteados. Estos ML pueden ser desde modelos simples, para abordar la desambiguación de las palabras en sus posibles categorías léxicas, hasta modelos estructurados en dos niveles, para resolver el etiquetado léxico y el análisis sintáctico superficial de manera integrada. Además, se ha presentado una técnica para obtener ML especializados (lexicalizados) que permiten representar restricciones léxico-contextuales relevantes.

Los diferentes ML considerados se han evaluado y contrastado experimentalmente en varias tareas de desambiguación, diseñadas sobre corpora en inglés y en castellano, obteniendo resultados similares o, en algunos casos, superiores a los de otras aproximaciones.

Se ha diseñado un entorno gráfico que integra todas las herramientas desarrolladas y que facilita su uso para usuarios no expertos. Además, se proporcionan una serie de facilidades que ayudan a la supervisión por lingüistas de corpora etiquetados y analizados parcialmente de manera automática.

Las contribuciones de esta tesis se detallan en cada uno de los capítulos. A continuación se resaltan las más importantes:

- Se ha utilizado un formalismo homogéneo –modelos de estados finitos– para representar tanto modelos estadísticos (n-gramas) como modelos sintácticos obtenidos mediante técnicas de inferencia gramatical (ECGI). Esta representación dota de gran flexibilidad al sistema ya que se puede utilizar cualquier modelo regular, obtenido por otras técnicas de inferencia gramatical, o diseñado por expertos y posteriormente reestimando a partir de datos.

- En el capítulo 4 se han propuesto métodos de suavizado aplicables a cualquier autómata de estados finitos. En este trabajo se han utilizado sobre los modelos ECGI y se han evaluado experimentalmente sobre diferentes tareas de desambiguación.

- Se ha presentado una técnica que permite enriquecer los modelos contextuales, utilizados para el etiquetado léxico y el análisis superficial, con la incorporación de ciertas palabras a los mismos, además de las categorías léxicas. Se han utilizado con éxito criterios totalmente automáticos para establecer el conjunto de palabras a especializar (las palabras más frecuentes del conjunto de entrenamiento), aunque se pueden aplicar otros.

- Respecto a la evaluación del sistema de etiquetado se ha realizado un conjunto de experimentos sobre diferentes corpora. Los mejores resultados se han obtenido con modelos de bigramas especializados alcanzando una precisión del 97.3% sobre el WSJ (asumiendo un analizador morfológico ideal) y de 97.4% sobre LexEsp (utilizando el analizador MACO). Se ha abordado la tarea de etiquetado y supervisión de nuevos corpora, en concreto sobre el corpus en castellano BDGEO. Utilizando una técnica de 'bootstrapping' se ha constatado la utilidad del sistema propuesto para este tipo de tareas. Por otra parte, la tarea de supervisión por expertos también se facilita con el uso del entorno

gráfico presentado en el capítulo 7. Sobre tareas restringidas (corpus BDGEO, experimentos cerrados) las prestaciones de etiquetado usando modelos ECGI son superiores.

- Los ML integrados presentados en el capítulo 6 se han evaluado sobre diferentes tareas de análisis superficial. La principal característica de estos modelos es que permiten elegir cualquier combinación de ML en sus dos niveles. Además, con éstos se puede realizar en un único proceso integrado el etiquetado léxico de textos y el análisis superficial, con el consiguiente aumento de la eficiencia computacional y sin que las prestaciones se vean afectadas de manera significativa. Esta propiedad diferencia esencialmente nuestro sistema de los principales sistemas aparecidos recientemente en este campo, en los que el etiquetado léxico y el análisis sintáctico superficial se realizan de manera secuencial. No obstante, nuestro sistema también puede abordar ambos problemas de manera separada (proceso secuencial). Los resultados de precisión y cobertura obtenidos son similares a otras aproximaciones estadísticas, con la ventaja de que no requieren textos de entrada etiquetados con categorías léxicas, puesto que ambos procesos se realizan de manera integrada. Se ha realizado una constrastación experimental rigurosa –tarea compartida– con otras aproximaciones utilizando el mismo conjunto de aprendizaje y de prueba. Se ha obtenido una precisión del 90.63%, una cobertura del 89.65% y un factor $F_{\beta=1} = 90.14$ sobre el total de las unidades sintácticas consideradas.

8.2 Trabajos Futuros

Los trabajos futuros que se derivan de este trabajo se centran en varias direcciones. Por una parte en el refinamiento del sistema construido y por otra, en su aplicación a otras tareas de desambiguación o su integración en otras aplicaciones.

8.2.1 Refinamiento de los Modelos

- Uso de otras técnicas de inferencia gramatical que proporcionen modelizaciones más complejas que los modelos de n-gramas. En ese sentido, aunque se ha utilizado el algoritmo ECGI para tal fin, los modelos obtenidos presentan una

excesiva complejidad, sin que ello repercuta, de manera apreciable, en unas mejores prestaciones, al menos en las tareas abordadas en esta tesis. Por otra parte, como los métodos de suavizado introducidos para los modelos ECGI son directamente aplicables a cualquier autómata de estados finitos, se pueden usar para cualquier autómata obtenido por otro método de inferencia gramatical o desarrollado manualmente siguiendo criterios lingüísticos y posteriormente entrenados a partir de corpora anotados con información lingüística.

- Se pretende hacer uso del analizador sintáctico parcial (APOLN) para que proporcione autómatas de estados finitos probabilísticos derivados de las definiciones de expresiones regulares y entrenados posteriormente a partir de corpus. Con esto se permitiría usar modelos de diferentes unidades sintácticas, similares o más complejas que las utilizadas. Además, se podría extender el sistema para otras lenguas como el catalán, o refinar la aproximación presentada para el castellano.

- Explorar otros criterios para definir el conjunto de palabras que se tienen en cuenta en los modelos contextuales especializados.

- Refinamiento del modelo léxico, principalmente para el tratamiento de las palabras desconocidas. En ese sentido se debería incorporar un analizador morfológico para el inglés, con lo que se podrían establecer comparaciones cuantitativas con otras aproximaciones bajo las mismas condiciones.

- Uso de técnicas de combinación de los resultados obtenidos con los diferentes modelos contextuales presentados con el fin de obtener prestaciones mayores.

8.2.2 Aplicaciones del Sistema Desarrollado

El sistema se puede aplicar directamente a diferentes problemas que necesiten un modelo de lenguaje.

- En sistemas de reconocimiento automático del habla, o en sistemas de traducción, para construir ML de categorías (autómatas y traductores de estados finitos) en vez de palabras y así reducir el tamaño y la complejidad de los mismos. Además, se podrían utilizar los modelos especializados con el fin de

obtener modelizaciones a nivel de palabra (para las más frecuentes) y a nivel de categoría léxica para el resto, con lo que el problema de las palabras desconocidas se simplificaría. También se podrían extender los modelos estructurados introduciendo niveles superiores que contemplaran aspectos semánticos, roles (como sujeto y objeto), o niveles inferiores que representaran modelizaciones acústicas en aplicaciones de reconocimiento automático del habla.

- En tareas de diálogo también se podría utilizar el modelo integrado para abordar problemas de comprensión. Para ello se debe disponer de una segmentación de las frases en unidades semánticas.

- Se estudiará la posibilidad de utilizar los diferentes niveles de anotación proporcionados por el sistema que se ha desarrollado (etiquetado léxico y análisis sintáctico superficial) en sistemas de recuperación de información.

Apéndice A

Conjunto de Categorías Léxicas

En este apéndice se describe el conjunto de categorías léxicas utilizadas en los experimentos realizados para validar el sistema de etiquetado léxico y análisis sintáctico superficial propuesto es este trabajo.

Se describen las categorías PAROLE utilizadas en la anotación de los corpora en castellano *LexEsp* y *BDGEO*. Son 230 etiquetas estructuradas que representan información de la categoría, subcategoría, rasgos morfológicos como género y número, información del modo, tiempo y persona para los verbos, etc. Se describe la estructura de la etiquetas completas así como el conjunto de categorías reducido empleado en el aprendizaje de los modelos contextuales.

Para el inglés se listan el conjunto de etiquetas léxicas definido en el corpus *Wall Street Journal*. También se presentan las etiquetas sintácticas utilizadas en la anotación del análisis sintáctico completo de las frases. Además, se presenta un ejemplo de frase analizada completamente y su transformación a unidades sintácticas no solapadas ('*chunk*'), utilizando la notación IOB1.

A.1 Estructura Completa de las Categorías Léxicas PAROLE

- Adjetivo

	Atributo	**Valor**	**Código**
1	categoría	Adjetivo	A
2	tipo	Calificativo	Q
3	grado	Positivo	P
		Comparativo	C
		Superlativo	S
		Intensivo	I
		Apreciativo	A
4	genéro	masculino	M
		femenino	F
		común	C
5	número	singular	S
		plural	P
		invariable	I
6	caso	-	0
7	función	modificador	M
		especificador	S

- Adverbio

	Atributo	**Valor**	**Código**
1	categoría	Adverbio	R
2	tipo	General	G
3	grado	Positivo	P
		Comparativo	C
		Superlativo	S
		Intensivo	I
		Apreciativo	A
4	función	modificador	M
		especificador	S
5	pronominalidad	interrogativo	Q
		relativo	R

- Artículo

	Atributo	**Valor**	**Código**
1	categoría	Artículo	T
2	tipo	Definido	D
		Indefinido	I
		Personal	P
3	genéro	masculino	M
		femenino	F
		común	C
4	número	singular	S
		plural	P
5	caso	-	0

- Determinantes

	Atributo	**Valor**	**Código**
1	categoría	Determinante	D
2	tipo	Demostrativo	D
		Posesivo	P
		Interrogativo	T
		Exclamativo	E
		Indefinido	I
3	persona	primera	1
		segunda	2
		tercera	3
4	genéro	masculino	M
		femenino	F
		común	C
5	número	singular	S
		plural	P
		invariable	N
6	caso	-	0
7	poseedor	singular	S
		plural	P

- Conjunciones

	Atributo	**Valor**	**Código**
1	categoría	Conjunción	C
2	tipo	Coordinada	C
		Subordinada	S
3	?	-	0
4	posicion	-	0

- Nombre

	Atributo	**Valor**	**Código**
1	categoría	Nombre	N
2	tipo	común	C
		Propio	P
3	genéro	masculino	M
		femenino	F
		común	C
4	número	singular	S
		plural	P
		invariable	N
5	caso	-	0
6	genéro semántico	-	0
7	grado	apreciativo	A

- Verbo

	Atributo	**Valor**	**Código**
1	categoría	Verbo	V
2	tipo	Principal	M
		Auxiliar	A
3	Modo	Indicativo	I
		Subjuntivo	S
		Imperativo	M
		Condicional	C
		Infinitivo	N
		Gerundio	G
		Participio	P
4	tiempo	presente	P
		imperfecto	I
		futuro	F
		pasado	S
5	persona	primera	1
		segunda	2
		tercera	3
6	número	singular	S
		plural	P
7	genéro	masculino	M
		femenino	F

- Numerales

	Atributo	**Valor**	**Código**
1	categoría	Numeral	M
2	tipo	Cardinal	C
		Ordinal	O
3	genéro	masculino	M
		femenino	F
		común	C
4	número	singular	S
		plural	P
5	caso	-	0
6	función	pronominal	P
		determinante	D
		adjetivo	A

- Pronombre

	Atributo	**Valor**	**Código**
1	categoría	Pronombre	P
2	tipo	Personal	P
		Demostrativo	D
		Posesivo	X
		Indefinido	I
		Interrogativo	T
		Relativo	R
3	persona	primera	1
		segunda	2
		tercera	3
4	genéro	masculino	M
		femenino	F
		común	C
5	número	singular	S
		plural	P
		invariable	N
6	caso	nominativo	N
		acusativo	A
		dativo	D
		oblicuo	O
7	poseedor	singular	S
		plural	P
8	"politeness"	"polite"	(usted)P

- Interjecciones I
- Abreviatura Y
- Residuales X
- Puntuacion F

A.2 Categorías Léxicas PAROLE

A partir del conjunto completo de categorías PAROLE se define el siguiente conjunto reducido de etiquetas léxicas.

AQAdjetivos
C0 Conjunción sin clasificar
CC Conjunción Coordinada
CS Conjunción Subordinada
D0 Determinante sin clasificar
DD Determinante Demostrativo
DE Determinante Exclamativo
DI Determinantes Indefinidos
DP Determinante Posesivo
DT Determinante Interrogativo
E0 Términos Extranjeros
I Interjecciones
MC Numeral Cardinal
MO Numeral Ordinal
NC Nombre Común
NP Nombre Propio
P0 Pronombre sin clasificar
PD Pronombre Demostrativo
PI Pronombre Indefinido
PP Pronombre personal
PR Pronombre Relativo
PT Pronombre Interrogativo
PX Pronombre Posesivo
RG Adverbios y Fases Adverbiales
SP Preposiciones
TD Artículos
TI Determinante Indefinido
VAC
V Verbo **A** Auxiliar **C** Condicional
VAG G Gerundio
VAI I Otros tiempos de indicativo
VAM M Imperativo
VAN N Infinitivo
VAP P Participio
VAS S Subjuntivo
VMC M Principal
VMG
VMI
VMM
VMN
VMP
VMS
W Fecha
X Residuales
Y Abreviaturas
Z Cifras
SIGNOS DE PUNTUACIÓN
Faa ¡**Fah** [**Fai**¿
Fal { **Fap** (**Fc** ,
Fca ! **Fcd** ” **Fch**]
Fci ? **Fcl** } **Fcp**)
Fcs ’ **Fdp** : **Fg** -
Fgd – **Fp** . **Fpc** ; **Fps** ...
Fs / **Ftp** % **Fac** ≪ **Fcc** ≫

A.3 Categorías *Penn Treebank*

- **Conjunto de etiquetas léxicas**

1.	CC	Coordinating conjunction	2.	CD	Cardinal number
3.	DT	Determiner	4.	EX	Existencial *there*
5.	FW	Foreing Word	6.	IN	Preposition/sub. conjunction
7.	JJ	Adjective	8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative	10.	LS	List item marker
11.	MD	Modal	12.	NN	Noun, singular or mass
13.	NNS	Noun, plural	14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural	16.	PDT	Predeterminer
17.	POS	Possessive ending	18.	PRP	Personal pronoun
19.	PP$	Possessive pronoun	20.	RB	Adverb
21.	RBR	Adverb, comparative	22.	RBS	Adverb, superlative
23.	RP	Particle	24.	SYM	Symbol
25.	TO	to	26.	UH	interjeccion
27.	VB	Verb, base form	28.	VBD	Verb, pat tense
29.	VBG	Verb, gerund/present participle	30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd ps. isng. present	32.	VBZ	Verb, 3rd ps. sing. present
33.	WDT	*wh*-determiner	34.	WP	*wh*-pronoun
35.	WP$	Possessive *wh* -pronoun	36.	WRB	*wh*-adverb
37.	#	Pound sign	38.	$	Dollar sign
39.	.	Sentence-final punctuation	40.	,	Comma
41.	:	Colon, semi-colon	42.	(	Left bracket character
43.	)	Right bracket character	44.	“	Straight double quote
45.	‘	Left open single quote	46.	“	Left open double quote
47.	’	Right close single quote	48.	”	Right close double quote

- **Conjunto de etiquetas sintácticas**

1.	ADJP	Adjective phrase
2.	ADVP	Adverb phrase
3.	NP	Null phrase
4.	PP	Prepositional phrase
5.	S	Simple declarative clause
6.	SBAR	Clause introduced by subordinating conjunction or 0 (see below)
7.	SBARQ	Direct question introduced by wsh-word or wh-phrase
8.	SINV	Declarative sentence with subject-aux inversion
9.	SQ	Subconstituent os SBARQ excluding wh-word or wh-phrase
10.	S-CLF	it-clef, e.g. it was Casey who threw the ball
11.	SQ-CLF	interrogative it-cleft, e.g. was it Casey who threw the ball
12.	RRC	reduced relative clause, complementizer and finite verb are missing
13	FRAG	clause fragment
14.	VP	Verb phrase
15.	WHADVP	wh-adverb phrase
16.	WHNP	wh-noun phrase
17.	WHPP	wh-prepositional phrase
18.	QP	quantifier phrase
19.	PRT	particle, i.e. separated verb prefix
20.	UCP	unlike coordinate phrase
21.	PRN	parenthetical
22.	NX	head of a complex noun phrase
23.	NAC	not a constituent; to show scope of certain prenominal modifiers in a noun phrase
24.	INTJ	interjection
25.	CONJP	conjunction phrase, only used whith adjacent multi-element conjunctions
26.	X	Constituent of unknown or uncertain category

- **Null elements**

1.	*	"Undertood" subject of invinitive or imperative
2.	0	Zero variant of that in subordinate clauses
3.	T	Trace-marks position where moved wh-constituent is interpreted
4.	NULL	Marks position where preposition is interpreted in pied-piping contexts

- **Roles:**

Text Categories	
-HLN	headlines, datelines
-TTL	titles
-LST	list markers, i.e. mark list items in a text

Grammatical Functions	
-CLF	true clefts, see S-CLF, and SQ-CLF above
-NOM	non-NP functioning as NP
-ADV	clausal, and nominal adverbials
-LGS	logical subjects in passives
-PRD	non-VP predicates
-SBJ	surface subject
-TPC	topicalized, frontend constituent
-CLR	closely related
-DTV	dative PP-object

Semantic Roles	
-VOC	vocative
-DIR	direction, trajectory
-LOC	manner
-PRP	purpose, reason
-TMP	temporal phrases
-BNF	benefactive
-PUT	locative complement of the verb 'put'
-EXT	extent, spatial extent of an activity

- **Ejemplo de anotación de frase**

Análisis sintáctico completo para una frase extraída del corpus WSJ. También aparecen la categoría léxica asociada a cada palabra.

```
( (S
   (NP-SBJ
     (NP (NNP Rockwell) (NNP International) (NNP Corp.) (POS 's) )
     (NNP Tulsa) (NN unit) )
   (VP (VBD said)
     (SBAR (-NONE- 0)
       (S
         (NP-SBJ (PRP it) )
         (VP (VBD signed)
           (NP
             (NP (DT a) (JJ tentative) (NN agreement) )
             (VP (VBG extending)
               (NP
                 (NP
                   (NP (PRP$ its) (NN contract)
                     (S (-NONE- *ICH*-1) ))
                   (PP (IN with)
                     (NP (NNP Boeing) (NNP Co.) )))
                 (S-1
                   (NP-SBJ (-NONE- *) )
                   (VP (TO to)
                     (VP (VB provide)
                       (NP
                         (NP (JJ structural) (NNS parts) )
                         (PP (IN for)
                           (NP
                             (NP (NNP Boeing) (POS 's) )
                             (CD 747) (NNS jetliners) ))))))))))))
   (. .) ))
```

A continuación se presenta un ejemplo de transformación del análisis completo de la frase anterior –notación del *Penn Treebank*– a segmentos no solapados o *chunk* utilizando la notación IOB1. Dicha transformación se ha realizado mediante el *script* que se puede obtener en la dirección *http://ilk.kub.nl/~sabine/chunklink*. La conversión de árboles a estructuras no recursivas no es un problema trivial. Las dificultades que conlleva se pueden consultar en (Tjong-Kim-Sang and Buchholz, 2000).

```
Rockwell/B-NP International/I-NP Corp./I-NP 's/B-NP Tulsa/I-NP unit/I-NP
said/B-VP it/B-NP signed/B-VP a/B-NP tentative/I-NP agreement/I-NP
extending/B-VP its/B-NP contract/I-NP with/B-PP Boeing/B-NP Co./I-NP
to/B-VP provide/I-VP structural/B-NP parts/I-NP for/B-PP
Boeing/B-NP 's/B-NP 747/I-NP jetliners/I-NP ./O
```

Interprentando las etiquetas IOB1 se obtiene la siguiente segmentación en unidades sintácticas:

```
[NP Rockwell International Corp.] [NP 's Tulsa unit] [VP said] [NP it]
[VP signed] [NP a tentative agreement][VP extending] [NP its contract]
[PP with] [NP Boeing Co.] [VP to provide][NP structural parts]
[PP for][NP Boeing][NP 's 474 jetliners] .
```

Apéndice B

Corpus BDGEO

En este apéndice se presenta una muestra de frases del corpus BDGEO, etiquetadas de manera automática con el sistema propuesto y utilizando modelos ECGIE con suavizado BDFF. Estos modelos se han inferido incluyendo frases de la tarea en el aprendizaje y utilizando el conjunto de etiquetas PAROLE mostrado en el Apéndice A.2. Además, se ilustra el proceso mediante el cual se completan las etiquetas léxicas con información morfológica.

B.1 Frases del Corpus BDGEO

- ¿~Fai Cuál~PT es~VAI el~TD caudal~NC del~SP río~NC más~RG caudaloso~AQ que~PR desemboca~VMI en~SP el~TD mar~NC Mediterráneo~NP ?~Fci
- ¿~Fai Está~VMI el~TD archipiélago~NC de~SP las~TD Canarias~NP en~SP el~TD mar~NC en~SP que~PR desemboca~VMI el~TD río~NC Ebro~NP ?~Fci
- ¿~Fai Cuál~PT es~VAI el~TD número~NC de~SP habitantes~NC de~SP la~TD capital~NC de~SP la~TD comunidad~NC autónoma~AQ de~SP mayor~AQ extensión~NC ?~Fci
- ¿~Fai Cuál~PT es~VAI la~TD altura~NC del~SP pico~NC Aneto~NP ?~Fci
- ¿~Fai Cuántos~DT ríos~NC nacen~VMI en~SP los~TD Pirineos~NP que~PR desemboquen~VMS en~SP el~TD Mediterráneo~NP ?~Fci

- ¿~Fai Qué~DT ríos~NC nacen~VMI en~SP la~TD comunidad~NC Navarra~NP ?~Fci
- Ciudades~NC por~SP las~TD que~PR pasa~VMI el~TD río~NC Tajo~NP .~Fp
- Extensión~NC de~SP la~TD autonomía~NC donde~PR está~VMI la~TD ría~NC de~SP Arosa~NP .~Fp
- Dime~VMM el~TD nombre~NC de~SP las~TD montañas~NC cuya~PR altura~NC supera~VMI los~TD 1000~Z m~NC .~Fp
- ¿~Fai Existe~VMI algún~DI golfo~NC en~SP el~TD País_Vasco~NP donde~PR desemboque~VMS una~TI ría~NC que~PR no~RG pase~VMS por~SP una~TI ciudad~NC costera~AQ ?~Fci
- ¿~Fai Cuál~PT es~VAI el~TD nombre~NC del~SP mar~NC que~PR baña~VMI una~TI ciudad~NC de~SP más~RG de~SP 2~Z millones~NC de~SP habitantes~NC ?~Fci
- ¿~Fai Cuántos~DT habitantes~NC tiene~VMI la~TD ciudad~NC menos~RG poblada~VMP de~SP Cataluña~NP ?~Fci
- ¿~Fai En~SP qué~PT mar~NC desemboca~VMI el~TD mayor~AQ de~SP los~TD ríos~NC que~PR pasan~VMI por~SP Mallorca~NP ?
- ¿~Fai Puede~VMI decirme~VMN los~TD ríos~NC de~SP longitud~NC mayor~AQ de~SP 100~Z km~NC de~SP Asturias~NP ?~Fci
- De~SP las~TD comunidades~NC por~SP las~TD que~PR pasa~VMI el~TD Tajo~NP ,~Fc ¿~Fai cuál~PT de~SP ellas~PP tiene~VMI la~TD capital~NC con~SP más~RG habitantes~NC ?~Fci
- Picos~NC de~SP más~RG de~SP 1000~Z m~NC que~PR hay~VAI en~SP el~TD Sistema_Ibérico~NP .~Fp
- Dime~VMM el~TD nombre~NC de~SP un~TI pico~NC con~SP menos~RG de~SP 1000~Z m~NC de~SP altura~NC .~Fp
- Obtener~VMN el~TD nombre~NC de~SP las~TD comunidades~NC que~PR limitan~VMI con~SP el~TD mar~NC Mediterráneo~NP .~Fp

B.2 Etiquetas Completas

Una vez realizado el proceso de etiquetado con las categorías léxicas del apéndice A.1, si se desea tener información más completa (etiquetas PAROLE definidas en el apéndice A.2), en la mayoría de los casos se puede recuperar la etiqueta completa, teniendo en cuenta la información suministrada por el analizador morfológico MACO, sin ningún tipo de ambigüedad.

Por ejemplo en la frase:

¿ ~Fai Existe ~VMI algún ~DI golfo ~NC en ~SP el ~TD País_ Vasco ~NP donde ~PR desemboque ~VMS una ~TI ría ~NC que ~PR no ~RG pase ~VMS por ~SP una ~TI ciudad ~NC costera ~AQ ? ~Fci

para la mayor parte de las palabras (p.e. **Existe, costera)** se puede recuperar la etiqueta completa:

- **Existe** existir VMMP2S0 existir **VMI**P3S0

  Existe ~VMI → *Existe ~VMI*P3S0

- **costera** costera NCFS000 costero AQ0FS00

  costera ~AQ → **AQ**0FS00

Sin embargo, para otras, principalmente para la categorías verbales (p.e. **desemboque, pase)** queda cierta ambigüedad, como se ve en los siguientes ejemplos, cuando se intenta determinar la persona de la forma verbal.

- **desemboque** desembocar **VMS**P1S0 desembocar **VMS**P3S0 desembocar VMMP3S0 desemboque NCMS000

  desemboque ~VMS → [**VMS**P1S0|**VMS**P3S0]

- **pase** pasar VMSP1S0 pasar VMSP3S0 pasar VMMP3S0 pase NCMS000

  pase ~VMS → [VMSP1S0|VMSP3S0]

Apéndice C

Palabras Especializadas en los Modelos Contextuales

En este apéndice se muestra el conjunto de palabras que se ha tenido en cuenta en los modelos contextuales especializados, tanto para el inglés (corpus WSJ) como para el castellano (corpus LexEsp).

C.1 Sobre el Corpus WSJ

A	Aside	Early	If
About	At	Elsewhere	In
After	Back	Even	Indeed
All	Because	Eventually	Instead
Almost	Before	Every	Just
Along	Besides	Finally	Late
Already	Both	First	Later
Also	By	For	Like
Although	Clearly	From	Maybe
Among	Currently	Furthermore	Meanwhile
An	Despite	Generally	More
Another	During	Here	Moreover
Any	Each	How	Most
As	Earlier	However	Much

Nearly	These	aggressively	barely
Neither	This	ago	basically
Never	Those	ahead	because
Nevertheless	Though	alike	before
No	Through	all	behind
Nonetheless	Thus	allegedly	below
Not	Typically	almost	beneath
Now	Under	alone	best
OF	Unfortunately	along	better
Obviously	Unless	already	between
Of	Unlike	also	beyond
On	Until	although	both
Once	Up	altogether	broadly
Only	What	always	by
Otherwise	Whatever	amid	carefully
Over	When	among	certainly
Overall	Where	an	clearly
Perhaps	Whether	annually	close
Previously	Which	another	closely
Rather	While	any	closer
Recently	Who	anymore	commonly
Right	Why	anytime	completely
Separately	With	anyway	considerably
Shortly	Within	anywhere	consistently
Similarly	Without	apart	constantly
Since	Yet	apiece	currently
So	a	apparently	daily
Some	above	approximately	deeply
Sometimes	abroad	around	definitely
Soon	absolutely	as	deliberately
Still	across	aside	desperately
THE	actively	at	despite
That	actually	automatically	differently
The	after	away	directly
Then	again	back	double
There	against	badly	down

downward	first	indirectly	nearly
dramatically	for	inevitably	necessarily
due	forever	initially	neither
during	formally	inside	never
each	formerly	instead	newly
earlier	forth	internationally	next
early	forward	into	no
easily	freely	jointly	nonetheless
economically	frequently	just	normally
effectively	from	largely	north
either	fully	last	not
else	further	late	notably
elsewhere	generally	lately	now
enough	gradually	later	obviously
entirely	greatly	less	occasionally
equally	half	like	of
especially	hard	likely	off
essentially	harder	literally	officially
even	hardly	little	often
eventually	heavily	long	on
ever	here	longer	once
every	high	lower	only
everywhere	higher	mainly	onto
exactly	highly	many	openly
except	historically	marginally	originally
exclusively	home	maybe	otherwise
extremely	how	meanwhile	out
fairly	however	merely	outside
far	if	moderately	over
fast	illegally	modestly	overly
faster	immediately	more	overseas
federally	in	most	partially
fiercely	incorrectly	mostly	particularly
finally	increasingly	much	partly
financially	indeed	narrowly	past
firmly	indefinitely	near	per

perfectly
perhaps
personally
plus
politically
poorly
possibly
potentially
precisely
pretty
previously
primarily
prior
privately
probably
promptly
properly
publicly
quickly
quietly
quite
rapidly
rarely
rather
readily
really
reasonably
recently
regardless
regularly
relatively
repeatedly
reportedly
respectively
right
roughly
routinely
seasonally
seemingly
separately
seriously
severely
sharply
short
shortly
significantly
simply
simultaneously
since
slightly
slowly
smoothly
so
solely
some
somehow
sometime
sometimes
somewhat
somewhere
soon
sooner
south
specifically
steadily
still
strongly
subsequently
substantially
successfully
suddenly
sufficiently
supposedly
sure
surely
surprisingly
swiftly
temporarily
tentatively
than
that
the
then
there
thereafter
thereby
therefore
these
this
those
though
through
throughout
thus
together
too
totally
toward
traditionally
truly
twice
typically
ultimately
unanimously
under
undoubtedly
unexpectedly
unfairly
unless
unlike
until
unusually
up
upon
upward
usually
very
via
virtually
voluntarily
well
what
whatever
when
whenever
where
whether
which
while
who
whom
whose
why
widely
wildly
with
within
without
worth
yesterday
yet

C.2 Sobre el Corpus LexEsp

Al	Más	bien	no
Ambos	Nada	como	para
Aunque	Ni	contra	pero
Bajo	No	cuando	poco
Bien	Para	de	por
Como	Pero	del	que
Contra	Poco	desde	se
Cuando	Por	durante	seguro
De	Que	entonces	ser
Del	Se	entre	siempre
Desde	Seguro	eran	sin
Durante	Ser	incluso	sobre
Entonces	Siempre	la	sus
Entre	Sin	las	sí
Eran	Sobre	le	también
Incluso	Sus	les	todas
La	Sí	lo	todo
Las	También	los	total
Le	Todas	mediante	Último
Les	Todo	misma	Único
Los	al	mismo	último
Mediante	ambos	más	único
Misma	aunque	nada	
Mismo	bajo	ni	

Bibliografía

Abney, S. (1991). *Parsing by Chunks.* R. Berwick, S. Abney and C. Tenny (eds.) Principle–based Parsing . Kluwer Academic Publishers, Dordrecht.

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

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

Aduriz, I., Alegria, I., Arriola, J., Artola, X., de Ilarraza, A. D., Ezeiza, N., Gojenola, K., and Maritxalar, M. (1995). Different Issues in the Desing of a Lemmatizer/Tagger for Basque. In *Proceedings of the EACL SIGDAT Workshop*, Dublin, Ireland.

Allen, J. F. (1995). *Natural Language Understanding.* Computer Science. 2nd. ed. Benjamin Cummings.

Angluin, D. (1982). Inference of Reversible Languages. *Journal of the ACM*, 29(3):741–765.

Angluin, D. and Smith, C. (1983). Inductive Inference: Theory and Methods. *Computing Surveys*, 15(3):46–62.

Argamon, S., Dagan, I., and Krymolowski, Y. (1998). A Memory–based Approach to Learning Shallow Natural Language Patterns. In *Proceedings of the joint 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics, COLING-ACL*, pages 67–73, Montréal, Canada.

Aït-Mokhtar, S. and Chanod, J. (1997). Incremental Finite-State Parsing. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, Was-

hington D.C., USA.

Bahl, L. R. and Mercer, R. L. (1976). Part–of–speech Assignment by a Statistical Decision Algorithm. In *IEEE International Symposium on Information Theory*, pages 88–89.

Baker, J. (1979). Trainable Grammars for Speech Recognition. In *Speech Communication Papers for the 97th Meeting of the Acoustical Society of America*, pages 547–550. Klatt & Wolf (eds.).

Baum, L. E. (1972). An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of a Markov Process. *Inequalities*, 3:1–8.

Benedí, J. M. and Sánchez, J. A. (2000). Combination of N-Grams and Stochastic Contex-Free Grammars for Language Modeling. In *Proceedings of the COLING–2000*, Saarbrücken, Germany.

Berwick, R. and Pilato, S. (1987). Learning Sintax by Automata Induction. *Machine Learning*, 2:9–38.

Black, E., Jelinek, F., Lafferty, J., Mercer, R. L., and Roukos, S. (1992). Decision Tree Models Applied to the Labeling of Text with Parts–of–speech. In *Proceedings of the DARPA Workshop on Speech and Natural Language*, San Mateo, CA.

Bordel, G. (1993). Modelización del Lenguaje: Una visión general desde el análisis de los lenguajes k-explorables en sentido estricto (n-gramas). Internal Report: DSIC-II/40/93, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia.

Bordel, G. (1994). Back-off Smoothing in a Syntactic Approach to Language Modelling. In *Proceedings of International Conference on Speech and Language Processing, ICSLP-94*, pages 851–854.

Bourigault, D. (1992). Surface Grammatical Analysis for the Extraction of Terminological Noun Phrases. In *Proceedings of the 15th International Conference on Computational Linguistics*, pages 977–981.

Brants, T. (1996). Estimating markov model structures. In *Proceedings of 4th International Conference on Spoken Language Processing.*

Brants, T. (1999). Cascaded Markov Models. In *Proceedings of the EACL99*, Ber-

gen, Norway.

Brants, T. (2000). TnT – a statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing (ANLP-2000)*, Seattle, WA.

Brill, E. (1992). A Simple Rule–Based Part–of–speech Tagger. In *Proceedings of the 3rd Conference on Applied Natural Language Processing, ANLP*, pages 152–155. ACL.

Brill, E. (1993a). *A Corpus–based Approach to Language Learning.* Phd. Thesis, Department of Computer and Information Science, University of Pennsylvania. http://www.cs.jhu.edu/~brill/acadpubs.html.

Brill, E. (1993b). Automatic Grammar Induction and Parsing Free Text: A Transformation–based Approach. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics.*

Brill, E. (1994). Some Advances in Rule–based Part–of–speech Tagging. In *Proceedings of the 12th National Conference on Artificial Intelligence, AAAI*, pages 722–727. http://www.cs.jhu.edu/~brill/acadpubs.html.

Brill, E. (1995). Transformation–based Error–driven Learning and Natural Language Processing: A Case Study in Part–of–speech Tagging. *Computational Linguistics*, 21(4):543–565.

Brill, E. and Wu, J. (1998). Classifier Combination for Improved Lexical Disambiguation. In *Proceedings of the joint 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics, COLING-ACL*, pages 191–195, Montréal, Canada.

Briscoe, E. J. (1994). *Prospects for Practical Parsing of Unrestricted Text: Robust Statistical Parsing Techniques.* N. Oostdijk and P. de Haan (eds.), Corpus-Based Research into Language. Rodopi, Amsterdam.

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

Cardie, C. and Pierce, D. (1999). The Role of Lexicalization and Pruning for Base Noun Phrase Identification. In *Proceedings of the Sixteenth National Conference*

on Artificial Intelligence. http://www.cs.cornell.edu/home/pierce/papers.html.

Carmona, J., Cervell, S., Màrquez, L., Martí, M., Padró, L., Placer, R., Rodríguez, H., Taulé, M., and Turmo, J. (1998). An Environment for Morphosyntactic Processing of Unrestricted Spanish Text. In *Proceedings of the 1st International Conference on Language Resources and Evaluation, LREC*, pages 915–922, Granada, Spain.

Chen, K. and Chen, H. (1995). Extracting Noun Phrases from Large-Scale Texts: A Hybrid Approach and Its Automatic Evaluation. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, ACL*.

Church, K. W. (1988). A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In *Proceedings of the 1st Conference on Applied Natural Language Processing, ANLP*, pages 136–143. ACL.

Church, K. W. and Gale, W. A. (1991). A Comparison of the Enhanced Good–Turing and Deleted Estimation Methods for Estimating Probabilities of English Bigrams. *Computer Speech and Language*, 5:19–54.

Clarksond, P. and Ronsenfeld, R. (1997). Statistical Language Modeling using the CMU-Cambridge Toolkit. In *Proceedings of Eurospeech*, Rhodes, Greece.

Cutting, D., Kupiec, J., Pederson, J., and Sibun, P. (1992). A Practical Part–of–speech Tagger. In *Proceedings of the 3rd Conference on Applied Natural Language Processing, ANLP*, pages 133–140. ACL.

Daelemans, W., Bosch, A. v., and Weijters, T. (1997). *IGTree: Using Trees for Compression and Classification in Lazy Learning Algorithms*. D. Aha (ed.), Artificial Intelligence Review 11, Special issue on Lazy Learning. Kluwer Academic Publishers.

Daelemans, W., Buchholz, S., and Veenstra, J. (1999). Memory-Based Shallow Parsing. In *Proceedings of EMNLP/VLC-99*, pages 239–246, University of Maryland, USA.

Daelemans, W., Zavrel, J., and Berck, P. (1996a). Part-of-Speech Tagging for Dutch with MBT, a Memory-based Tagger Generator. In *Congresboek van de Interdisciplinaire Onderzoeksconferentie Informatiewetenchap*, TU Delft.

Daelemans, W., Zavrel, J., Berck, P., and Gillis, S. (1996b). MBT: A Memory–Based Part–of–speech Tagger Generator. In *Proceedings of the 4th Workshop on*

Very Large Corpora, pages 14–27, Copenhagen, Denmark.

Demartas, E. and Kokkinakis, G. (1995). Automatic Stochastic Tagging of Natural Language Text. *Computational Linguistics*, 21(2).

DeRose, S. J. (1988). Grammatical Category Disambiguation by Statistical Optimization. *Computational Linguistics*, 14:31–39.

Derouault, A. M. and Merialdo, B. (1984). Language Modelling at the Syntactic Level. In *Proceedings of the 7th International Conference on Pattern Recognition.*

Díaz, A., Peinado, A., Rubio, A., E.Segarra, N.Prieto, and F.Casacuberta (1998). ALBAYZIN: a Task-Oriented Spanish Speech Corpus. In *Proceedings of the 1st International Conference on Language Resources and Evaluation, LREC*, Granada, Spain.

Ejerhed, E. (1988). Finding Clauses in Unrestricted Text by Finitary and Stochastic Methods . In *Proceedings of Second Conference on Applied Natural Language Processing*, pages 219–227. ACL.

Fourney, G. (1973). The Viterbi Algorithm. In *Proceedings in IEEE*, volume 61, pages 268–278.

Francis, W. and Kučera, H. (1982). *Frequency Analysis of English Usage: Lexicon and Grammar*. Houghton Mifflin, Boston.

Franz, A. (1997). Independence Assumption Considered Harmful. In *Proceedings of ACL-EACL*, Madrid, Spain.

Fu, K. and Booth, T. (1975). Grammatical Inference: introduction and survey. Parts I and II. *IEEE Transactions on Systems, Man and Cybernetics*, 5:303–309,409–423.

Fu, K. and Booth, T. (1982). *Syntactic Pattern Recognition and Applications.* Prentice-Hall.

García, P. and Vidal, E. (1990). Inference of K-testable Languages In the Strict Sense and Application to Syntactic Pattern Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI*, 12(9):920–925.

García, P., Vidal, E., and Casacuberta, F. (1987). Local Languages, the Sucessor Method, and a Step towards a General Methodology for the Inference of Regular Grammars. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI*, 9(6):841–845.

Garside, R., Leech, G., and Sampson, G., editors (1987). *The Computational Analysis of English: A Corpus-Based Approach*. London and New York: Longman.

González, R. and Thomason, M. (1978). *Syntactic Pattern Recognition*. Addison-Wesley Publishing Company.

Good, I. J. (1953). The Population Frequencies of Species and the Estimation of Population Parameters. *Biometrika*, 40.

Greene, B. B. and Rubin, G. M. (1971). Automatic Grammatical Tagging of English. Technical Report, Department of Linguistics, Brown University.

Grefenstette, G. (1996). Light Parsing as Finite State Filtering. In *Proceedings of the ECAI Workshop on Extended Finite State Models of Language*, Budapest, Hugary.

Halteren, H. v., Zavrel, J., and Daelemans, W. (1998). Improving Data Driven Wordclass Tagging by System Combination. In *Proceedings of the joint 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics, COLING-ACL*, pages 491–497, Montréal, Canada.

Hindle, D. (1983). User manual for Fidditch. Technical memorandum 7590–142, Naval Research Laboratory.

Jelinek, F. (1991). Up from trigrams! In *Proceedings of European Conference on Speech Communication and Technology EUROSPEECH-91*, pages 1037–1040.

Jelinek, F. and Mercer, R. L. (1985). Probability Distribution Estimation from Sparse Data. Tecnical Disclosure Bulletin, IBM.

Johansson, S. (1986). The Tagged LOB corpus: User's Manual. Technical report, Norwegian Computing Centre for Humanities, Bergen, Norway.

Karlsson, F. (1990). Constraint Grammar as a Framework for Parsing Running Text. In *Proceedings of 13th International Conference on Computational Linguistics, COLING*, volume 3, pages 168–173, Helsinki, Finland. Karlgren, H (ed.) COLING–90.

Katz, S. M. (1987). Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35.

Kim, J. D., Lee, S. Z., and Rim, H. C. (1999). HMM Specialization with Selecti-

ve Lexicalization. In *Proceedings of the join SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora(EMNLP-VLC-99)*.

Klein, S. and Simmons, R. (1963). A Computational Approach to Grammatical Coding of English Words. *JACM*, 10:334–337.

Kupiec, J. (1992). Robust Part–of–speech Tagging Using a Hidden Markov Model. *Computer Speech and Language*, 6.

Lari, K. and Young, S. (1991). Application of Stochastic Context-Free Grammars using the Inside-Outside Algotithm. *Computer Speech and Language*, 5:237–257.

Lee, S.-Z., ichi Tsujii, J., and Rim, H.-C. (2000). Lexicalized Hidden Markov Models for Part-of-Speech Tagging. In *Proceedings of 18th International Conference on Computational Linguistics*, Saarbrucken, Germany.

Leech, G., Garside, R., and Atwell, E. (1983). Automatic Grammatical Tagging of the LOB Corpus. *ICAME News*, 7:13–33.

Magerman, D. M. (1996). Learning Grammatical Structure Using Statistical Decision–Trees. In *Proceedings of the 3rd International Colloquium on Grammatical Inference, ICGI*, pages 1–21. Springer-Verlag Lecture Notes Series in Artificial Intelligence 1147.

Maltese, G. and Mancini, F. A. (1991). A Technique to Automatically Assign Parts-of-Speech to Words Taking into Account Word-Ending Information through a Probabilistic Model. In *Proceedings of European Conference on Speech Communication and Technology EUROSPEECH-91*, pages 753–756.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2).

Màrquez, L. (1999). *Part–of–Speech Tagging: A Machine–Learning Approach based on Decision Trees*. Phd. Thesis, Dep. Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya.

Màrquez, L., Padró, L., and Rodríguez, H. (1998). Improving Tagging Accuracy by Voting Taggers. In *Proceedings of the 2nd Conference on Natural Language Processing & Industrial Applications, NLP+IA/TAL+AI*, pages 149–155, New Brunswick, Canada.

Màrquez, L. and Rodríguez, H. (1998). Part-of-Speech Tagging Using Decision Trees. In C. Nédellec and C. Rouveirol, editor, *LNAI 1398: Proceedings of the 10th European Conference on Machine Learning, ECML'98*, pages 25–36, Chemnitz, Germany. Springer.

Merialdo, B. (1994). Tagging English Text with a Probabilistic Model. *Computational Linguistics*, 20(2):155–171.

Miclet, L. (1990). Grammatical Inference. *Syntactic and Structural Pattern Recognition and Applications*, pages 237–290.

Molina, A., Pla, F., Moreno, L., and Prieto, N. (1999a). Incremental Partial Parser of Unrestricted Natural Language Sentences. In *Proceedings of VIII National Symposium on Pattern Recognition and Image Analysis*, pages 171–178, Bilbao, Spain.

Molina, A., Pla, F., Moreno, L., and Prieto, N. (1999b). APOLN: A Partial Parser of Unrestricted Text. In *Proceedings of 5th Conference on Computational Lexicography and Text Research COMPLEX-99*, pages 101–108, Pecs, Hungary.

Muñoz, M., Punyakanok, V., Roth, D., and Zimak, D. (1999). A Learning Approach to Shallow Parsing. In *Proceedings of EMNLP-WVLC'99*, Association for Computational Linguistics. http://l2r.cs.uiuc.edu/~dnar/Papers/emnlp99.ps.gz.

Ney, H. and Kneser, K. (1991). On smoothing techniques for bigram-based natural language modelling. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing ICASSP-91*, pages 825–828, Toronto.

Ney, H. and Kneser, K. (1994). On Structuring Probabilistic Dependencies in Stochastic Language Modelling. *Computer Speech and Language*, 8:1–38.

Oflazer, K. and Kuruöz, I. (1994). Tagging and Morphological Disambiguation of Turkish Text. In *Proceedings of the 4th Conference on Applied Natural Language Processing, ANLP*. ACL.

Oncina, J. (1991). *Aprendizaje de Lenguajes Regulars y Funciones Subsecuenciales.* Phd. Thesis, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia.

Oncina, J. and García, P. (1992). *Inferring Regular Languages in Polynomial Updated Time*, volume 1 of *Machine Perception and Artificial Intelligence. Pérez de la Blanca, Sanfeliu, Vidal (eds).* World Scientific Publ.

Oncina, J., García, P., and Vidal, E. (1993). Learning Subsequencial Transducers for Pattern Recognition Interpretation Tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI.*

Osborne, M. (1999). MDL-based DCG induction for NP identification. In Miles Osborne and Erik Tjong Kim Sang, editor, *CoNLL-99 Computational Natural Language Learning*, Association for Computational Linguistics.

Padró, L. (1996). POS Tagging Using Relaxation Labelling. In *Proceedings of the 16th International Conference on Computational Linguistics, COLING*, pages 877–882, Copenhagen, Denmark.

Padró, L. (1998). *A Hybrid Environment for Syntax–Semantic Tagging.* Phd. Thesis, Dep. Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya. http://www.lsi.upc.es/~padro.

Padró, L. and Màrquez, L. (1998). On the Evaluation and Comparison of Taggers: the Effect of Noise in Testing Corpora. In *Proceedings of the joint 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics, COLING-ACL*, pages 997–1002, Montréal, Canada.

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

Pereira, F. and Schabes, Y. (1992). Inside-Outside Re-estimation from Partially Bracketed Corpora. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 128–135.

Pla, F. (1999). Aproximación Probabilística al Etiquetado Automático de Textos. Technical Report, DSIC–II/14/99.

Pla, F. and Molina, A. (1999). Etiquetado Morfosintáctico del Corpus BDGEO. In *Proceedings of the CAEPIA*, Murcia, España.

Pla, F., Molina, A., and Prieto, N. (2000a). Tagging and Chunking with Bigrams. In *Proceedings of the COLING–2000*, Saarbrücken, Germany.

Pla, F., Molina, A., and Prieto, N. (2000b). An Integrated Statistical Model for Tagging and Chunking Unrestricted Text. In *Proceedings of the Text, Speech*

and Dialogue 2000, Brno, Czech Republic.

Pla, F., Molina, A., and Prieto, N. (2000c). Improving Chunking by means of Lexical-Contextual Information in Statistical Language Models. In *Proceedings of 18th CoNLL-2000 and LLL-2000*, Lisbon, Portugal.

Pla, F. and Prieto, N. (1998). Using Grammatical Inference Methods for Automatic Part–of–speech Tagging. In *Proceedings of 1st International Conference on Language Resources and Evaluation, LREC*, Granada, Spain.

Prieto, N. (1988). Extensión estocástica del algoritmo ECGI y su aplicación al reconocimiento de dicciionarios difíciles. In *III Simposium Nacional de Reconocimiento de Formas y Análisis de Imágenes*, Oviedo.

Prieto, N. (1995). *Aprendizaje de Modelos Semánticos para Sistemas de Comprensión del Habla*. Phd. Thesis, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia.

Rabiner, L. R. and Juang, B. H. (1986). An Introduction to Hidden Markov Models. *IEEE ASSP MAGAZINE*.

Ramshaw, L. and Marcus, M. (1995). Text Chunking Using Transformation-Based Learning. In *Proceedings of third Workshop on Very Large Corpora*, pages 82–94. ftp://ftp.cis.upenn.edu/pub/chunker/wvlcbook.ps.gz.

Ratnaparkhi, A. (1996). A Maximum Entropy Part–of–speech Tagger. In *Proceedings of the 1st Conference on Empirical Methods in Natural Language Processing, EMNLP*.

Ribera, X., Molina, A., and Pla, F. (2000). Herramienta para el etiquetado léxico y análisis sintáctico de textos orientado a la construcción de corpus supervisados. In *Proceedings of SPLN'2000*, Vigo. Spain.

Roche, E. and Schabes, Y. (1995). Deterministic Part–of–speech Tagging with Finite State Transducers. *Computational Linguistics*, 21(2):227–253.

Rosenfeld, R. (1994). *Adaptive Statistical Language Modelling: A Maximum Entropy Approach*. Phd. Thesis, School of Computer Science, Carnegie Mellon University.

Rosenfeld, R. (1996). A Maximum Entropy Approach to Adaptive Statistical Language Modeling. *Computer Speech and Language*, 10:187–228.

Rulot, H. (1992). *ECGI: un algoritmo de inferencia gramatical mediante corrección de errores*. Phd. Thesis, Universidad de Valencia.

Rulot, H., Prieto, N., and Vidal, E. (1989). Learning accurate finite-state structural models of words through the ECGI algorithms. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*.

Rulot, H. and Vidal, E. (1987). Modelling (sub)string-length-based constraints through a Grammatical Inference Method. *Pattern Recognition: Theory and Applications*.

Sampson, G. (1995). *English for the Computer. The SUSANNE Corpus and Analytic Scheme*. Clarendon Press, Oxford.

Samuel, K. (1998). Lazy Transformation-Based Learning. In *Proceedings of the 11th International Florida Artificial Intelligence Research Symposium Conference*, pages 235–239. cmp–lg/9806003.

Samuelsson, C. (1993). Morphological Tagging Based Entirely on Bayesian Inference. In *Proceedings of the 9th Nordic Conference of Computational Linguistics*, Stockholm, Sweden.

Samuelsson, C. (1995). A Novel Framework for Reductionistic Statistical Parsing. In *Proceedings of the 4th International Workshop on Parsing Technologies*, pages 208–215, Prague/Karlovy Vary, Czech Republic.

Samuelsson, C. and Voutilainen, A. (1997). Comparing a Linguistic and a Stochastic Tagger. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 246–253, Madrid, Spain. http://www.coli.uni-sb.de/~christer.

Sánchez, F. and Nieto, A. F. (1995). Desarrollo de un etiquetador morfosintáctico para el español. In *Proceedings of the 11th Congreso de la Sociedad Española para el Procesamiento del Lenguage Natural, SEPLN*, Universidad de Deusto, Bilbo, Spain. cmp-lg/9505035.

Sanchis, E. (1994). *Modelización estructural de unidades subléxicas del castellano mediante una técnica de inferencia gramatical basada en el análisis sintáctico corrector de errores*. Phd. Thesis, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia.

Schmid, H. (1994). Probabilistic Part–of–speech Tagging Using Decision Trees. In *Proceedings of the Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK.

Segarra, E. (1993). *Una aproximación inductiva a la comprensión del discurso continuo.* Phd. Thesis, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia.

Skut, W. and Brants, T. (1998a). A Maximum–Entropy Partial Parser for Unrestricted Text. In *Proceedings of the 6th Workshop on Very Large Corpora*, Montréal, Canada. cmp–lg/9807006.

Skut, W. and Brants, T. (1998b). Chunk Tagger – Statistical Recognition of Noun Phrases. In *Proceedings of the ESSLLI'98 Workshop on automated Acquisition of Syntax and Parsing*, University of Saarbrücken. cmp–lg/9807007.

Tjong-Kim-Sang, E. F. (2000a). Noun Phrase Representation by System Combination. In *Proceedings of ANLP-NAACL 2000*, Washington, USA. Morgan Kaufman Publishers. http://lcg-www.uia.ac.be/~erikt/papers/naacl2000.ps.

Tjong-Kim-Sang, E. F. (2000b). Text Chunking by System Combination. In *Proceedings of 18th CoNLL-2000 and LLL-2000*, Lisbon, Portugal.

Tjong-Kim-Sang, E. F. and Buchholz, S. (2000). Introduction to the CoNLL-2000 Shared Task: Chunking. In *Proceedings of 18th CoNLL-2000 and LLL-2000*, Lisbon, Portugal.

Tjong-Kim-Sang, E. F., Daelemans, W., Déjean, H., Koeling, R., Krymolowsky, Y., Punyakanok, V., and Roth, D. (2000). Applying System Combination to Base Noun Phrase Identification. In *Proceedings of 18th International Conference on Computational Linguistics COLING'2000*, pages 857–863, Saarbrücken, Germany. Morgan Kaufman Publishers. http://lcg-www.uia.ac.be/~erikt/papers/coling2000.ps.

Tjong-Kim-Sang, E. F. and Veenstra, J. (1999). Representing Text Chunks. In *Proceedings of EACL'99*, Association for Computational Linguistics. http://xxx.lanl.gov/abs/cs.CL/9907006.

Veenstra, J. (1998). Fast NP Chunking Using Memory-based Learning Techniques. In *Proceedings of BENELEARN-98: Eighth Belgian-Dutch Conference on Machine Learning*, Wageningen, the Netherlands. ftp://ilk.kub.nl/pub/papers/ilk.9807.ps.gz.

Veenstra, J. (1999). Memory-Based Text Chunking. In *Proceedings of ACAI*, Chania, Greece.

Vidal, E., Casacuberta, F., and a, P. G. (1993). Syntactic Learning Techniques for Language Modeling and Acoustic Phonetic Decoding (Grammatical Inference and Automatic Speech Recognition). In *NATO-ASI New Advances and Trends in Speech Recognition And Coding*, pages 95–201, Bubión, Granada.

Vidal, E., Prieto, N., Sanchis, E., and Rulot, H. (1988). *Application of the Error Correcting Grammatical Inference Method (ECGI) to multi-speaker IWR.* Springer Verlag. Ed. H. Niemann.

Viterbi, A. J. (1967). Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm. *IEEE Transactions on Information Theory*, pages 260–269.

Voutilainen, A. (1993). NPTool, a Detector of English Noun Phrases. In *Proceedings of the Workshop on Very Large Corpora.* ACL.

Voutilainen, A. and Järvinen, T. (1995). Specifying a Shallow Grammatical Representation for Parsing Purposes. In *Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics, EACL*, Dublin, Ireland.

Voutilainen, A. and Padró, L. (1997). Developing a Hybrid NP Parser. In *Proceedings of the 5th Conference on Applied Natural Language Processing, ANLP*, pages 80–87, Washington DC. ACL.

Weischedel, R., Schwartz, R., Palmucci, J., Meteer, M., and Ramshaw, L. (1993). Coping with Ambiguity and Unknown Words through Probabilistic Models. *Computational Linguistics*, 19(2):260–269.

XTAG (1998). A Lexicalized Tree Adjoining Grammar for English. IRCS Tech Report 98-18, The XTAG Research Group. University of Pennsylvania. http://www.cis.upenn.edu/~xtag/tech-report/.

Young, S. and Bloothooft (editors), G. (1997). *Corpus–based Methods in Language and Speech Processing.* An ELSNET book. Kluwer Academic Publishers, Dordrecht.