TEXTOS de APOYO

LINGÜÍSTICA

24

Antonio Moreno Sandoval

Lingüística computacional

EDITORIAL SÍNTESIS

LINGÜÍSTICA COMPUTACIONAL

Introducción a los modelos simbólicos, estadísticos y biológicos

Vallehermoso, 34. 28015 Madrid
Teléfono 91 593 20 98
http://www.sintesis.com

ISBN: 84-7738-617-X
Depósito legal: M. 35.709-1998

Impreso en España. Printed in Spain

Índice

Prólogo

Digámoslo claramente, no es fácil hablar con los ordenadores
y a veces es más fácil utilizar un intérprete.
Es económico y efectivo, tanto para el hombre como para el ordenador,
hablar lenguas diferentes e interactuar a través de un intermediario.

Martin Kay

La diversidad es la cualidad más universal.

Montaigne

Este libro está dirigido especialmente a estudiantes de Lingüística y de Computación que buscan una introducción al estado de la cuestión en los años noventa. Los excelentes manuales clásicos en la disciplina, como Winograd (1983), Grishman (1986), Allen (1987;1995) y Gazdar y Mellish (1989), reflejan el conocimiento de los años ochenta: tratan casi exclusivamente de la aplicación de modelos simbólicos al tratamiento computacional del lenguaje humano. Sin embargo, los noventa se han caracterizado por el significativo impulso de los modelos probabilísticos, basados en enormes corpus de datos. El presente libro de texto recoge las nuevas aportaciones y su combinación en sistemas híbridos, que buscan el procesamiento más eficaz de las emisiones lingüísticas.

Paralelamente, a mediados de los ochenta ya había un numeroso grupo de investigadores en Inteligencia Artificial que trabajaban en una aproximación diferente a los modelos matemáticos, en este caso inspirada en mode-

los biológicos. Todo un capítulo se dedica a presentar la aplicación de redes neuronales y algoritmos genéticos al procesamiento lingüístico.

En las Conclusiones se hace balance del estado actual de la cuestión y se lanza una predicción sobre el futuro cercano de los sistemas de procesamiento del lenguaje natural. En resumen, la estructura del libro se organiza en torno a *modelos* y desde una *perspectiva histórica*, posiblemente la manera más eficaz de introducir a los neófitos en un nuevo campo de conocimiento. Los capítulos centrales están organizados según el mismo esquema: presentación de los problemas que puede tratar el modelo en cuestión, así como los resultados que podemos esperar con la aplicación de sus distintas técnicas, para terminar con las limitaciones (inherentes o coyunturales) del modelo.

Una de las razones por las que hemos adoptado esta fórmula histórico-paradigmática es que, afortunadamente, hay una excelente bibliografía introductoria sobre *técnicas* y *aplicaciones*, aunque casi toda ella accesible únicamente en inglés. Este manual pretende modestamente complementar algunas parcelas que generalmente no recogen las obras de referencia habituales. Si utilizamos una metáfora culinaria, podríamos decir que éste no es un libro de recetas de cocina, sino de consejos generales para escoger apropiadamente el menú en función de los comensales. Una buena aproximación pedagógica podría consistir en utilizar este manual en conjunción con alguno de los otros, de manera que el nuestro sirva de acercamiento a las posibilidades y limitaciones de cada método, para luego pasar al estudio y aplicación de las técnicas específicas expuestas, por ejemplo, en los manuales de Gazdar y Mellish o el de Allen.

¿Qué conocimientos previos se necesitan para empezar este libro? Dado que es un manual introductorio, se asume que es un primer contacto con la Lingüística Computacional. Sin embargo, se presuponen algunas nociones básicas de teoría gramatical y computacional, que se pueden encontrar en los manuales mencionados o en cualquiera de los libros de texto que se utilizan en las facultades y escuelas españolas.

Por otra parte, el presente es uno de los primeros textos sobre el tema escrito en español y que toma como referencia básica el procesamiento del español. Aunque el ámbito de la Lingüística Computacional, en principio, puede abarcar todas o muchas de las lenguas naturales, lo cierto es que sólo un grupo reducido de lenguas ha recibido tratamiento informático. Además, cada lengua tiene sus propias peculiaridades, en especial su representación (orto)gráfica y aquellos puntos de mayor riqueza y ambigüedad estructural, que merecen una atención especial. Los manuales más conocidos reflejan claramente las idiosincrasias del procesamiento del inglés, prestando poca atención a aspectos esenciales para el español, como son la morfología, el orden casi libre de los constituyentes o la omisión del sujeto.

Buscando la sintonía con los tiempos, esta *introducción* no incluye una extensa bibliografía. Al contrario, sólo una breve lista de libros selecciona-

dos y comentados. El lector no se encontrará desprovisto, sin embargo, de la información necesaria para seguir investigando por su cuenta: proporcionamos un conjunto de direcciones de internet, cuyo contenido es actualizado permanentemente, donde el lector puede encontrar toda la bibliografía complementaria que necesite. El acceso *on-line* a la mayoría de trabajos que se publican en este campo facilita la información última a toda la comunidad científica. Parece que está cada vez más cerca el fin de las interminables bibliografías recopiladas de distintas fuentes, generalmente de difícil acceso, al tiempo que cada vez vemos más cerca el acceso al documento original, mostrado desde múltiples punteros. Afortunadamente, el acceso a la Lingüística Computacional actualmente es mucho más fácil y amplio que hace quince o veinte años, cuando estaba limitado por la disponibilidad de recursos e información. Los futuros lingüistas computacionales, sin embargo, tienen por delante el mismo desafío que los pioneros: conseguir que los ordenadores nos entiendan hablando nuestra propia lengua.

Con respecto a la elaboración del libro, el autor quiere expresar su agradecimiento a una serie de instituciones y personas. En cuanto a las primeras, debo mencionar a los grupos de investigación y proyectos en los que he participado: Universidad Autónoma de Madrid (tanto en el proyecto Eurotra como en el Laboratorio de Lingüística Informática), el Centro de Investigación de IBM en Madrid, la Universidad de Nueva York (grupo PROTEUS) y la Universidad Politécnica de Madrid (grupo ARIES de la ETSI de Telecomunicaciones). Todos ellos me han proporcionado un ámbito humano y científico inmejorable, lo que ha convertido la investigación en Lingüística Computacional en un placer (a pesar de los agobios de la búsqueda de financiación).

Por otra parte, este libro se ha escrito sobre la base de dos cursos impartidos en la Universidad de Granada, en 1994 (a estudiantes postgraduados) y en 1997 (a profesores de la Universidad Taras Shevchenko de Kiev). Agradezco sinceramente al Departamento de Lingüística General de Granada la confianza y la oportunidad de preparar dichos cursos, pues ha sido fundamental para encontrar el "tono" introductorio de este manual.

He dejado para el final los agradecimientos personales. En primer lugar, a F. Marcos Marín, quien me dio la oportunidad de empezar a trabajar en Lingüística Computacional a mediados de los ochenta, nada más acabar mis estudios de licenciatura. Théophile Ambadiang, José Miguel Goñi, José Carlos González, Ralph Grishman, José María Guirao, Catherine Macleod, Cristina Olmeda y Brian White han leído y comentado fragmentos del manuscrito, y sus observaciones juiciosas han servido para depurar el texto original. Como se suele (y se debe) decir en estos casos, sólo el autor es responsable de los errores, de contenido y forma, que permanezcan.

1.

Delimitación del campo de la Lingüística Computacional

Esta disciplina trata básicamente de dos cosas: *lenguas naturales* y *ordenadores*. Muchas líneas de investigación comparten ambos objetivos, aunque desde perspectivas diferentes. Hay una tradición de más de cuarenta años –los primeros proyectos datan de los cincuenta y la *Association for Computational Linguistics*, ACL, se fundó en 1962– que ha ido modelando las técnicas, los métodos y las aplicaciones de tal manera que a finales de los noventa contamos con una buena perspectiva para establecer sus límites con otras áreas de conocimiento.

El punto de partida será una serie de cuestiones básicas: ¿Qué es la Lingüística Computacional (LC)? ¿Es equivalente a Procesamiento del Lenguaje Natural (PLN) y a Ingeniería Lingüística? ¿Qué diferencia, si es que existe, hay entre Lingüística Computacional y Lingüística Informática? Como es habitual también en otras áreas de conocimiento de reciente creación y en pleno desarrollo, nos enfrentamos al problema de la delimitación del objeto de estudio y a la identificación terminológica.

Siguiendo a Grishman (1986), se puede definir la *Lingüística Computacional* como "el estudio de los sistemas de computación utilizados para la comprensión y la generación de las lenguas naturales". Una definición equivalente nos proporciona Allen (1995) para Procesamiento del Lenguaje Natural: "El objetivo de esta investigación es crear modelos computacionales del lenguaje lo suficientemente detallados que permitan escribir programas informáticos que realicen las diferentes tareas donde interviene el lenguaje natu-

ral". Por tanto, Lingüística Computacional y Procesamiento del Lenguaje Natural tratan de lo mismo: del desarrollo de programas de ordenador que simulan la capacidad lingüística humana. Esta definición permite distinguir LC-PLN, por una parte, de la Inteligencia Artificial y, por otra, de la Lingüística Informática.

La Inteligencia Artificial (IA) se encarga de codificar en un programa facultades cognitivas como la inferencia, la toma de decisiones, la adquisición de conocimiento experto, etc. En este sentido, la LC es una parte integrante de la IA, de la misma forma que para muchos lingüistas la Lingüística es parte de la Psicología por tratar de una de las capacidades cognitivas por excelencia, el lenguaje. De hecho, las aproximaciones al PLN desde la IA adoptan una perspectiva más global: el tratamiento de la estructura lingüística (sintaxis y semántica, básicamente) se integra como un módulo de entrada/salida dentro de un sistema compuesto por más conocimientos.

Por ejemplo, pensemos en un sistema experto para consultar el fondo de una biblioteca o la compra de libros por internet. Si el programa está dotado de algún módulo de PLN debería permitir consultas sencillas utilizando expresiones como "quiero novelas rusas, traducidas al castellano, opcionalmente al inglés o al alemán". Obviamente, la base de datos y el sistema de búsqueda que proporciona los resultados de la consulta no formarían parte del módulo de procesamiento del lenguaje natural. Por otra parte, si el sistema experto no contara con un módulo PLN, es probable que no interpretara correctamente la petición. Dependería, por ejemplo, de si tiene algún medio de saber que "castellano" y "español" en este contexto son sinónimos, o que "al inglés o al alemán" tienen sobreentendido (o elidido) el participio "traducidas". Perfectamente se puede diseñar un programa que sólo identifique información relevante (en nuestro ejemplo, "novelas rusas", "castellano", "inglés", "alemán"), pero reconocer esas palabras-clave no implica que el programa haya analizado la secuencia escrita: la petición original realizada en español incluye unos matices ("opcionalmente", X "o" Y) que tienen que interpretarse lingüísticamente para proporcionar una información adecuada a la consulta mencionada. Lo que el usuario pide no es que se le muestren todas las posibilidades que permitan las combinaciones de "novelas rusas", "castellano", "inglés" y "alemán", sino una lista ordenada en primer lugar por novelas rusas en español, y a continuación novelas rusas en inglés y novelas rusas en alemán. Parece muy difícil que una búsqueda no basada en el procesamiento lingüístico pueda producir ese resultado. Por tanto, restringimos la definición de sistema PLN a todo módulo o programa que verdaderamente procese estructura lingüística, no simplemente palabras.

Análogamente, se podría afirmar que la LC es una parte de la *Lingüística Informática*, si entendemos esta última como la disciplina que abarca todo

uso de ordenadores con relación al lenguaje y las lenguas. Aquí, se incluirían no sólo los sistemas que simulan el lenguaje humano, sino todo tipo de programas y herramientas informáticas que ayudan en el estudio de las lenguas y de la Lingüística. Por ejemplo, programas para consulta de corpus y diccionarios electrónicos, programas para elaboración de listas de palabras, frecuencias y concordancias, programas para la comparación automática de versiones del mismo texto, etc. Al principio, toda investigación lingüística que implicara el uso de ordenadores se consideraba LC, pero el tiempo ha ido especializando las tareas y los términos. Como señalan Gazdar y Mellish (1989) actualmente la parcela del conocimiento que trata de la investigación

CUADRO 1.1. Ordenadores en la investigación lingüística y literaria, según Butler (1985).

Áreas de conocimiento	*Aplicaciones*
Investigación literaria y estilística	Análisis cuantitativos de textos literarios (listas de palabras, índices, concordancias); análisis métricos, sintácticos y semánticos sobre textos etiquetados manualmente; asignación de autoría y localización cronológica.
Lexicografía	*Corpus electrónicos y bases de datos* lexicográficas; lematización automática, ordenación alfabética; edición e impresión.
Edición de textos	Edición crítica de textos literarios: colación automática de variantes.
Enseñanza y aprendizaje de lenguas	Estudios sobre corpus (p. e., frecuencias) para establecer prioridades en el diseño de métodos pedagógicos; presentación de los materiales pedagógicos en formato electrónico (aprendizaje asistido por ordenador).
Simulación del procesamiento del lenguaje humano	Sistemas de análisis y generación de lenguas naturales; sistemas basados en la escritura y en el habla; traducción automática; sistemas de síntesis y reconocimiento de habla.
Lingüística descriptiva basada en corpus	Corpus en formato electrónico, anotados morfológica y sintácticamente.
Análisis computacional de datos no textuales	Programas estadísticos aplicados a la sociolingüística, enseñanza de lenguas, dialectología.

lingüística y literaria con medios informáticos no se considera parte de la LC, y, por tanto, no se tratará en el libro. Los lectores interesados pueden consultar Marcos Marín (1994) o contactar con las siguientes asociaciones: *Association for Literary and Linguistic Computing* o *Computers and the Humanities*. El cuadro 1.1 presenta un panorama de las perspectivas diferentes en las que se usan los ordenadores en la investigación lingüística y literaria, basado en el libro de Butler, *Computers in Linguistics*. Es interesante comprobar en este caso cómo las aplicaciones no han cambiado mucho desde la publicación del libro en 1985, aunque sí la calidad de los programas ahora existentes comparados con los de hace más de una década.

Por último, tampoco se debe identificar LC con *Ingeniería Lingüística* o *Industrias de la lengua*. Ambos términos se han puesto de moda en los noventa a raíz de su utilización en los programas marco de investigación de la Unión Europea. Por Ingeniería Lingüística se entiende toda aquella aplicación potencialmente comercial que implique el uso de nuevas tecnologías y lenguas. En este sentido, se incluye la edición electrónica (diccionarios, libros, periódicos), los productos multimedia, etc. Por supuesto, también tiene cabida todo sistema PLN comercial (traducción automática, corrector gramatical, reconocedor de habla...). Sin embargo, no podemos incluir una parte importante de la investigación en LC: aquella que no tiene como primer objetivo la comercialización de un producto, como es el caso de numerosos proyectos en universidades y centros de investigación públicos y privados.

En resumen, y por oposición a las otras disciplinas, la LC trata de la construcción de sistemas informáticos que procesen estructura lingüística y cuyo objetivo sea la simulación parcial de la capacidad lingüística de los hablantes de una lengua, independientemente de su carácter comercial o de investigación básica.

1.1. El modelo computacional del lenguaje

El enfoque que predomina en LC es el que entiende que el lenguaje es un proceso comunicativo donde emisor y receptor procesan determinada información en función de un conocimiento lingüístico y un conocimiento del mundo (pragmático) compartido (Winograd, 1983). La tarea del lingüista computacional es reflejar la organización y funcionamiento de las estructuras y procesos lingüísticos, por una parte, y de las estructuras y procesos cognitivos, por otra. Esto implica la modelización tanto de la competencia como de la actuación lingüística, así como la inclusión de factores extralingüísticos (el conocimiento del mundo) en el modelo. Contrástese con los objetivos de gran parte de la Lingüística Teórica: estudio de la competencia (conocimiento lingüístico que los hablantes tienen de su lengua) y el rechazo de la

actuación y de las cuestiones extralingüísticas en el proceso comunicativo. De ello hablaremos más extensamente en el apartado 1.2.

La característica esencial de un modelo es que nos permite inferir algo acerca de la cosa modelada. El modelo computacional del lenguaje presupone una *modelización matemática* previa de la lengua en cuestión: "si el objeto matemático *A* es un modelo del fenómeno lingüístico *B*, entonces *A* puede someterse a la investigación deductiva y, como *A* es un sustituto o imitador parcial de *B*, los resultados extraídos tendrán cierta significación con respecto de *B*" (Marcus *et al.*, 1978). Dado que para cualquier fenómeno lingüístico es posible encontrar una gran variedad de modelos matemáticos y que cada uno de estos modelos proporciona un conocimiento parcial sobre el fenómeno en cuestión, se plantean dos decisiones complejas:

1. Emplear el *método más apropiado para cada caso.*
2. Combinar distintas aproximaciones para evitar la omisión de aspectos esenciales.

A lo largo del libro se analizarán ejemplos variados de cada una de las estrategias. En líneas generales, se puede decir que en los primeros tiempos se prefería utilizar un único enfoque, aunque últimamente se están imponiendo los modelos mixtos.

Básicamente, hay dos tipos de modelos matemáticos del lenguaje:

a) *Modelos simbólicos* (también conocidos como *algebraicos* o *axiomáticos*): construidos a partir de la teoría de conjuntos y de la lógica matemática. Son sistemas formales axiomáticos compuestos por un conjunto de unidades (símbolos) y de reglas (que establecen las combinaciones entre las unidades). Se postulan unas propiedades generales sobre los elementos así como sus relaciones y, a partir de estos axiomas, se obtienen nuevas propiedades de manera deductiva. Estos modelos intentan reflejar la *estructura lógica del lenguaje*. Los ejemplos más conocidos son las gramáticas generativas, las gramáticas categoriales y las gramáticas de dependencias.
b) *Modelos probabilísticos*: (también *estadísticos* o *estocásticos*), desarrollados a partir de la Teoría de la Información y la estadística. Estudian las lenguas como un conjunto de sucesos que presentan una determinada *frecuencia*: cada *fonema*, cada *morfema*, cada *categoría sintáctica*, cada sintagma, cada significado tiene una cierta probabilidad de aparecer en un determinado contexto. Los modelos probabilísticos se fundamentan sobre los datos recogidos en los corpus lingüísticos. Cuanto más variado y mayor sea el número de datos utilizados mejor será el modelo, de ahí que se los conozca también como modelos

cuantitativos. Los primeros ejemplos aparecieron en los años cincuenta y sesenta (diccionarios de frecuencias, estudios estilísticos de autores) y actualmente han recuperado el interés gracias al desarrollo tecnológico que permite tratar enormes cantidades de datos con ordenadores y programas de acceso fácil a cualquier investigador.

Antes de continuar es necesario hacer algunas observaciones:

1. Normalmente, para construir un modelo matemático de un fenómeno lingüístico, se parte de una descripción proporcionada por las gramáticas y estudios puramente descriptivos y no matemáticos. De ahí la importancia y el reconocimiento a toda la labor de observación lingüística anterior. Lo tratamos en el apartado 1.1.1.
2. La lingüística matemática no cubre todo el campo de estudio del lenguaje, pero es particularmente eficaz en los aspectos cuantitativos y formales (apartado 1.1.2.)
3. El desarrollo tanto de la Lingüística Matemática y de la Informática en los últimos treinta o cuarenta años ha permitido que se utilicen los ordenadores para "simular" el funcionamiento de nuestra capacidad lingüística, estableciéndose una analogía entre los programas y la forma en que comprendemos y producimos emisiones verbales (apartado 1.1.3).

1.1.1. Necesidad de descripciones previas a la modelización

CASO PRÁCTICO
Traducción de una descripción gramatical
a una descripción formal

Ejemplo 1: regla fonológica de nasalización de las vocales del español

Daremos primero la descripción proporcionada por Alcina y Blecua en su *Gramática Española* (1975):

> En los casos siguiente la consonante nasal impregna de nasalidad a la vocal anterior:
>
> - En posición inicial absoluta: [õnθe] (once).
> - Vocal situada entre consonantes nasales: [mãno] (mano).

Su formalización en una regla dependiente del contexto (de este tipo de reglas hablaremos en el capítulo 3) quedaría:

$$[V_o] \rightarrow [V_n] / \{ (\# \_) \mid (N \_ N) \}$$

donde:

V_o	es Vocal oral.
V_n	es Vocal nasal.
$\rightarrow$	"se realiza como".
/	"en el contexto".
{ ... }	"diferentes opciones en disyunción" separadas por el símbolo " | ".
(...)	una opción.
#	"posición inicial absoluta".
N	es Nasal.
_	es la posición que ocupa en el contexto la unidad mencionada.

Ejemplo 2: reglas sintácticas de la concordancia interna en español

Dice el *Esbozo*: "La concordancia es en nuestra lengua la igualdad de género y número entre adjetivo o artículo y sustantivo". Este enunciado se puede expresar formalmente con una regla sintagmática independiente del contexto aumentada con una estructura de rasgos:

$$\begin{array}{ccccccccc} SN & \rightarrow & (Det) & & (Adj) & & N & & (Adj) \\ \begin{bmatrix} Num = \alpha \\ Gen = \beta \end{bmatrix} & & \begin{bmatrix} Num = \alpha \\ Gen = \beta \end{bmatrix} & & \begin{bmatrix} Num = \alpha \\ Gen = \beta \end{bmatrix} & & \begin{bmatrix} Num = \alpha \\ Gen = \beta \end{bmatrix} & & \begin{bmatrix} Num = \alpha \\ Gen = \beta \end{bmatrix} \end{array}$$

La regla debe interpretarse de la siguiente manera: un SN se reescribe como un determinante (opcional), un adjetivo (opcional), un nombre (obligatorio) y un adjetivo (opcional). Los paréntesis indican opcionalidad. Las fórmulas entre grandes corchetes representan información de las categorías sintácticas en formato de rasgo. Un rasgo es un par "atributo = valor". En estas reglas sólo aparecen dos rasgos "Gén(ero)" y "Núm(ero)". Para sus valores, se utilizan variables (α y β), que indican que todos los elementos de la regla llevan necesariamente el mismo valor para el rasgo en cuestión, independientemente de cuál sea éste (por ejemplo, los valores para Num pueden ser o "singular" o "plural"). Este tipo de codificación formal es típico de las gramáticas de unificación, que trataremos en el capítulo 3.

En estos ejemplos se ha visto cómo descripciones expresadas mediante una lengua natural (aprovechando el carácter metalingüístico del propio lenguaje natural) se han traducido a descripciones expresadas mediante una lengua artificial. Las lenguas artificiales se caracterizan por tener una sintaxis y una semántica bien definidas y sin ambigüedad, lo que no ocurre siempre con las lenguas naturales, donde una expresión puede interpretarse a veces de diferentes maneras.

1.1.2. La imposibilidad de agotar el estudio del lenguaje con medios matemáticos y computacionales

CASO PRÁCTICO
Literatura creada por ordenador

Hace treinta años se pusieron de moda los programas que creaban poesía al azar. A pesar de que los programas en sí mismos son bastantes triviales, en su momento causaron sensación (aunque nada comparable al programa ELIZA, que imitaba a un psiquiatra). Su estructura es la siguiente:

1. *Creación de patrones:* se pueden extraer a partir de un poema concreto, o imitando un modelo poético como el soneto o el haiku (forma japonesa). Los patrones son fragmentos de la estructura del poema que se van a sustituir en cada proceso de generación de un poema.
2. *Creación de una base de datos*, con distintas construcciones que son intercambiables en un mismo punto del patrón. Así por ejemplo, podemos agrupar los distintos datos por categorías sintácticas, por expresiones, etc. El requisito es que cada dato asignado a un tipo de patrón esté en relación paradigmática con los otros datos de ese patrón, es decir, que pueda ocupar la misma distribución dentro de la estructura oracional.
3. *Utilización de una función aleatoria:* se necesita de algún procedimiento que escoja al azar los datos y los distribuya en las posiciones correspondientes en los patrones.

Naturalmente, algunos de los poemas producidos tienen cierto atractivo poético, pero no podemos considerar que estos programas reproducen la creatividad poética. Construir un sistema experto que imite a un poeta es una tarea todavía no conseguida, porque para ello habría que adquirir el conocimiento del poeta para escribir poemas. Como todos los que han escrito alguna vez poesía saben, ese conocimiento no es fácil de expresar conscientemente, y mucho menos de formalizar. Lo mismo se podría decir de un

programa que escribiera novelas policíacas, o cualquier otro tipo de género literario.

La clave de ello está en que cualquier obra literaria es más que la suma de sus partes, parafraseando a Aristóteles. No basta con tener una estructura y cambiar sus elementos constituyentes por otros equivalentes. Una obra de creación literaria es un sistema complejo: cuando uno lo descompone en partes y las estudia por separado (digamos, los personajes, la trama, el estilo, el ambiente, etc.) su resultado, a lo sumo, es una interpretación parcial del contenido que transmite la obra. Es probable que aunque desmenuzáramos la obra en muchísimos componentes su reconstrucción nunca agotaría las interpretaciones posibles, como demuestran los millares de comentarios críticos sobre Shakespeare o *El Quijote*. Esta situación supone un gran atractivo para los estudiosos literarios, pero hace sospechosa (afortunadamente) toda pretensión de crear un crítico literario automático o un novelista computacional. La creación literaria no parece un fenómeno formalizable lógica o estadísticamente, al menos en sus aspectos más interesantes.

1.1.3. La mente humana y los ordenadores

En cuanto a la comparación entre la *mente humana* y un *ordenador* destacan dos factores:

a) Ambos son procesadores de información que pueden manipular símbolos y realizar procesos complejos, incluyendo inferencia, aprendizaje de conocimiento nuevo y toma de decisiones, a partir de conocimiento anterior almacenado.
b) La experimentación con el ordenador nos permite manipular, probar y contrastar nuestros modelos sobre la mente/cerebro. Podemos llegar a explicar las regularidades de los fenómenos lingüísticos como consecuencia de nuestra investigación con simulaciones por ordenador.

El punto de partida de la LC es que tanto los ordenadores como el cerebro humano pueden procesar símbolos (entidades con un significante y un significado relacionados por convención). A diferencia de otros sistemas simbólicos (por ejemplo, el álgebra o la notación musical) que pueden ser utilizados por el hombre y las computadoras, el lenguaje nos permite comunicar y entender gran variedad de información (ideas, percepciones, sentimientos, imágenes, creencias, hipótesis, etc.). Uno de los objetivos de la LC es intentar hacer explícito el conocimiento lingüístico a través de la simulación por ordenador.

La utilización de ordenadores como mecanismos que imitan a otros mecanismos es una de las ideas centrales de la Inteligencia Artificial. Sus orígenes se pueden rastrear incluso antes de la aparición real de los ordenadores: un artículo clásico de Turing de 1937, donde trata la construcción de un computador universal que pueda imitar a cualquier otro mecanismo que compute siempre que cuente con una memoria ilimitada (la conocida *máquina de Turing*). Por tanto, desde los orígenes de los ordenadores, la analogía entre mente y computadora se establece en la estructura lógica, no en la base física o biológica.

La manera habitual de implementar la simulación es dividiendo el proceso simulado en componentes. Por ejemplo, cualquier sistema PLN está organizado en diferentes módulos: reconocimiento léxico y morfológico, análisis sintáctico, interpretación semántica y pragmática. La mayoría de estos sistemas, sin embargo, utilizan una estrategia lineal que no se corresponde con el procesamiento simultáneo y en paralelo que realiza nuestro cerebro. Sin entrar en la discusión de la existencia real de dichos componentes, parece que hay muchas evidencias (sobre todo a partir de los experimentos de Marslen-Wilson en los ochenta) de que consultamos simultáneamente diferentes componentes lingüísticos en el momento de procesar una emisión lingüística. Esto ha estimulado la aparición de una aproximación inspirada en el funcionamiento neuronal, que probablemente represente la manera más cercana a la simulación del cerebro por parte de una computadora.

Esta nueva corriente, conocida por conexionismo o procesamiento distribuido en paralelo, supone una visión alternativa radicalmente diferente a la presentada por el paradigma mayoritario que utiliza la metáfora "mente como ordenador". En el capítulo 5 se introducirán sus planteamientos esenciales.

1.1.4. Información analógica e información digital

Si entendemos el lenguaje básicamente como un mecanismo de transmisión y codificación de información, podemos encontrar otra analogía con los ordenadores, que son otro tipo de mecanismo para manejar información.

Pensemos en el proceso de adquisición de información del entorno por parte de un agente cognitivo, sea éste un cerebro o una máquina. Este proceso se caracteriza por transformar lo continuo en discreto. Supongamos que las propiedades de un objeto se manifiestan a través de algún tipo de señal. La información analógica se caracteriza por ser una señal continua, mientras que la información digital es una señal discreta.

Estos términos se pueden aplicar metafóricamente a cualquier tipo de información. Veamos un ejemplo intuitivo: si le contamos a una persona que

no haya leído *El Quijote*, que trata de "las aventuras de un loco que se cree caballero andante" estamos haciendo una *digitalización* extrema del contenido de la novela de Cervantes: obviamente *El Quijote* trata de muchas más cosas. De la lectura "continua" de la novela, nosotros hemos extraído una información que hacemos explícita de manera discreta. Con esta información nuestro interlocutor puede hacer poco (sólo lo que pueda inferir de conceptos como "loco" y "caballero andante"). Si en lugar de eso le proporcionamos la novela, tendrá la oportunidad de extraer mucha más información que la que nosotros le hemos dado, y con toda probabilidad diferente en parte a la que nosotros hemos extraído de la lectura del libro. Nuestro ejemplo es demasiado radical, por lo que conviene aclarar algunos puntos.

La digitalización es un término técnico que se aplica al proceso por el cual se traduce de un código complejo de cualquier tipo a un código binario, que es muy sencillo porque emplea únicamente combinaciones de 0 y 1 (*bits*) de determinada longitud.

La digitalización puede tener muchos grados, y su definición depende del número de bits de que conste cada elemento del código. Si observamos las figuras donde se muestra la conversión de una onda analógica en digital y la representación de una A impresa de manera convencional frente a la proporcionada por una impresora digital, podemos entender que si la definición es baja, el contorno es muy fragmentado. En cambio, se han conseguido resoluciones digitales tan elevadas (en discos compactos y en impresiones láser) que apenas se distinguen de sus modelos analógicos originales. Si lo aplicamos a nuestro (exagerado) ejemplo de *El Quijote*, es como si en lugar del resumen tan lacónico le proporcionaramos a nuestro oyente un resumen muy detallado y riguroso de cada capítulo. Una edición crítica, de alguna forma, también es "digitalizar" información (es decir, hacerla discreta y explícita). En este sentido tendríamos que hablar de aumento de información, no de pérdida. En cualquier caso, utilizamos la dicotomía analógico/digital en un sentido muy amplio.

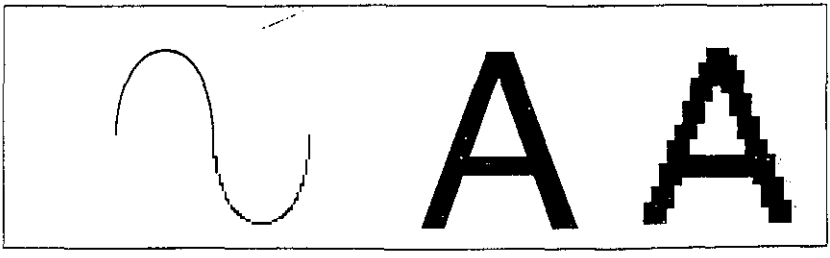


Figura 1.1. Representaciones analógica y digital de una onda y de un carácter tipográfico.

Siguiendo a Devlin (1991), podemos decir que la extracción o adquisición de información implica una conversión de información analógica en digital. Este proceso se divide en dos etapas:

1. *Percepción*, durante la cual el agente cognitivo (cerebro o máquina) accede directamente a la información del entorno por medio de algún sensor.
2. *Cognición*, durante la cual el agente selecciona elementos específicos de información del *continuum*. Es cuando se produce la conversión analógico-digital.

Desde este punto de vista, los sistemas de PLN son el producto de una digitalización en diferentes etapas que empieza en la recogida de datos lingüísticos, su descripción, formalización y, finalmente, codificación en un programa concreto. El carácter reduccionista inherente de la digitalización explica por qué nunca podremos obtener los mismos resultados que se obtienen en las emisiones lingüísticas naturales.

Como en todo sistema digital, la clave para acercarse al modelo analógico está en el refinamiento del muestreo.

Recogida de datos	→	Descripción	→	Formalización	→	Codificación en un programa

Figura 1.2. Proceso de digitalización de la información lingüística.

1.2. Relación entre Lingüística Teórica y Lingüística Computacional

A pesar de que ambas comparten el mismo objetivo –la descripción y explicación de los procesos lingüísticos– tienen enfoques y métodos bastante diferentes.

1.2.1. Enfoques

La *Lingüística Teórica* (y en concreto las gramáticas generativas) se ha centrado en la *competencia de los hablantes* (*conocimiento lingüístico* que les permite entender y producir oraciones gramaticales y rechazar las agramaticales), en los *universales lingüísticos* (principios gramaticales aplicables a todas las lenguas) y en el descubrimiento de la *teoría gramatical más sim-*

ple y más restringida formalmente que sea capaz de dar cuenta de las lenguas naturales. Con estos tres objetivos teóricos principales, los lingüistas pretenden explicar los *mecanismos innatos* del lenguaje que permiten a los hablantes aprender y utilizar su lengua tan fácilmente. Sus esfuerzos por evaluar teorías alternativas les conducen a veces a estudiar oraciones peculiares que algunos lingüistas computacionales considerarían patológicas (Grishman, 1986).

La *Lingüística Computacional* se ocupa del desarrollo de programas para tratar una lengua natural, pero estos sistemas tienen generalmente una aplicación limitada a un dominio restringido (técnico, científico, legal, administrativo, etc.) por lo que sólo procesan un subconjunto, llamado también *sublengua*. Los lingüistas computacionales están dispuestos a aceptar soluciones aproximadas que cubran la mayor parte de las oraciones más frecuentes y a conformarse con un sistema que no pueda tratar algunas oraciones peculiares. Por otra parte, la exigencia de construir sistemas que funcionen en situaciones comunicativas reales les ha llevado a buscar el tratamiento del *proceso total del lenguaje*, es decir, tanto la competencia como la actuación. Necesitan tratar el uso lingüístico real, que está cargado de factores extralingüísticos, ignorados tradicionalmente por los lingüistas teóricos. En Lingüística Teórica es frecuente explicar unos cuantos ejemplos de oraciones y dar por hecho que la teoría funcionará de forma análoga en el resto de casos similares. Los lingüistas computacionales, sin embargo, están obligados a tratar todos los casos que aparezcan y saben que, aunque sean muy parecidos, pueden presentar diferencias que requieran modificaciones a la regla propuesta. Por último, los lingüistas computacionales recurren a otras disciplinas que estudian el lenguaje desde otros puntos de vista como la Psicología, la Lógica y la Inteligencia Artificial para encontrar solución a estos problemas.

1.2.2. Métodos

Desde hace más de treinta años el paradigma racionalista predomina en *Lingüística Teórica*. Se recurre típicamente a la *introspección* como método de obtención de los datos lingüísticos para el desarrollo de la teoría gramatical: el lingüista actúa como hablante ideal que reconoce las oraciones gramaticales y agramaticales basándose en su competencia y eliminando los factores extralingüísticos que se dan en las situaciones de uso real. La investigación se centra en contrastar hipótesis que se han establecido *a priori* siguiendo un modelo deductivo. Sin embargo, hay que destacar que en los últimos años el enfoque empiricista, que parte de datos objetivos para elaborar teorías, está consiguiendo importantes avances gracias a la aplicación

de los ordenadores en disciplinas como la Lingüística basada en corpus, donde se trabaja con colecciones extensas de datos lingüísticos orales o escritos.

Aunque, como en toda ciencia, el rigor y la explicitud son requisitos de cualquier gramática formal, en la práctica muchos modelos teóricos no están especialmente interesados en la construcción de gramáticas concretas y extensas, con todos los fenómenos tratados hasta el menor detalle.

En una *gramática computacional*, sin embargo, esto es el punto de partida: todas las construcciones que puedan aparecer en una aplicación real tienen que ser explícitamente codificadas en el programa, pues de otra forma no serán reconocidas por el sistema. La explicitud es un rasgo necesario en cualquier gramática computacional. Por otra parte, aunque el lingüista computacional utiliza básicamente su competencia para construir la gramática, también recurre constantemente a *datos empíricos* (extraídos de usos lingüísticos reales como textos o consultas directas a hablantes). Trata de ajustarse lo más posible a las situaciones de uso. En los últimos años, con el avance de los proyectos empiricistas, se han llegado a crear sistemas en los que las reglas gramaticales se han inferido de forma automática a partir de los datos, como veremos en el capítulo 5.

En resumen, ambas disciplinas lingüísticas se complementan, pues estudian el lenguaje desde enfoques distintos. La Lingüística Teórica está interesada primordialmente en la *elegancia* de la teoría, mientras que la Lingüística Computacional está interesada en la *eficacia* de su sistema. Por otra parte, a pesar de que la Lingüística Computacional ha surgido de la Lingüística Matemática, el hecho de haberse ido especializando en otro tipo de objetivos y problemas ha marcado un cierto distanciamiento entre ambas, parecido al que existe entre la Lingüística Teórica y disciplinas lingüísticas como la Psicolingüística o la Sociolingüística, que tienen relación con otras ciencias sociales. La diferencia esencial es enfocar el estudio o bien desde la competencia o bien desde la actuación. Esto hace que una solución teórica no implique necesariamente una solución al problema correspondiente en la Lingüística Computacional. Como en otras áreas de la ciencia, se necesita un esfuerzo considerable para pasar de una teoría formal elegante a una teoría computable.

Esa búsqueda de una implementación eficiente ha llevado a los lingüistas computacionales a explorar teorías gramaticales que fueran, por una parte, muy explícitas en la descripción de los fenómenos y, por otra, lo más simples posible desde una perspectiva formal y computacional. El ejemplo más conocido son las gramáticas de unificación y rasgos (Shieber, 1986; Moreno Sandoval, *en preparación*). A principios de los ochenta surgió un nuevo tipo de gramáticas generativas que tenían como características distintivas el uso generalizado de rasgos en la descripción y la operación de unificación de estructuras de rasgos como principal mecanismo para manejar la información. En realidad su utilización comenzó primero en la LC con el modelo de

M. Kay *(Functional Unification Grammar)*, pero rápidamente se extendió a teorías lingüísticas como LFG *(Lexical Functional Grammar)*, GPSG *(Generalized Phrase Structure Grammar)* o HPSG *(Head-driven Phrase Structure Grammar)*. Posteriormente, varias gramáticas computacionales se han inspirado en alguna de estas teorías. Hasta la fecha es el mayor punto de confluencia entre la LT y la LC.

A pesar de esta separación entre ambas perspectivas lingüísticas, parece que en el futuro el acercamiento será mayor, sobre todo si en LT se acepta el carácter experimental de la simulación por ordenador para validar hipótesis, como ocurre en otras ciencias.

1.3. Aplicaciones de la Lingüística Computacional

Para terminar de perfilar el contenido de esta disciplina, es interesante hacer una presentación rápida de sus principales aplicaciones prácticas. Estas aplicaciones se pueden agrupar en varios bloques, como se verá a continuación.

1. Sistemas que tratan de emular la capacidad humana de procesar lenguas naturales. Dentro de este grupo, las aplicaciones más importantes son:

   — *Traducción automática*. Fue una de las primeras metas que se fijó la lingüística computacional y es también una de las tareas más complicadas a las que se enfrenta. Se trata de tomar oraciones (o textos completos) en una lengua, que se denomina lengua fuente, y producir de forma automática una traducción a otra lengua, la lengua meta. En los sistemas más complicados, suele haber más de un par de lenguas. La traducción puede partir de textos escritos o hacerlo a partir de emisiones orales, en cuyo caso necesita contar con un modelo que reconoce e interpreta la voz.
   — *Recuperación y extracción de información*. Son dos aplicaciones muy relacionadas, aunque no idénticas, cuyo objetivo es tratar la información almacenada en las grandes bases de datos textuales. La *recuperación de información* se ocupa de tomar la consulta de un usuario a una base de datos y elegir entre todos los textos que se tienen archivados aquellos que mejor responden a las condiciones de búsqueda planteadas. Cuanto mayor sea el número de textos y más diversos sean los temas de los que tratan, más difícil será responder con exactitud. Por eso, hay que diseñar sistemas que "entiendan" realmente la pregunta y que sean capaces de reconocer qué documen-

tos de la base de datos tocan ese tema y cuáles no, algo que un sistema simple basado en palabras clave no puede hacer. Por su parte, la *extracción de información* pretende "leer" grandes cantidades de texto (por ejemplo, historiales médicos, noticias de agencia, resúmenes de bolsa), reconocer la información importante contenida en ellos y trasladarla a un formato predefinido para que pueda ser tratatada y recuperada con mayor facilidad.

— *Interfaces hombre-máquina.* Son sistemas pensados para facilitar las relaciones entre los usuarios y el ordenador. Su objetivo es que el usuario pueda dirigirse a la máquina en su lengua natural (por ejemplo, el español) en lugar de tener que utilizar lenguajes informáticos, instrucciones complicadas o menús que restringen las posibilidades a unas pocas opciones. Este contacto entre el hombre y la máquina se puede hacer por escrito o directamente a través de la voz, para lo que se necesita incorporar al sistema un módulo que entienda la voz humana (por ejemplo, durante una comunicación telefónica).

2. Sistemas que ayudan en las tareas lingüísticas. Este segundo grupo está formado por herramientas que pueden ser utilizadas por los lingüistas para facilitarles ciertas tareas complejas. Algunas aplicaciones de este tipo son:

— *Herramientas de análisis textual.* Su objetivo es determinar frecuencias de aparición de ciertas palabras o construcciones en los textos, distinguir las concordancias, realizar estadísticas, encontrar regularidades y, en general, liberar al lingüista de ciertas tareas engorrosas que son necesarias para realizar un análisis textual riguroso.

— *Herramientas para manejo de corpus.* Las aplicaciones más destacadas son los etiquetadores categoriales, que asignan a cada palabra del corpus su categoría sintáctica correspondiente, y los analizadores sintácticos, que permiten disponer de una gran colección de datos lingüísticos estructurados.

— *Bases de datos lexicográficas.* El campo de la lexicografía es uno de los que más puede beneficiarse de la gran capacidad de almacenamiento que ofrecen los ordenadores. Por un lado, se puede trasladar un diccionario convencional a un soporte magnético, lo que facilita un acceso más rápido y flexible a la información. Por otro, se pueden crear diccionarios con técnicas propias de la lexicografía computacional que contengan información de varios niveles de descripción lingüística codificados según un sistema de rasgos coherente y completo. También tiene cada vez mayor interés la creación de *bases de datos terminológicas*, que contienen infor-

mación sobre dominios específicos –medicina, ingeniería, informática, arquitectura– que utilizan y crean continuamente nuevos términos técnicos que necesitan ser actualizados. En ocasiones, son plurilingües para facilitar la traducción exacta y constante de los términos entre distintas lenguas.

3. Programas de ayuda a la escritura y composición textual. Las aplicaciones comprendidas en este grupo han sido ampliamente desarrolladas y cualquier usuario habitual de un procesador de textos está familiarizado con ellas.

   — *Correctores ortográficos.* Casi todos los procesadores de textos disponen de una herramienta que revisa el texto escrito y detecta errores de ortografía. La corrección puede ser totalmente automática o con la ayuda del usuario. Para que sean efectivos, deben tener un nivel mínimo de conocimiento lingüístico: análisis morfológico, separación de sílabas...
   — *Correctores sintácticos y de estilo.* Consituyen un paso más allá de los correctores ortográficos, permitiendo detectar fallos de concordancia y oraciones incompletas o incorrectas así como defectos estilísticos. Estas aplicaciones necesitan cierto grado de conocimiento lingüístico y suelen llevar a cabo un verdadero análisis sintáctico. Sin embargo, normalmente no toman las decisiones por sí mismas, sino que se limitan a detectar y sugerir posibles correcciones al usuario.

4. Enseñanza asistida por ordenador. Éste también es un campo de aplicaciones en continua expansión y que tiene varias vertientes. La más importante es la de los *programas educativos para la enseñanza de lenguas extranjeras.* El objetivo es utilizar las tecnologías de procesamiento de lenguas naturales para ayudar a los alumnos en la adquisición de lenguas distintas a la materna. Suelen hacer uso de recursos multimedia, combinando texto, imágenes y voz. Muchos realizan un verdadero análisis sintáctico, ya que plantean ejercicios gramaticales y de composición corrigiendo posteriormente las respuestas.

1.4. Ideas principales del capítulo

La Lingüística Computacional (Procesamiento del Lenguaje Natural) trata de la construcción de sistemas informáticos que procesen realmente estructura lingüística y cuyo objetivo sea la simulación parcial de la capacidad lin-

güística humana. Ésta es la principal diferencia con respecto a otras disciplinas que también relacionan lengua y ordenadores, como la Lingüística Informática o la Ingeniería Lingüística.

La Lingüística Computacional está obligada a mantener un rigor formal que permita el procesamiento del lenguaje por el ordenador. Para ello, parte de un modelo matemático del lenguaje, que puede ser de tipo simbólico o probabilístico.

La Lingüística Teórica y la Computacional tienen bastantes diferencias en cuanto a sus enfoques, métodos y objetivos. La LT se centra en analizar la competencia de los hablantes, utiliza la introspección como principal fuente para obtener sus datos y suele llegar a sus conclusiones mediante métodos deductivos. Sus principales objetivos son conseguir una teoría gramatical, simple, elegante, restringida y que dé cuenta de los universales lingüísticos. La LC está interesada en el uso lingüístico, utiliza datos procedentes de situaciones comunicativas reales y sus métodos de investigación pueden ser tanto deductivos como inductivos. Su objetivo final es obtener un sistema que funcione, es decir, que procese estructura lingüística de una forma computacionalmente eficiente.

La Lingüística Computacional es una disciplina aplicada. Entre sus usos principales figuran, entre otros, la traducción automática, los interfaces hombre-máquina, la recuperación y extracción de información y los correctores sintácticos y estilísticos.

1.5. Ejercicios

1. ¿Por qué los sistemas expertos normalmente utilizan menús para establecer sus búsquedas, en lugar de permitir expresiones en lengua natural? Téngase en cuenta el coste de producir un módulo de entrada y salida que procese una lengua natural y los resultados finales ofrecidos por el sistema experto.
2. Asóciense las aplicaciones con sus respectivas áreas de conocimiento.

   — *Aplicaciones:* Programas de autoría de texto basados en frecuencias de uso. Aprendizaje de lenguas por ordenador. Correctores automáticos gramaticales y de estilo. Codificación de archivos digitales.
   — *Áreas de conocimiento:* Lingüística Computacional, Inteligencia Artificial, Lingüística Informática, Ingeniería Lingüística.

3. ¿Cómo se traduciría la información de un diccionario impreso a un diccionario computacional? Practíquese con varios diccionarios para comprobar que la falta de sistematización en la microestructura de las entra-

das del diccionario es responsable de la dificultad de traducción. La sistematización también es información. Compruébese la hipótesis de que "cuando la información no existe en un modelo dado, difícilmente se puede añadir cuando se traduce a otro modelo".

4. ¿Qué aplicaciones nuevas existen actualmente con respecto a las mencionadas por Butler en 1985?
5. Analícese el siguiente fragmento sobre uno de los peligros más frecuentes de toda modelización matemática:

> Cualquier modelo, invariablemente, lleva consigo rasgos no sólo de la cosa que está siendo modelada, sino también de la estructura en la que está hecha ese modelo, y a menudo es bastante difícil decidir si un aspecto particular del modelo dice algo acerca de la cosa modelada o de la teoría que subyace al modelo (Devlin, 1991: 75; trad. nuestra).

6. Asóciense diferentes aspectos de la realidad lingüística con el método que mejor los pueda tratar (simbólico, estadístico, no matemático). Compárese la lista proporcionada con la que pueda confeccionarse al terminar la lectura del libro.
7. Apliquese la metáfora de la información analógica y digital para explicar las diferencias entre Lingüística Teórica y Lingüística Computacional. ¿Cuál de las dos supone mayor digitalización de los fenómenos lingüísticos reales? ¿Cómo se puede medir el refinamiento de la digitalización en cada caso?
8. ¿Qué aplicaciones pueden tener más interés y futuro? Hágase una lista por orden de preferencia. ¿Hay alguna relación clara entre interés social de la aplicación y la complejidad de los problemas que tiene que resolver?

2.

Panorama general de la Lingüística Computacional

2.1. La Lingüística Computacional como ciencia aplicada

Construir un sistema de procesamiento de una lengua natural, al igual que cualquier sistema informático, es fundamentalmente *una obra de ingeniería* (Grishman, 1986): integra diferentes tipos de conocimiento (sintáctico, semántico, discursivo, pragmático) y su utilización eficaz dentro de un programa. Hay ciertas técnicas generales en la construcción de sistemas que se suelen utilizar para facilitar el trabajo.

La *modularidad* consiste en dividir el sistema en componentes relativamente independientes. La división de un problema nos permite abordar los subproblemas individualmente. Ésta es la estrategia clásica de los niveles lingüísticos separados. Se basa en la idea de que cualquier fenómeno lingüístico complejo se puede descomponer en una serie de procesos más simples, los cuales pueden ser hasta cierto punto independientes unos de otros. La estratificación y modularidad es muy importante para que un sistema sea flexible y ampliable, ya que asegura que los efectos provocados por las reglas puedan ser localizados más fácilmente. Permite, además, que diferentes investigadores se concentren en problemas concretos y que las modificaciones en cada módulo no impliquen cambios generales en todas las partes del sistema. La clave está en una integración efectiva de los distintos niveles, mediante la definición de lo que cada componente espera a la entrada y de lo que produce como salida.

Esta estrategia se puede aplicar fácilmente al lenguaje porque el conocimiento lingüístico se puede organizar en diferentes niveles o componentes. Esquemáticamente, un sistema de procesamiento de lenguas naturales tiene, o puede contener, al menos parte de los siguientes módulos:

- *Conocimiento fonético y fonológico:* trata de las realizaciones acústicas. Es un componente que sólo aparece en los sistemas de reconocimiento y síntesis de habla, que se tratarán en el apartado 2.4.
- *Conocimiento morfológico:* tiene que ver con la formación interna de las palabras. Muchos sistemas no lo tratan, sobre todo los más antiguos, o lo incluyen dentro de la gramática, formando un conjunto de conocimiento morfosintáctico. El tratamiento computacional de la morfología no es esencial en algunas aplicaciones y en algunas lenguas con morfología flexiva muy sencilla (como es el caso del inglés). En cambio, para una lengua rica en formas flexionadas (como el español, el alemán o el finés) el procesamiento morfológico es obligado si queremos evitar la expansión innecesaria de formas plenamente flexionadas en el diccionario, ya que si no se cuenta con un procesador morfológico necesariamente cada palabra tiene que ser codificada con sus rasgos morfosintácticos en el lexicón.
- *Conocimiento sintáctico:* es una de las partes básicas de cualquier sistema, pues se encarga de reconocer la estructura de las oraciones.
- *Conocimiento semántico:* la otra parte imprescindible de cualquier sistema, ya que sin la semántica no podríamos asignar significado a las estructuras analizadas, y sin interpretación nuestra tarea carecería de utilidad práctica.
- *Conocimiento contextual,* a veces denominado *pragmático:* incluye toda información no intrínsecamente lingüística que influye en el procesamiento e interpretación de emisiones lingüísticas. Podemos hacer una partición en dos:

  a) *Conocimiento del discurso:* aquí se tratan los aspectos de interpretación afectados por las oraciones emitidas anteriormente. En concreto, este conocimiento se utiliza para interpretar los pronombres anafóricos y los aspectos temporales.
  b) *Conocimiento del mundo:* incluye todo el conocimiento conceptual del mundo que tienen en cuenta los hablantes cuando se comunican mediante una lengua. Este conocimiento sirve para comprender mucha información sobreentendida pero no expresada explícitamente en las oraciones.

El análisis de cualquier oración sigue los pasos que se muestran en la figura 2.1.

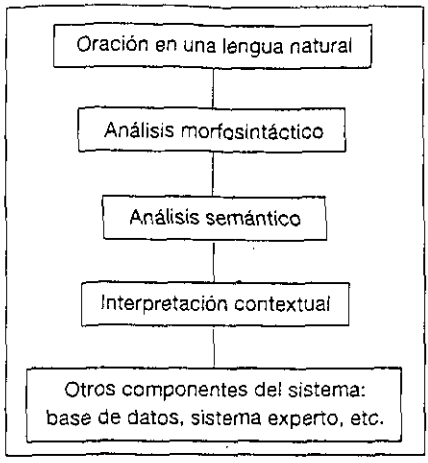


Figura 2.1. Etapas en el procesamiento computacional del lenguaje.

La utilización de *formalismos gramaticales* es otra técnica muy extendida para simplificar sistemas complejos. Codificar directamente el conocimiento gramatical en un programa es una labor muy costosa además de suponer el hecho de que los lingüistas computacionales sean también programadores (lo cual no es generalmente cierto ni deseable). La práctica común es separar el trabajo de los lingüistas y el de los informáticos. Los primeros cuentan con un formalismo (equivalente a un lenguaje de programación de alto nivel como Prolog o LISP) en el que escriben las reglas de la gramática y el diccionario. Los informáticos se encargan de la programación del algoritmo de análisis y de la integración de los distintos módulos. Lo habitual es que un compilador (un programa que traduce código escrito en un lenguaje de alto nivel a código máquina) convierta la gramática escrita en el formalismo concreto a un código ejecutable más eficiente. De esta forma se puede desarrollar la gramática y hacer modificaciones sin implicar cambios generales en el sistema. Una ventaja adicional es que al escribir la gramática en un formalismo facilitamos la comprensión y documentación de la gramática. En la actualidad existen entornos de desarrollo de sistemas PLN que facilitan enormemente el trabajo de los lingüistas computacionales. Algunos de ellos, como la plataforma ALEP, desarrollada con fondos de la Unión Europea, están disponibles de manera gratuita (aunque con ciertas restricciones).

En resumen, la creciente y continua ampliación de la cobertura de los sistemas ha obligado a dividir y especializar la compleja tarea de desarro-

llar un sistema PLN, pero también ha redundado en la disponibilidad de herramientas y técnicas que faciliten su elaboración.

2.2. Análisis y generación

En la figura 2.1 no aparece ninguna flecha que indique el sentido del procesamiento. Éste puede partir de la oración para llegar al componente no lingüístico (lo que se conoce por *análisis* o *reconocimiento*). O a la inversa, empezar con una información no expresada en términos lingüísticos (por ejemplo, una forma lógica) que se convierte en una oración en lengua natural (denominado *generación* o *síntesis*). En palabras de Grishman (1986), el análisis lingüístico es una tarea de "traducción de una lengua natural a una representación semántica formal", mientras que la generación es la operación inversa: "la traducción de una representación semántica formal a una lengua natural".

El sentido del procesamiento depende de la aplicación. Por ejemplo, un programa de diálogo con una base de datos contará con un componente que reconozca las preguntas de los usuarios y otro que genere respuestas. Un sistema de traducción automática, por otra parte, se compone de un analizador de la lengua fuente y de un generador de la lengua meta. También nos podemos encontrar con sistemas que sólo tengan el componente de reconocimiento, por ejemplo, la mayoría de los sistema de recuperación o extracción de información del texto.

¿Hacen falta dos tipos de conocimiento diferentes, uno para análisis y otro para generación, o se puede utilizar la misma fuente de información lingüística para ambas tareas? Desde el punto de vista teórico es preferible utilizar los mismos componentes lingüísticos. El argumento fundamental es la economía del sistema, ya que tener reglas diferentes para cada tarea es más costoso (tanto computacional como psicológicamente) que utilizar el mismo conocimiento pero distinta forma de procesarlo.

En la práctica, sin embargo, muchos investigadores en LC no están de acuerdo con la eficiencia de un conocimiento único: los problemas que surgen en la generación a menudo son distintos de los del análisis, incluso para el mismo tipo de dominio.

El argumento de los partidarios de distintos tipos de conocimiento se basa en el hecho de que somos capaces de reconocer más oraciones de las que podemos elaborar. Esta afirmación es bastante intuitiva, aunque no está demostrado si esa descompensación se debe a factores puramente lingüísticos o de otro tipo, como conocimiento del mundo o competencia comunicativa. Al llevar este hecho al tratamiento informático de una lengua, se dan dos tipos de expectativas:

1. Se espera que el programa reconozca la mayoría de las paráfrasis posibles de un mensaje, mientras que basta con que genere una de las formas posibles y gramaticales del mensaje.
2. Cuando reconocemos oraciones somos bastante tolerantes a pequeños errores gramaticales, pues lo que nos interesa es interpretar el mensaje. En cambio, cuando producimos oraciones, somos en general más cuidadosos para evitar pérdida de información. Análogamente, esperamos que el programa sea flexible y reconozca oraciones incluso con problemas de buena formación. Cuando genera mensajes, esperamos que estén bien construidos y sobre todo que sean fáciles de entender.

El segundo punto ha motivado que muchos sistemas utilicen gramáticas diferentes para cada tarea. Las reglas de análisis sobreanalizan, es decir, reconocen más oraciones de las propiamente gramaticales; y las reglas de generación infrageneran, es decir, producen menos oraciones de las posibles, pues seleccionan sólo unos pocos modos de expresar ideas y eventos.

2.3. Reconocimiento y síntesis del habla

El procesamiento del lenguaje natural se puede dividir en dos grandes tipos de aplicaciones, en función de si tratan lengua en forma escrita u oral. Son mucho más usuales los primeros, por varias razones: el reconocimiento acústico es más difícil, por motivos que veremos enseguida; además, el formato escrito es mucho más fácil de codificar digitalmente. Sin embargo, el atractivo de los sistemas de habla es considerable: la expresión oral sigue siendo el modo prioritario de comunicación lingüística, y además es la manera más cómoda de favorecer la interacción entre el hombre y la máquina, sobre todo porque libera las manos para otras tareas.

Los sistemas de habla no han empezado a ser suficientemente rápidos y fiables hasta los años noventa. Precisamente el avance de los modelos probabilísticos ha contribuido a que aparecieran los primeros sistemas comerciales eficientes. Los avances tecnológicos en reconocimiento acústico y velocidad de procesamiento de los ordenadores son el otro factor que está convirtiendo los sistemas de habla en una de las áreas con más futuro dentro de la LC.

¿Por qué el reconocimiento del habla es más difícil que el reconocimiento de texto? Hay grandes diferencias en los problemas y técnicas que se emplean en cada tarea:

1. Mayor grado de incertidumbre en una entrada acústica que gráfica. En el lenguaje escrito el sistema conoce exactamente las palabras que

tiene que procesar. Por supuesto, tiene niveles de incertidumbre provocados por la ambigüedad y la imposibilidad de registrar todas las unidades y reglas de una lengua, pero al menos cuenta con la certeza de conocer los elementos que forman la oración. Sin embargo, los sistemas de habla sólo cuentan con una estimación de lo que se ha dicho, precisamente porque tienen que identificar las unidades fonológicas para luego convertirlas a información morfosintáctica.

2. El lenguaje oral es bastante diferente al escrito desde el punto de vista estructural. Se puede comprobar comparando oraciones de un corpus textual con las de uno oral: su estructura, longitud y enlace con otras unidades es marcadamente distinto.
3. Ambas formas de lenguaje tienen información adicional propia del medio. El texto contiene información tipográfica (*cursiva*, **negrita**, etc.) y gráfica (esquemas, figuras, etc.) que proporcionan una valiosa ayuda para la interpretación del documento. La emisión oral contiene, por otra parte, considerable información prosódica (especialmente la entonación) que contribuye decisivamente en la interpretación del mensaje.
4. El lenguaje oral presenta numerosas rectificaciones y repeticiones, ya que el emisor se corrige a sí mismo o intenta parafrasear lo dicho anteriormente, en un intento de mejorar la comprensión por parte del receptor. Los textos no suelen contener este tipo de estructura fragmentada y redundante.
5. En el lenguaje oral se suele mantener una rica interacción entre emisor y receptor, que incluye mucha información contextual que no suele aparecer en el lenguaje escrito.

Al igual que en los sistemas de procesamiento textual, los sistemas de procesamiento de habla pueden dividirse en dos grandes grupos: los de *reconocimiento* (o análisis) y los de *síntesis* (o generación). Análogamente, los primeros son más numerosos que los segundos.

Otra clasificación importante es la que divide a los sistemas de reconocimiento en dos tipos:

1. *Reconocedores de palabras aisladas*, que analizan una palabra cada vez. Son la mayoría de los sistemas que se encuentran en el mercado.
2. *Reconocedores de habla continua*, que analizan emisiones completas. Estos sistemas empiezan a conseguir resultados prácticos.

Por último, en función de dos parámetros como el tamaño del vocabulario reconocido y el número de hablantes que pueden utilizarlo, los sistemas pueden ir desde los más sencillos (pocas palabras y un solo hablante) a los más complejos (varios miles de palabras y distintos hablantes).

Las aplicaciones de estos sistemas abarcan desde el dictado (es decir, transcripción de voz a texto, muy útil para profesionales como médicos o psicólogos) hasta sistemas de traducción automática (por ejemplo, para cambio de divisas).

La arquitectura básica de un sistema completo de reconocimiento y síntesis de habla se muestra en la figura 2.2, basada en Allen (1995). Los elementos encerrados en cajas representan distintos componentes del procesamiento, mientras que los otros elementos son los resultados que produce cada componente.

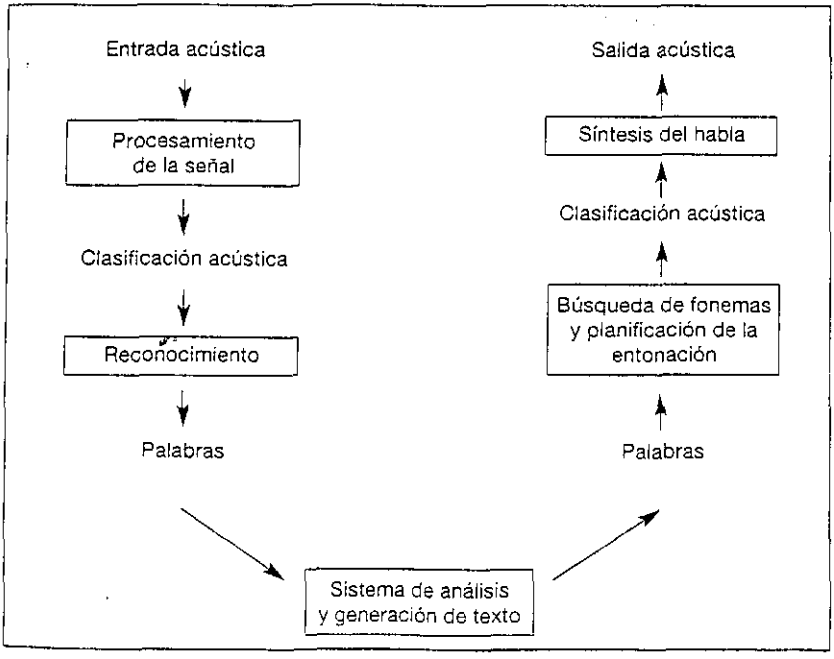


Figura 2.2. Arquitectura general de un sistema de habla.

La figura 2.2 resalta la parte de reconocimiento y producción de habla, ya que comprime todo el tratamiento lingüístico no acústico en una única etapa. Asumimos que la figura 2.1 se incluye en "Sistema de análisis y generación de texto".

Las distintas etapas se pueden resumir de la siguiente manera:

1. *Procesamiento de la señal acústica:* los sonidos producidos por un hablante son convertidos a un formato digital mediante un conversor analógico-digital. La señal digital entonces es procesada para extraer información fonémica. La señal en sí misma es demasiado complicada de representar directamente, sobre todo por su gran variabilidad: dos emisiones de la misma palabra producidas por el mismo hablante pueden mostrar representaciones muy distintas. La clave está en identificar una serie de rasgos acústicos que representen los aspectos menos variables de cada sonido. La técnica principal de reconocimiento de rasgos significativos es utilizar algún tipo de análisis de espectro, como por ejemplo la Transformada Rápida de Fourier. Entre los rasgos que se emplean podemos mencionar los tres formantes de las vocales, que sirven para identificarlas con bastante fiabilidad. La señal continua tiene que ser fragmentada en diferentes segmentos y la salida es una clasificación acústica de la información reconocida. Toda esta fase es puramente física.
2. *Reconocimiento de fonemas:* esta etapa implica procesamiento lingüístico, ya que consiste en proyectar los rasgos acústicos a sus correspondientes fonemas. La tarea es compleja porque no sólo no se da una correspondencia directa entre ambos, sino que se produce una gran variación (acento, tonode voz, etc.), tanto entre diferentes hablantes como en un mismo locutor. Hay que añadir además los ruidos que puedan interferir y la naturaleza continua y no discreta de cualquier emisión oral. En esta fase se suelen aplicar técnicas de modelos markovianos (que veremos en el capítulo de los modelos probabilísticos) para identificar la secuencia más probable de palabras. Esta secuencia es el resultado final del reconocedor de habla, que sirve como entrada, ya en formato de texto, al sistema de análisis.
3. *Análisis de la emisión y generación de una respuesta:* en esta fase se produce todo el procesamiento. Si la aplicación es un programa de dictado automático, entonces se suele analizar para encontrar algún error estructural o semántico. Recordemos que la transcripción reconocida sólo es la combinación más probable, pero puede contener errores que a veces se corrigen con información lingüística adicional. Si la aplicación es un sistema de traducción automática, entonces es necesario que el sistema genere una respuesta.
4. *Síntesis del habla:* una vez obtenida una oración para ser transmitida, antes de ser sintetizada tiene que someterse a una planificación sobre los rasgos prosódicos y la entonación que acompañarán a los sonidos. Los aspectos prosódicos son muy importantes para la compre-

sión de la lengua oral, ya que determinan, entre otras cosas, si se trata de una pregunta o de una aserción. La caracterización de la prosodia tiene mucha importancia en un sistema de síntesis, hasta el punto de que su ausencia provoca emisiones muy artificiales y difíciles de entender.

En el capítulo 5 trataremos cómo los modelos probabilísticos han permitido el avance tecnológico de estos sistemas en los últimos diez años. Un dato: algunos sistemas actuales tienen una tasa de precisión en el reconocimiento de emisiones acústicas del 95%. Es decir, reconocen 19 de cada 20 palabras.

2.4. Breve historia de la Lingüística Computacional

2.4.1. Problemas iniciales

Los trabajos pioneros en el campo del PLN se dieron en los años cincuenta y principios de los sesenta y, concretamente, en traducción automática (TA). Estos primeros sistemas fueron un fracaso por varias razones:

1. El bajo nivel de desarrollo de la Lingüística Matemática y de los conocimientos sobre el lenguaje. Recordemos que Chomsky introdujo sus ideas revolucionarias a finales de los cincuenta, y que hasta bien entrados los sesenta nó hubo un desarrollo aceptable de la Gramática Generativa.
2. Los primeros ordenadores no estaban diseñados para trabajar en PLN. Además de la potencia y capacidad de procesamiento, el principal obstáculo con que se enfrentaron los primeros lingüistas computacionales fue que no existían lenguajes de programación capaces de trabajar, de manera fácil y eficiente, con palabras y símbolos. Incluso hoy en día los ordenadores representan las palabras como secuencias de bytes (combinaciones de 0 y 1). Pero en la actualidad contamos con ordenadores y lenguajes de programación que permiten tratar problemas bastante complejos sobre estructuras lingüísticas.

En los primeros tiempos de la informática lo que mejor sabían hacer los ordenadores era contar. De ahí que las primeras aplicaciones útiles de los ordenadores a la investigación lingüística fueran en el campo de la Lingüística Estadística. Por ejemplo, el estudio del léxico de autores, confección de diccionarios de frecuencias, elaboración de índices y concordancias, etc.

Los primeros sistemas de *Traducción Automática* asumían muy poco o ningún conocimiento lingüístico teórico. La idea fundamental era que las len-

guas son sistemas de códigos. Por lo tanto, traducir de un código lingüístico a otro consistiría en sustituir un elemento de la lengua fuente por su equivalente en el código de la lengua objeto. Es decir, sustituir una palabra o un grupo de palabras por sus correspondientes. Con este planteamiento tan simplista, que negaba el hecho de que el significado es lo esencial en cualquier comunicación lingüística, no se pudo llegar muy lejos. En 1966 un famoso informe de la National Academy of Sciences de los EE UU afirmaba que con la tecnología existente en la época no se podía alcanzar ningún éxito en traducción automática. Esto supuso un largo período de inactividad investigadora en TA hasta que a finales de los setenta, gracias a los considerables avances experimentados en lingüística y computación en las dos décadas anteriores, empezaron de nuevo los grandes proyectos en TA.

Gazdar y Mellish (1989) señalan que la mayoría de los avances en PLN en los setenta y ochenta se debió a un *cambio en el enfoque teórico y práctico en Informática*. Al principio los ordenadores se usaron fundamentalmente para el cálculo aritmético. Los primeros lenguajes de programación, como FORTRAN, obligaban al programador a pensar en términos de números y especificar el código a un nivel muy cercano al código máquina. Los avances en lenguajes de programación supusieron la aparición de lenguajes de alto nivel, como Prolog, que permiten al programador especificar las instrucciones en términos de conceptos orientados al problema, en nuestro caso la manipulación de símbolos complejos como palabras u oraciones. La existencia, además, de compiladores que traducen los programas escritos en lenguajes de alto nivel a código máquina o lenguajes de bajo nivel ha liberado a los programadores de la costosa tarea de volver a replantearse el tratamiento de un problema particular en otro lenguaje de programación más eficiente.

2.4.2. Los años setenta: primeros sistemas funcionales

El momento que marca la diferencia entre los primeros sistemas de PLN y los desarrollos actuales se produjo con la aparición del programa SHRDLU de Winograd, en 1971. Su contribución más importante fue demostrar que un ordenador podía entender una lengua natural en un dominio restringido (un universo de "bloques"). SHRDLU podía interpretar preguntas y órdenes sencillas, así como realizar inferencias, explicar sus acciones y aprender nuevas palabras, todo ello integrado en un programa de ordenador, lo que hasta la fecha no se había conseguido.

En los sistemas de los años setenta la gramática (es decir, el conocimiento lingüístico) y el parser (el procedimiento que compara las oraciones de entrada con las reglas gramaticales) estaban entremezclados dentro del programa. Las técnicas más extendidas para escribir gramáticas computacionales

fueron las *Redes de Transición Recursiva* (RTN, su acrónimo en inglés) y sus derivadas, las *Redes de Transición Aumentadas* (ATN, su acrónimo en inglés). Una red de transición es una representación de una gramática regular o de estados finitos. Está formada por una serie de estados unidos por unos arcos etiquetados con símbolos terminales (palabras o categorías léxicas). Hay un estado inicial y uno o varios finales. Podemos interpretar una red de transición como un mapa con distintos caminos que nos permite encontrar las expresiones gramaticales de una lengua. Una oración pertenece a la lengua definida por la red si hay un camino desde el estado inicial hasta cualquier estado final. Lo veremos detalladamente en el siguiente capítulo.

Una red de transición recursiva permite que las estructuras que se repitan (especificando que un elemento puede aparecer 0 o más veces) puedan ser expresadas como subredes, de tal forma que es posible construir grandes redes de forma modular. Las redes de transición recursivas permiten tratar de forma natural, clara y eficiente desde el punto de vista computacional, las estructuras recursivas tan habituales en las lenguas naturales, lo que no puede hacer una simple red de transición de estados. Las redes de transición aumentadas (ATN) se desarrollaron para tratar problemas típicos de las gramáticas transformacionales. Son RTN con procedimientos para poner en funcionamiento restricciones gramaticales y generar una estructura profunda. Es decir, se han añadido procedimientos para establecer condiciones y guiar más eficientemente al parser. En realidad las ATN son un lenguaje de programación para construir analizadores sintácticos (Gazdar y Mellish, 1989) y se caracterizan por su estilo típicamente procedural. Esta manera de escribir programas consiste en especificar paso a paso todas las computaciones que tienen que realizar. Los programas de este estilo son muy rígidos y esto implica que no pueden ser trasladados fácilmente para que traten problemas distintos (un nuevo dominio, una lengua distinta); en otras palabras, son difícilmente reutilizables.

2.4.3. Los años ochenta: lenguajes declarativos y gramáticas no transformacionales

Los años ochenta supusieron un cambio radical en las técnicas utilizadas en los sistemas PLN. En el plano lingüístico, los investigadores empezaron a explorar las ventajas de utilizar formalismos gramaticales más sencillos que las gramáticas transformacionales. En el plano informático, el estilo declarativo se fue imponiendo. Los sistemas se van haciendo más flexibles, pues no están concebidos para un determinado conjunto de problemas o aplicaciones. Son sistemas "portables", que se pueden aplicar a nuevos campos. La declaratividad consiste en proporcionar la descripción de las reglas de una

lengua, independientemente de la forma en que el parser vaya a utilizarlas para analizar las cadenas de entrada. Los sistemas de estilo procedural exigían diferentes gramáticas para generación y análisis. Con el estilo declarativo lo que se persigue es tener un único componente de reglas y utilizarlo de manera diferente según la tarea. Todo ello se consiguió gracias a tres importantes innovaciones: los *formalismos de unificación*, los *lenguajes declarativos de programación lógica* como Prolog, y los *chart parsers*.

Los formalismos de unificación permiten definir gramáticas independientes del contexto aumentadas con rasgos, cuya función es representar la información gramatical que hay en las estructuras sintácticas. En estos formalismos el componente léxico (diccionario) es muy importante ya que asocia cadenas lingüísticas (palabras o morfemas) con su información gramatical. La idea básica es que la gramática contenga reglas sencillas y que sean la información léxica y la unificación (un mecanismo que consiste en combinar la información de los distintos elementos hijos) las que lleven todo el peso del procesamiento.

Prolog es un lenguaje inherentemente declarativo y que permite realizar directamente la unificación, con lo que libera al lingüista de pensar en problemas de procesamiento.

Como consecuencia de la falta de interés por las cuestiones de eficiencia computacional, los sistemas se hicieron más completos pero mucho más lentos y menos robustos. (La "robustez" es un término informático para indicar que el programa responde a problemas inesperados.) Los *chart parsers* son una técnica de desarrollo de analizadores sintácticos que se caracteriza por su capacidad para almacenar resultados intermedios durante el procesamiento estructural. Esta técnica se emplea habitualmente para mejorar la eficiencia de los sistemas declarativos y de unificación.

De estas tres innovaciones hablaremos en el siguiente capítulo.

2.4.4. Los años noventa: ascenso de los modelos probabilísticos

Los años noventa también han supuesto un cambio de tendencia. Los sistemas de los ochenta estaban basados fundamentalmente en el conocimiento gramatical. En la pasada década se consiguió extender apreciablemente la cobertura sintáctica y los dominios de aplicación de los sistemas. A medida que los sistemas se fueron haciendo más complejos se vio más evidentemente que nuestro conocimiento lingüístico actual tiene unos límites: está basado en la competencia del lingüista, que establece un modelo teórico sobre una lengua; pero los sistemas de PLN son ante todo sistemas prácticos que tienen que resolver casos reales de uso (no competencia). Y las situaciones reales distan mucho de ser ideales. Después de una década de fuerte inversión en la

investigación en PLN, los sistemas resultantes no eran capaces de responder de forma eficaz a problemas concretos. Esto ha provocado una doble reacción:

1. *Búsqueda de aplicaciones realistas*, por ejemplo, herramientas de ayuda al escritor como los correctores gramaticales; traducción asistida por el ordenador en lugar de ambiciosos sistemas de traducción automática.
2. *Ampliación de la cobertura del sistema a cualquier tipo de texto*. Los sistemas PLN se han caracterizado siempre por ser programas que se aplican a dominios restringidos, pero la enorme cantidad de información que está accesible actualmente a través de corpus, bases de datos e Internet exige que los sistemas de recuperación y extracción de información sean capaces de discriminar la información relevante. No se busca tanto la calidad como la cantidad de información procesada: es mejor, por ejemplo, tener varias traducciones parciales e imperfectas que ninguna; o recuperar mucha información no relevante si también se incluye la información que andamos buscando.

El objetivo de ambas tendencias es conseguir aplicaciones más eficientes que reporten beneficios directos a la sociedad, después de una fuerte inversión de dinero y esperanzas en las investigaciones de los años ochenta. Este claro giro hacia la parte más aplicada y comercial ha favorecido en gran medida el resurgir de las *técnicas probabilísticas basadas en grandes corpus* de datos lingüísticos. En la actualidad, muchos sistemas incorporan una mezcla de conocimiento declarativo y conocimiento estadístico para mejorar las limitaciones inherentes a cada modelo, sobre todo para resolver el problema de la ambigüedad.

2.5. Ideas principales del capítulo

Los sistemas PLN combinan conocimiento lingüístico a distintos niveles mediante técnicas informáticas. Como cualquier obra de ingeniería informática, se busca la integración modular de cada componente.

La complejidad de los problemas tratados ha obligado a dividir y a especializar cada parte del sistema, y como consecuencia de ello han surgido técnicas y herramientas que faciliten el desarrollo de tales sistemas: lenguajes simbólicos declarativos, formalismos gramaticales y técnicas de *parsing* eficientes.

El procesamiento lingüístico puede tener dos sentidos: partir de una lengua natural y llegar a una representación en una lengua formal (análisis) o la operación inversa, de una información codificada en una lengua artificial obtener su "traducción" a una lengua natural (generación).

Los problemas del reconocimiento y síntesis de habla son bastante diferentes y más complejos de resolver actualmente que los de procesamiento de texto. Las razones son de diversa índole: mayor incertidumbre, más información sobreentendida, redundancia, proyección de una representación acústica a una representación textual.

El desarrollo histórico de los sistemas de PLN ha estado marcado por desarrollos teóricos y técnicos en Lingüística y en Computación. En las últimas décadas la LC ha estimulado la aparición de nuevas ideas en los mencionados campos y se ha convertido en un importante evaluador de teorías.

2.6. Ejercicios

1. Compárese la evolución de la Lingüística Computacional con la de la Lingüística Teórica, destacando los períodos de convergencia y de divergencia entre ambas. Abstrayendo los condicionamientos sociales y tecnológicos; ¿cuál sería la relación ideal entre ambas disciplinas?
2. Analícense los argumentos aportados en favor de dos tipos de conocimiento diferentes para análisis y generación, y propónganse una manera alternativa empleando un único conocimiento para ambos procesos.
3. Si se dispone de algún programa de reconocimiento/síntesis de habla (algunos de ellos se comercializan por un precio muy asequible), clasifíquese según los criterios expuestos. Explórense las posibilidades del programa y determínense sus límites (número de usuarios, número de palabras reconocidas, etc.).

3.

Modelos simbólicos I: fundamentos

3.1. Introducción

Los modelos simbólicos son los predominantes en las ciencias cognitivas, ya sea Lingüística, Psicología o Inteligencia Artificial. Entienden que los procesos mentales, entre ellos el lenguaje, se basan en la manipulación de símbolos. La Lingüística Computacional ha sido un campo completamente dominado por este paradigma, hasta la emergencia de la alternativa estadística en la última década. Los manuales más conocidos de la disciplina (Winograd, 1983; Grishman, 1986; Allen, 1987; Gazdar y Mellish, 1989) tratan exclusivamente de estos modelos. Este capítulo, a pesar de su extensión en comparación con otras partes del libro, es necesariamente más limitado en cobertura que los manuales mencionados. Se ha adoptado este planteamiento para dar más actualidad y diversidad a la exposición incluyendo otros modelos no simbólicos, lo que obliga a buscar cierto equilibrio entre las distintas partes.

3.1.1. Perspectiva histórica

Como adelantamos en el capítulo 1, el simbolismo tiene sus raíces en la Lógica: los sistemas lógicos clásicos consisten en procedimientos para manipular símbolos. Por ejemplo, en la lógica proposicional los símbolos representan proposiciones (u oraciones) y sus conectivas. Los silogismos son sis-

temas de reglas que, mediante la manipulación de símbolos, deducen los valores de verdad de una proposición.

Los sistemas simbólicos son objetos matemáticos definidos a partir de un conjunto de expresiones (los *axiomas*) y un conjunto de reglas (las *reglas de derivación*). Estas reglas transforman (es decir, deducen) los axiomas en expresiones nuevas, los *teoremas*. En breve veremos cómo el reconocimiento de oraciones gramaticales se puede entender como un tipo de deducción de teoremas (las oraciones) a partir de las reglas de derivación.

Serrano (1983) nos recuerda que el origen del fundamento lógico de las teorías formales en Lingüística está en la crisis de los fundamentos de las matemáticas a finales del siglo pasado. Cantor, Frege y Russell contribuyeron decisivamente en la elaboración de una teoría de los sistemas formales simbólicos, cuyos frutos se están recogiendo en la segunda mitad del siglo XX.

Igualmente, gran parte de la Computación, y más concretamente la Inteligencia Artificial, está fundamentada en la Lógica Formal, aunque han conseguido ampliar la perspectiva puramente logicista al desarrollar la idea de que los ordenadores son mecanismos generales de manipulación de símbolos: los programas "inteligentes" no sólo sirven para probar teoremas lógicos o realizar inferencias, sino que pueden jugar al ajedrez, dar consejos de experto o traducir. Desde la perspectiva informática, en las primeras décadas del tratamiento computacional del lenguaje natural se emplearon básicamente modelos simbólicos.

Dentro de la Lingüística, Chomsky fue el primero en introducir de manera sistemática el paradigma lógico formal. Es bien conocido que después de haberse dedicado, por recomendación de su maestro Z. Harris, a estudiar Lógica y Filosofía de la Ciencia, Chomsky revolucionó el campo de la Lingüística con sus propuestas. Entre ellas destacaremos la concepción de lengua como conjunto infinito de oraciones generadas por una gramática, que es definida –por primera vez– como un sistema formal caracterizado por unidades y reglas de buena formación. Este mecanismo gramatical puede concebirse como un autómata que genera todas y sólo aquellas oraciones gramaticales de una lengua. La inclusión de la noción de recursividad, término de orígenes matemáticos, es una de sus aportaciones esenciales, pues explica, junto con la composicionalidad, el carácter eminentemente creativo del lenguaje natural: con medios finitos conseguimos producir y entender infinitas oraciones.

La otra gran aportación de Chomsky, desde la perspectiva formal y computacional, es su famosa jerarquía de gramáticas generativas. Desde su definición a finales de los años cincuenta, es el punto de referencia dentro de la Teoría de las Lenguas formales y autómatas.

Las repercusiones de las ideas de Chomsky tanto en Lingüística Teórica como Computacional son bien conocidas: baste decir que consiguió esta-

blecer un espacio teórico común entre las lenguas artificiales (como las lenguas de las Matemáticas, la Lógica o la Informática) y las lenguas naturales. Precisamente la formalización de las lenguas naturales permitió un enorme avance en el desarrollo de los sistemas PLN. Los primeros sistemas de traducción automática, por ejemplo, eran simples programas que cambiaban una palabra en la lengua fuente por otra en la lengua meta. Se asumía que la traducción consistía en la sustitución de un elemento en un código por el correspondiente en otro código. No trataban ningún tipo de estructura interna, *ni empleaban ningún tipo de conocimiento lingüístico*. La aportación teórica de Chomsky permitió reconocer estructuras de constituyentes mediante la aplicación sucesiva de reglas de reescritura (es decir, reglas que expanden un símbolo en una cadena de símbolos subordinados).

Es curioso, sin embargo, que a pesar de haber sido el pionero de la utilización de autómatas abstractos para especificar gramáticas, y de incluir un componente "computacional" en su último modelo minimista, Chomsky nunca ha estado interesado por implementar sus gramáticas en programas de ordenador.

En los años setenta empezaron a surgir dentro del paradigma generativo modelos alternativos al transformacional. En concreto, la aparición de las llamadas *gramáticas de unificación* (Shieber, 1986; Moreno Sandoval en preparación) vino a rellenar el hueco de la aplicación directa de teorías lingüísticas al tratamiento informático de lenguas. Gran parte de la motivación y el éxito de estas teorías y formalismos se debe a su compromiso real con el componente computacional: son gramáticas que se pueden trasladar fácilmente a un programa. Esto supone, además, una división eficiente de la tarea, pues ya no es necesario que el lingüista computacional sea un especialista tanto en programación como en descripción de lenguas. Cada parte se puede desarrollar independientemente porque hay un lenguaje intermedio común. (El lingüista computacional ideal sigue siendo aquel capaz de realizar eficientemente las dos tareas, pero al menos no se condiciona el éxito del sistema a la existencia de esa *rara avis*.)

En resumen, desde los orígenes de la LC, y tanto en Inteligencia Artificial como en Lingüística, los sistemas formales han sido el paradigma predominante.

3.1.2. Características de los modelos simbólicos

Podemos destacar una serie de puntos:

— En los modelos simbólicos se emplea una *metalengua* para hablar de la realidad, que consiste en un conjunto de especificaciones para escri-

bir gramáticas. Como toda metalengua, sirve para fijar un conjunto de convenciones que beneficia la comunicación de ideas entre los miembros de la comunidad científica.

— Los símbolos como *representación del conocimiento:* se trata de un cuestión filosófica muy antigua. Tanto los racionalistas como los empiricistas defendían que el pensamiento y el conocimiento racional se basan en la manipulación lógica de ideas, concebidas como símbolos. Posteriormente, en la Lógica se ha desarrollado la Teoría de Modelos, que trata los símbolos como mecanismos de representación. Desde esta perspectiva, un *modelo* es un conjunto de entidades en un mundo posible y las proposiciones acerca del modelo son ciertas o falsas en función de las propiedades de las entidades del modelo. Este marco conceptual es la base para la interpretación semántica de numerosos sistemas de PLN.

— *Autonomía de la sintaxis* frente a la semántica: un sistema formal consta de una sintaxis (donde se determinan las condiciones de buena formación) y una semántica (donde se interpreta el significado de las expresiones bien formadas). Típicamente, las reglas de inferencia en un sistema formal permiten concentrarse en la sintaxis del modelo, independientemente de su interpretación. La lógica formal se caracteriza por la inferencia deductiva, que consiste en demostrar si una afirmación (la *conclusión)* se deduce necesariamente de sus *premisas*, es decir, si la conclusión es consistente con las premisas. En un sentido logicista extremo, el análisis sintáctico de una oración se puede entender como la conclusión (o derivación) a la que se ha llegado a partir de unas premisas (las reglas de la gramática). Destaca el hecho de que en los sistemas formales no se determina la validez o falsedad de las premisas, que dependerá del modelo del mundo. Lo característico de este método es que se concentra en la forma o estructura y eso ha favorecido un enorme avance en sintaxis en las últimas décadas.

Por otra parte, los modelos simbólicos, desde un punto de vista psicológico, suelen asociarse con el *mentalismo* o *conceptualismo:* una lengua natural es un sistema mental aprendido y compartido por los miembros de una comunidad lingüística. Este sistema está formado por signos (o símbolos) de naturaleza psicológica que representan objetos de la realidad. La mayor parte de las corrientes lingüísticas de este siglo han sido mentalistas, desde Saussure a Chomsky, con la excepción del estructuralismo americano. Significativamente, las otras opciones al modelo simbólico adoptan una postura contraria al mentalismo. Los *materialistas* o *realistas* entienden que el objeto de estudio de la Lingüística no está dentro de la mente de los hablantes sino en las manifestaciones externas y objetivas, es decir, los datos lingüísticos observables.

3.2. Fundamentos teóricos: las gramáticas formales

Una *gramática formal* es una especificación rigurosa y explícita de la estructura de una lengua. Por tanto, se utiliza el adjetivo "formal" en el sentido de "formalizado". Como anticipamos en el apartado anterior, las gramáticas formales se escriben siguiendo una convención, llamada metalengua o *formalismo gramatical*, que no es otra cosa que una lengua artificial creada para describir lenguas naturales, como el español o el japonés.

Las lenguas artificiales son muy valiosas para la descripción científica. Parece una cuestión aceptada que el contenido conceptual de cualquier ciencia puede ser expresado en un sistema formal. Su utilidad se debe a una serie de factores:

- Están *bien definidas:* el rasgo más destacado de las gramáticas formales es la ausencia de ambigüedad. Todo sistema formal debe establecer una relación inequívoca entre su sintaxis y su semántica.
- Son *rigurosos:* esto reduce los malentendidos y las disputas interpretativas, al tiempo que exigen una explicitud y claridad por parte del científico.
- Facilitan la *evaluación de las hipótesis:* la estructura lógica de los sistemas formales permite comprobar la consistencia de las conclusiones.
- Permiten hacer *predicciones:* análogamente, las conclusiones válidas basadas en premisas verdaderas pueden generalizarse. A su vez, la predicción es un procedimiento para comprobar la validez empírica de la hipótesis. En el caso de las gramáticas formales se puede predecir la gramaticalidad o no de una oración.
- Permiten el *desarrollo de aplicaciones:* cuando el conocimiento formalizado ha sido contrastado empíricamente, se puede utilizar para crear tecnología que resuelva problemas concretos. Los sistema PLN son un ejemplo.

Detrás de los modelos simbólicos formales subyace la creencia racionalista de que cualquier fenómeno no se produce por azar sino por causas que se pueden establecer como leyes o reglas. "El azar es la medida de nuestra ignorancia", decía Poincaré. Esta concepción determinista será revisada cuando tratemos los modelos probabilistas. De momento lo que interesa destacar es que muchos lingüistas piensan que el lenguaje tiene una naturaleza regular y lógica, y eso es lo que tratan de reflejar en sus gramáticas formales. En general, estas gramáticas han demostrado ser eficaces en la descripción y explicación de fenómenos relacionados con la *competencia* (el conocimiento que cada hablante tiene de su lengua materna), aunque su

éxito ha sido menor con aspectos relacionados con la *actuación* (el uso observable de dicho conocimiento).

3.2.1. Tipos de gramáticas formales

Las gramáticas formales más conocidas y utilizadas son las *gramáticas generativas*, propuestas por Chomsky a finales de los años cincuenta. Sin embargo, hay otros tipos de gramáticas que cumplen los requisitos de estar formalizadas rigurosamente. Se presenta a continuación una tipología de las teorías gramaticales más empleadas en LC:

— *Gramáticas generativas*, también conocidas como gramáticas de *estructura de frase* o *sintagmáticas*: incluyen las de estados finitos, las *transformacionales* y las *gramáticas de unificación*. Están constituidas por un conjunto de reglas generativas (o derivativas) que asignan explícitamente la estructura interna de las oraciones. Dichas reglas, llamadas de reescritura, operan sobre dos conjuntos de elementos, no terminales y terminales. Tienen la forma $\alpha \rightarrow \beta$ y se interpretan de la siguiente manera: cualquier símbolo que aparezca en el lado izquierdo de la regla (α) puede ser sustituido por los símbolos que aparezcan en la parte derecha (β). Tanto las gramáticas transformacionales como las de unificación son gramáticas generativas. En su acepción más amplia, gramática generativa es cualquier gramática que defina precisa y explícitamente las oraciones de una lengua (Bach, 1974). En ese sentido, la mayoría de las gramáticas citadas en esta lista pueden considerarse gramáticas generativas, y en especial las gramáticas categoriales y las de adjunción de rasgos. Sin embargo, el término "generativo" se suele emplear para referirse a las gramáticas sintagmáticas. Como son el tipo más extendido en Lingüística y Computación se tratarán más extensamente en los tres apartados siguientes.

— *Gramáticas categoriales*: son modelos basados en la Lógica para la descripción de la sintaxis de las lenguas naturales a partir de la Teoría de Tipos. Lesniewski dio su formulación original en 1929, y posteriormente han sido desarrolladas por Aidukiewicz, Bar-Hillel, Lambek y otros lógicos. Una gramática categorial típica está formada por dos categorías básicas, oración (o) y nombre (n), con las que se forman el resto de categorías derivadas. Así, un verbo es $n \setminus o$, que se interpreta "es un categoría que necesita un n para formar una o" y, más en general, $A \setminus B$ es una categoría compleja donde A indica con qué categoría debe combinarse para formar un nuevo constituyente y B es la categoría resultante de la combinación, o nuevo constituyente.

Sólo existe una regla básica, que establece la concatenación de dos categorías en una nueva. Esta regla se conoce como supresión, cancelación o borrado, y tiene un claro parecido con la operación aritmética de la división y las fracciones, donde hay un numerador y un denominador. El análisis oracional, por tanto, se obtiene mediante la concatenación y supresión de categorías hasta quedarse con el símbolo *o*. Las secuencias no gramaticales son aquéllas donde no se puede aplicar la regla de supresión. Otras características importantes son que todo elemento léxico tiene asignada una categoría y que las categorías derivadas pueden estar formadas a su vez por categorías derivadas y básicas. Este modelo, a pesar de su simplicidad aparente, puede dar cuenta de numerosas estructuras si se le añaden una serie de extensiones formales. Las gramáticas categoriales han influido en modelos semánticos como la Gramática de Montague, y también en modelos sintácticos como GPSG y HPSG (dos tipos de gramáticas de unificación que se tratarán en el apartado de la gramáticas independientes del contexto). Al ser sistemas muy formalizados, su aplicación computacional es fácil y se han utilizado como base para distintos sistemas de PLN.

— *Gramáticas de dependencias:* se trata de una aproximación a la estructura lingüística diferente a la de los constituyentes inmediatos. Sus orígenes se pueden rastrear en la gramática tradicional, donde habitualmente se habla de elementos modificados, regidos o dependientes de otros elementos controladores. La formalización del concepto de dependencia estructural llega con Tesnière en los años cincuenta. Hays en los sesenta desarrolló la gramática de dependencias para ser aplicada en programas de traducción automática. Los conceptos esenciales son los de *núcleo* y *modificador*, así como la subordinación (dependencia) del modificador al núcleo y su concepto inverso, el elemento controlado modifica al núcleo. En general, las gramáticas de dependencias y las independientes del contexto son equivalentes en cuanto a poder expresivo: ambas generan conjuntos de oraciones equivalentes. El principal atractivo de las gramáticas de dependencias es que pueden tratar de una manera natural los constituyentes discontinuos, limitación conocida de las gramáticas sintagmáticas. Casi todas las teorías generativas han introducido los conceptos de núcleo y de rección, influidas por el tratamiento de las gramáticas de dependencias. Desde el punto de vista computacional, el principal problema es que no se ha propuesto una manera sistemática de encontrar la estructura de dependencia correcta para una secuencia de palabras, y por tanto el proceso de *parsing* tiene una naturaleza muy *ad hoc* (Winograd, 1983).

— *Gramática sistémica:* desarrollada por el lingüista británico Halliday, esta teoría insiste en los aspectos funcionales y pragmáticos de la

comunicación lingüística. Winograd (1983) señala que sus raíces están más en la Antropología y la Sociología que en las matemáticas o la Lógica Formal, aunque eso no ha impedido su adaptación a sistemas computacionales. La gramática sistémica ha desarrollado formalmente la noción de elección y la organización de rasgos en sistemas de rasgos interdependientes, o jerarquías. Su interés se debe a que es un precursor de los sistemas computacionales basados en rasgos y en funciones y ha sido utilizada en algunos sistemas de generación, precisamente porque cuenta con un buen formalismo para organizar las elecciones que se realizan a la hora de planificar una oración.

— *Gramática de cadenas lingüísticas de Harris:* teoría desarrollada por el maestro de Chomsky. La gramática está formada por reglas que definen cadenas básicas que combinadas producen cadenas oracionales. Todo elemento básico (por ejemplo, un N o un V) puede convertirse en complejo mediante la adjunción de elementos a la derecha y a la izquierda. Son equivalentes a gramáticas generativas independientes del contexto (véase 3.2.4.) e incluyen un lenguaje especial para codificar restricciones. Este modelo fue uno de los primeros en implementarse en programas de ordenador. El propio Harris lo desarrolló a principios de los años sesenta en la Universidad de Pennsylvania. (Curiosamente, el maestro de Chomsky estuvo mucho más interesado en la aplicación computacional de su teoría que su discípulo.) Este modelo ha sido la base para la elaboración de las gramáticas computacionales de gran cobertura (Sager, 1981; Grishman, 1986).

— *Gramáticas de adjunción de árboles* (TAG): este formalismo eminentemente computacional está siendo desarrollado por Aravind Joshi y sus colegas desde mediados de los años setenta. En lugar de reglas de reescritura de cadenas de símbolos, utilizan reescritura de árboles. Los árboles elementales pueden ser de dos tipos: iniciales y auxiliares. Mediante la composición de distintos árboles elementales se obtienen árboles derivados. Dos son las operaciones de composición de árboles: la adjunción y la sustitución. Desde el punto de vista lingüístico, las gramáticas TAG presentan el atractivo de tener un dominio de localidad amplio, lo que permite expresar fácilmente dependencias lejanas entre distintas categorías. Desde el punto de vista computacional, las TAG se enmarcan dentro de las gramáticas ligeramente *(mildly)* dependientes del contexto (véase el siguiente apartado). En la última década se ha desarrollado una versión del formalismo que utiliza rasgos, lo que permite superar algunos problemas de su formulación original.

3.2.2. La jerarquía de Chomsky y el poder formal de las gramáticas

En los siguientes apartados nos concentraremos en los tipos de grámáticas más empleados en los sistemas computacionales. Empezaremos con una definición más formal de gramática sintagmática o de estructura de frase.

Una gramática sintagmática G está compuesta por cuatro elementos <Vn, Vt, R, O > donde:

- V_n: es un conjunto finito de *símbolos no terminales*. A veces se les conoce como *variables*. En las aplicaciones lingüísticas se corresponden con las categorías sintácticas.
- Vt: es un conjunto finito de *símbolos terminales*. Estos símbolos coinciden más o menos con las palabras de una lengua, si se trata de una gramática de una lengua natural. Son elementos con valor *constante* y se agrupan dentro del diccionario o lexicón.
- R: es un conjunto finito de *reglas*, también llamadas *producciones*. Tienen la forma $\alpha \rightarrow \beta$, donde α y β son cadenas de elementos de V_n y V_t.
- O: es el símbolo inicial. O es un elemento de V_n y tiene que aparecer al menos una vez en la parte izquierda de una regla o producción (es decir, en lugar de α). Representa la unidad superior y en una gramática de una lengua natural suele corresponder con la oración.

Los elementos terminales y no terminales constituyen los *términos* de la gramática, es decir, los elementos sobre los que operan las reglas. Las reglas gramaticales especifican las combinaciones permitidas de términos que dan lugar a cadenas "bien formadas" u oraciones de la lengua. En una gramática generativa las reglas tienen la forma $\alpha \rightarrow \beta$, donde α y β son cadenas de elementos terminales y/o no terminales. Por ejemplo, si utilizamos la convención "mayúsculas para los elementos no terminales y minúsculas para los terminales", estos son algunos ejemplos de reglas gramaticales:

(1a) SN → DET N.
(1b) DET → ART.
(1c) DET → POS.
(2a) ART → el | la | los | las.
(2b) POS→ mi | tu | su | mis | tus | sus.
(2c) N → abanico | hipopótamos | libertades.

Los ejemplos 1a-c sólo contienen elementos no terminales, por tanto, podemos considerarlos reglas de la gramática. En cambio, los ejemplos 2a-c tienen en su parte derecha exclusivamente un elemento terminal: se trata de

elementos léxicos. El símbolo "|" es una convención notacional para indicar que los elementos están en alternancia o disyunción. Sirve para hacer más compacta la descripción ya que, por ejemplo, 2c equivale a:

(3a) N → abanico
(3b) N → hipopótamos
(3c) N → libertades

Lo más habitual y práctico es separar estas reglas de la gramática y agruparlas en un componente aparte, el diccionario o lexicón.

Con las reglas de 1a-c podemos generar sintagmas nominales muy sencillos, como "el abanico", "los hipopótamos" o "tus libertades" (y por supuesto, combinaciones agramaticales como *"tus abanico", * "el hipopótamos" o *"tu libertades"; se tratará esta cuestión en el apartado 3.2.4.).

En general, una gramática G genera una lengua L(G). Existen varios tipos de gramáticas generativas, dependiendo de la forma de las cadenas α y β en las reglas. Cada tipo de gramática tiene reglas con una forma distinta. Las gramáticas de cierto tipo generan lenguas del tipo correspondiente. Dicho de otra manera, el tipo de G determina el tipo de L(G). Chomsky estableció una clasificación de tipos de gramáticas que se ha hecho famosa con el nombre de su creador, la *jerarquía de Chomsky*.

Esta jerarquía está organizada de acuerdo con el *poder generativo débil*. El concepto de poder generativo o formal se utiliza para referirse a la capacidad de predicción de una gramática. En concreto, el poder generativo débil concierne a qué tipo de oraciones puede reconocer la gramática como gramaticales.

Hay cuatro tipos de gramáticas generativas (denominados *tipo 0, tipo 1, tipo 2,* y *tipo 3),* cada uno definido por la clase de reglas que contiene. Se trata de una jerarquía implicativa, de modo que las lenguas definidas por gramáticas del tipo-i incluyen a todas las lenguas de tipo-$(i + 1)$, donde i puede ser 0, 1 o 2. Dicho de otra manera:

$$\text{Tipo } 3 \subset \text{Tipo } 2 \subset \text{Tipo } 1 \subset \text{Tipo } 0$$

es decir, tipo 3 es un subconjunto de (o está incluido en) tipo 2, etc.

Por tanto, las gramáticas de tipo 0 son las más poderosas y las de tipo 3 las más restringidas. Antes de pasar a analizar las características de cada tipo es pertinente saber las razones por las cuales se utiliza esta jerarquía. La cuestión puede formularse de una manera simple: si tenemos cuatro tipos de gramáticas, ¿cuál es la más apropiada para describir formalmente las lenguas naturales? Todo el mundo está de acuerdo en que la respuesta debe conjugar dos propiedades:

1. *Expresividad:* la gramática tiene que ser lo suficientemente poderosa como para abarcar todas las construcciones posibles en las lenguas naturales
2. *No sobregeneración:* la gramática tiene que ser suficientemente restringida para no permitir como válidas construcciones agramaticales.

Estos dos requisitos se deducen de la definición de gramática generativa: mecanismo para determinar *todas* y *sólo* aquellas oraciones gramaticales de una lengua. Estas condiciones son ideales ya que combinan expresividad y restricción al mismo tiempo, algo no conseguido en la práctica. Ahora bien, si todo el mundo está de acuerdo en los requisitos formales, hay gran discrepancia en cuanto a las argumentos a favor de una u otra clase de gramáticas debido precisamente a la dificultad de conseguirlos en una sola teoría. Hay gramáticas que son más expresivas y otras que son más restringidas, y la mayoría busca el equilibrio más eficaz. Hagamos un breve repaso del estado de la cuestión:

— Chomsky defendió en sus primeras obras, especialmente en *Estructuras sintácticas* (1957), la inadecuación de las gramáticas de estados finitos y de las independientes del contexto por falta de expresividad. Ese fue su principal argumento para defender las gramáticas transformacionales, ya que éstas sí son capaces de tratar todos los fenómenos de las lenguas naturales. Este argumento fue muy influyente durante décadas en Lingüística, desterrando del panorama teórico a las mencionadas gramáticas. Sin embargo, en Computación las gramáticas de estados finitos e independientes del contexto siguieron siendo estudiadas por su aplicación a las lenguas artificiales.
— En 1973, en un artículo influyente, Peters y Ritchie demostraron que matemáticamente las gramáticas transformacionales de aquella época eran equivalentes en poder formal a las máquinas de Turing, lo que significaba que con aquellas gramáticas se podía formalizar todo lo que se quisiera formalizar. Es decir, la argumentación ahora iba contra la sobregeneración de las gramáticas transformacionales. Si el poder expresivo es excesivo, entonces hay que aplicar algún tipo de restricción. La noción de *restricción* se utiliza en teoría gramatical para cumplir con el principal objetivo de una teoría lingüística que es proporcionar una explicación adecuada, eligiendo de las potenciales gramáticas que dan cuenta de los fenómenos lingüísticos la de menor poder o más restringida. El concepto de *restricción* está muy relacionado con la idea general aplicable a toda ciencia de conseguir una teoría lo más simple y elegante posible. La consecuencia de esta demostración fue la imposición de limitaciones cada vez más fuertes

al formalismo transformacional hasta el punto de que en la etapa de Rección y Ligamiento, en los años ochenta, las transformaciones se redujeron a una única: Muévase-α. El programa Minimista de los noventa insiste aún más en la reducción del aparato formal.

— Paralelamente a la evolución de las gramáticas transformacionales hacia modelos menos poderosos expresivamente, a principios de los años ochenta surge una serie de investigaciones, encabezadas por la figura de G. Gazdar, que pretende revisar las críticas iniciales de Chomsky a la inadecuación de las gramáticas independientes del contexto. Por una parte, cuestionan la necesidad de tener dos estructuras sintácticas diferenciadas, proponiendo la eliminación de la estructura profunda y consecuentemente de las transformaciones que relacionaban ambas estructuras. Por otra parte, defienden que las gramáticas independientes del contexto pueden dar cuenta de la inmensa mayoría de las estructuras lingüísticas conocidas. De hecho, hasta la fecha sólo se conocen dos lenguas que contengan ciertos fenómenos intratables por las gramáticas independientes del contexto. El dialecto suizo del alemán es una de ellas, como demostró Shieber en 1985.

— Por su parte, desde distintas posiciones computacionales se ha defendido últimamente la adecuación de las gramáticas regulares (o autómatas de estados finitos) para tratar de manera eficaz muchos aspectos de las lenguas naturales. Por ejemplo, Koskenniemi propuso en 1983 el modelo más utilizado hasta la fecha para el procesamiento morfológico. Su Modelo de Dos-Niveles se basa en la utilización de autómatas finitos para reconocer y generar formas gramaticales. También el modelo probabilista más empleado, los n-gramas, no es otra cosa que un autómata con información probabilista (las cadenas de Markov). La argumentación en estos casos no es de naturaleza formal, sino de eficiencia computacional: la gramática más adecuada será la que permita dar cuenta del procesamiento lingüístico en tiempo real, es decir, la gramática más eficiente. Hay una relación lógica, aunque no demostrada, entre gramática más eficiente y gramática menos poderosa. En ese sentido, una gramática independiente será preferible a una transformacional, y una gramática regular lo será aún más.

El lector se preguntará cómo es posible que haya tanta discrepancia en este tema, aparentemente tan esencial desde una perspectiva teórica y computacional: obviamente, el lingüista teórico necesita controlar el alcance de sus predicciones, y el lingüista computacional quiere conseguir el método más eficiente para tratar informáticamente una lengua natural.

La razón más importante, a nuestro juicio, es que no existen tipos puros de gramáticas (de igual forma que hasta la fecha no se han encontrado parámetros tipológicos generales para distinguir distintas clases de lenguas consistentemente y a todos los niveles). En la práctica, las gramáticas formales se van modificando según las necesidades particulares: se introducen restricciones para reducir su poder o se incluyen extensiones para aumentar su expresividad. Como consecuencia, no se puede decidir fácilmente si una gramática pertenece a un tipo o a otro, y sus propiedades matemáticas son más difíciles de conocer. Por tanto, la distancia entre las demostraciones teóricas y las realizaciones prácticas es tan grande que normalmente no se tiene muy en cuenta a la hora de desarrollar un sistema de PLN.

Un hecho que hay que tener en cuenta, por otra parte, es que tampoco todos los lingüistas están motivados de la misma forma por las propiedades formales de las gramáticas que escriben. Noam Chomsky, por ejemplo, que empezó desarrollando la teoría de las lenguas formales, ha ido dejando paulatinamente de interesarse por esta cuestión para concentrarse en otro tipo de adecuación gramatical: la explicación de cómo un niño aprende su lengua. El propio G. Gazdar, que en los años ochenta estaba muy interesado por la propiedades matemáticas de las gramáticas, trabaja en la actualidad sobre aspectos diferentes.

Evidentemente, la adecuación formal y la complejidad computacional han dejado de ser un tema importante, como lo fueron en la década pasada. Esto refleja también la evolución desde una etapa de fuerte teorización a otra más orientada a la obtención de resultados prácticos. Por otra parte, la teoría de las lenguas formales está incompleta, así que probablemente en el futuro volverá a ser tema de intensa investigación.

El cuadro 3.1. recoge los cuatro tipos de gramáticas y su relación con lenguas y autómatas. Un autómata es un mecanismo abstracto que realiza una serie de operaciones sobre una cadena de entrada. Los autómatas se suelen utilizar para decidir si una cadena concreta pertenece a una lengua. Los autómatas se clasifican en clases según el tipo de lengua que reconozcan.

La Teoría de los Autómatas y la Teoría de las Gramáticas formales tienen una estructura similar aunque traten de cosas diferentes –procedimientos y gramáticas, respectivamente–. Eso ha motivado la investigación de las relaciones entre ambas teorías: las gramáticas generan un tipo de lenguas y los autómatas reconocen un tipo de lenguas. El punto de conexión entre gramáticas y autómatas está en las lenguas.

Se adoptará una perspectiva aplicada en los dos apartados siguientes. Se tratarán únicamente los dos tipos menos poderosos porque son los más utilizados y más conocidos en computación.

CUADRO 3.1. Jerarquía de Chomsky.

Tipo	*Gramáticas*	*Restricciones a la forma de las reglas*	*Lenguas*	*Autómatas*
0	Irrestrictas	Ninguna: $\alpha_1...\alpha_n \longrightarrow \beta_1...\beta_n$	Enumerables recursivamente	Máquinas de Turing
1	Dependientes del contexto	La parte derecha contiene como mínimo los símbolos de la parte izquierda: $\alpha \longrightarrow \beta$ / X _ Y o alternativamente: $x\,\alpha\,z \longrightarrow x\,\beta\,z$	Dependientes del contexto	Autómatas linealmente finitos
2	Independientes del contexto	La parte izquierda sólo puede tener un símbolo: $\alpha \longrightarrow \beta$...	Independientes del contexto	Autómatas PDS *(Push Down Store)*
3	Regulares o de estados finitos	La regla sólo puede tener estas dos formas: A —> t B A —> t	Regulares	Autómatas finitos

3.2.3. Gramáticas regulares o de estados finitos

Una aclaración terminológica antes de nada, dado que hay varios nombres diferentes para el mismo tipo de gramática:

— *Gramáticas regulares* en Teoría de las Gramáticas Formales.
— *Autómatas de estados finitos* en Teoría de Autómatas.
— *Redes de transición* en Lingüística Computacional simbólica.
— *Cadenas de Markov* en Lingüística Computacional estadística (aunque en este caso es ligeramente diferente a los anteriores por cuanto que hay probabilidades asociadas a cada estado).

Los dos términos más empleados son el de *autómata de estados finitos* y el de *red de transición*. Una red de transición está formada por nodos o *estados* y *arcos etiquetados*. Cada arco representa una transición entre dos estados. Hay dos clases especiales de estados: los *estados iniciales*, que son los únicos que no reciben arcos procedentes de otros nodos, y los *estados finales*, que son los únicos de los que no parten transiciones a otros estados. En los diagramas, los estados se representan mediante círculos, los arcos median-

te flechas indicando el sentido de la transición. Los estados iniciales se marcan con una pequeña flecha y los estados terminales con un doble círculo.

Las gramáticas de estados finitos solo tienen dos tipos de reglas (Grishman 1986):

$$A \rightarrow tB$$
$$A \rightarrow t$$

donde A y B son elementos no terminales y t es un elemento terminal. Los elementos no terminales se representan gráficamente mediante un nodo, y los elementos terminales son las etiquetas de los arcos. A continuación se construye una pequeña gramática regular del español (cuadro 3.2), cuyo diagrama se muestra en la figura 3.1:

CUADRO 3.2. Pequeña gramática de estados finitos.

Reglas	*Elementos no terminales*	*Elementos terminales*
O → el ART	O	el
ART → niño N	ART	niño
ART → perro N	N	perro
N → ríe V	V	ríe
N → ladra V		ladra
V → .		

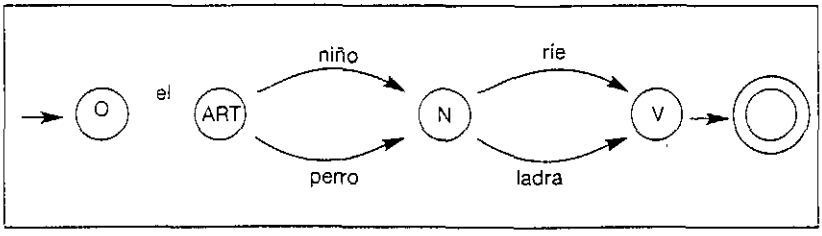


Figura 3.1. Diagrama para la gramática de estados finitos.

El diagrama debe leerse de la siguiente manera: el estado inicial O recibe un primer símbolo de entrada, *el*. Como hay un arco etiquetado precisa-

mente con ese símbolo, el autómata se mueve al segundo estado, ART. (Si en lugar de ser *el* la cadena de entrada es otro símbolo, digamos *la*, este autómata no tendría forma de seguir y se pararía: la cadena no sería reconocida como oración gramatical. Esto es válido para cualquier estado del cual no parta un arco que esté etiquetado con la cadena de entrada.) En el segundo estado puede recibir sólo dos cadenas de entrada, *niño* o *perro*. Cualquiera de las dos le lleva hasta el tercer estado, N. Igualmente, tiene dos arcos posibles, *ríe* o *ladra*, que acaban en el cuarto estado, V. Desde allí sólo queda el estado final y se acaba el reconocimiento.

Otra manera de representar autómatas finitos es mediante tablas de estados. Verticalmente se muestran los estados, cinco en nuestro ejemplo. Horizontalmente se representan los arcos. El orden de las columnas en la tabla no afecta a la operación, pero se suele escribir de manera que refleje el orden del autómata. En las casillas se coloca el número del estado al que se mueve la transición si el símbolo de entrada concuerda con el del arco. El cero indica que no hay una transición válida desde ese estado para ese símbolo. En el diagrama, es equivalente a la no existencia de arco etiquetado con el símbolo saliendo de un determinado estado. Cuando ocurre esto, como se ha dicho, el autómata rechaza la cadena. La siguiente tabla de estados representa la gramática de la figura 3.1:

	el	*niño*	*perro*	*ríe*	*llora*	.
1	2	0	0	0	0	0
2	0	3	3	0	0	0
3	0	0	0	4	4	0
4	0	0	0	0	0	5
5	0	0	0	0	0	0

Las redes de transición pueden funcionar como reconocedores o como generadores. En el primer caso, comprueba si la secuencia de palabras de entrada se corresponde con algún camino permitido en la red y acaba en un estado final. En el caso de la generación, la red va construyendo la oración siguiendo los arcos. El término "estados finitos" indica que hay un número finito de nodos. Con este autómata de cinco estados se generan o reconocen las siguientes oraciones:

(1) El niño ríe.
(2) El niño ladra.
(3) El perro ríe.
(4) El perro ladra.

Este ejemplo trivial nos muestra una limitación importante de las gramáticas regulares: un autómata de estados finitos no puede generar una lengua natural en su totalidad, que es infinita. Para conseguir lenguas regulares infinitas hay que permitir reglas del tipo

$$A \rightarrow a\,A$$

donde el origen y destino del arco es el mismo nodo (representado gráficamente en la figura 3.2). Por ejemplo, reglas del estilo de ADJ → pequeño ADJ, ADJ → simpático ADJ, etc. permitirían reconocer oraciones con infinitos adjetivos, uno detrás del otro. Con este tipo de reglas damos cuenta de la *iteración* o *repetición* de elementos.

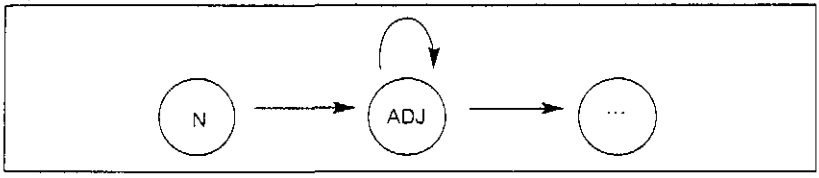


Figura 3.2. Ejemplo de iteración.

Pero la simple repetición de un nodo no es suficiente para tratar algunas construcciones de las lenguas naturales. Esta observación fue realizada por Chomsky a finales de los cincuenta. Demostró que las lenguas naturales tienen fenómenos recursivos que implican marcadores correlativos, como por ejemplo oraciones del tipo "si ... entonces ..." , "o bien ... o bien ..." que pueden estar formadas por un número indefinido de anidaciones:

> *Si* mañana llueve, *entonces o bien* vamos al cine *o bien, si* nos quedamos en casa, *entonces* ...

No es posible tratar este tipo de recursividad con una gramática regular dado que la única información que maneja es el estado en el que se encuentra. De esa manera no sabría cómo analizar las oraciones incrustadas, ya que no tiene manera de recordar las oraciones que ha generado ni en qué orden. En este punto merece la pena recordar los argumentos de Chomsky:

> Si estos procesos [las oraciones anidadas] no tienen un límite finito, podemos probar la inaplicabilidad literal de esta teoría elemental. Si los procesos tienen un límite, entonces la construcción de una gramática de

> estados finitos no será literalmente impensable, ya que será posible enumerar las oraciones, y una lista es esencialmente una gramática de estados finitos trivial. Pero esta gramática será tan compleja, que resultará de poca utilidad o interés. [...] Si una gramática no tiene artificios recursivos [...] será prohibitivamente compleja. Si cuenta con artificios recursivos de alguna especie, producirá un número infinito de oraciones (Chomsky, 1957: 39).

De este fragmento podemos extraer varias conclusiones pertinentes: desde un punto de vista teórico las gramáticas regulares son insuficientes ya que necesitan de mecanismos recursivos especiales (las gramáticas independientes del contexto son el tipo gramatical más sencillo capaz de dar cuenta de la recursividad en las lenguas naturales). Pero desde el punto de vista práctico, Chomsky reconoce implícitamente que si se descubriera algún límite a estos fenómenos entonces se podrían tratar con una red de estados finitos, aunque fuera muy compleja. Precisamente en este punto se apoyan quienes defienden el uso de autómatas finitos en PLN. ¿Existen acaso fragmentos de lenguas naturales que contengan fenómenos recursivos limitados, de manera que se puedan formalizar con un número finito de estados? Dado que las redes de transición son la aproximación más simple y fácil de implementar de cuantas hay en LC, es un buen argumento para utilizarlas. Muchos lingüistas computacionales creen que es posible y preferible por eficacia aplicar estas técnicas a la morfología y al reconocimiento léxico. Las reglas de flexión forman un conjunto (casi) cerrado y mucho más pequeño que las reglas de la sintaxis en cualquier lengua. La flexión morfológica no presenta en general fenómenos de anidamiento (aunque sí pueda darse en la derivación o la composición). ¿Por qué utilizar la potencia de otros tipos de gramáticas cuando se pueden utilizar los más sencillos? La utilización de autómatas de estados finitos para el procesamiento morfológico se ha convertido en la aplicación más popular (como se verá al hablar de ese nivel). También se ha utilizado en etiquetadores de categorías, dado que el número de palabras flexionadas de una lengua aunque grande e indeterminado no es infinito, como ocurre con las oraciones. Por ejemplo, para el español, con un diccionario de unos 40.000 lemas se pueden producir unas 500.000 formas flexionadas, y con ellas se da cuenta de la mayor parte de las palabras utilizadas realmente en la lengua. Existen lexicones, como el de Tzoukerman y Liberman (1990) que recogen muchas de esas formas utilizando un modelo de estados finitos. Como avisa Chomsky, la gramática se hace prohibitiva de leer y manejar para un humano, pero no así para una máquina. Los autómatas de estados finitos no explican psicológicamente el procesamiento lingüístico humano, pero para muchas tareas "finitas" proporcionan un método muy eficiente de computar. En los años noventa se han hecho muy populares. La compilación de Roche y Schabes (1997) recoge una buena muestra del estado actual de la cuestión.

3.2.4. Gramáticas independientes del contexto

Estas gramáticas contienen reglas con el siguiente formato:

$$\alpha \rightarrow \beta$$

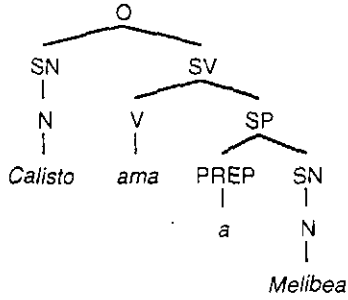
donde α es un elemento no terminal y β es una combinación de elementos terminales y/o no terminales, incluido el elemento vacío. Mediante estas reglas se pueden describir las estructuras sintácticas de múltiples fenómenos en las lenguas naturales. De hecho, en los primeros modelos transformacionales el componente gramatical básico estaba especificado con reglas independientes del contexto, dejando para las transformaciones la tarea de dar cuenta de fenómenos más complejos. Otras teorías generativas no transformacionales, como GPSG, defienden que todos los fenómenos se pueden describir con este tipo de gramáticas si se añaden algunas extensiones formales, como el uso de rasgos o la operación de unificación.

Las lenguas naturales tienen tanto estructura jerárquica como un ordenamiento lineal. Estas dos características se reflejan directamente con reglas independientes del contexto. Por una parte, toda regla expresa, mediante la separación en una parte derecha e izquierda con respecto a la flecha, un relación de dominio inmediato. El elemento de la izquierda es el nodo madre del que dependen los nodos hijos de la parte derecha. A su vez, los nodos hermanos (es decir, aquellos que aparecen en la parte derecha) presentan una ordenación lineal, donde cada elemento ocupa una posición. Estas dos propiedades de las gramáticas sintagmáticas se pueden ver claramente en una representación arbórea (figura 3.3).

Cualquier gramática sintagmática independiente del contexto está formada por un conjunto de reglas y un conjunto de entradas léxicas (o lexicón). El cuadro 3.3 muestra una gramática inicial para el español, que nos servirá para ilustrar una serie de convenciones.

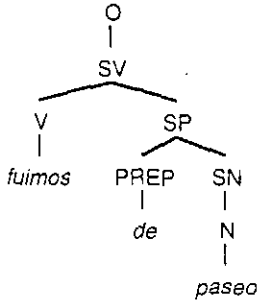
Las entradas léxicas aparecen agrupadas por categorías, entre llaves y separadas por el símbolo "|", que indica que son elementos en alternancia. Esta convención es equivalente a escribir, por ejemplo, DET → el, DET → la, etc. Tiene la ventaja de proporcionar una descripción más compacta. Esta gramática es capaz de reconocer y generar oraciones como las que se muestran en la figura 3.3. Estas gramáticas también proporcionan la estructura jerárquica interna de las oraciones. Para visualizarla gráficamente se utilizan dos tipos de representaciones: los llamados árboles de estructura sintagmática *(phrase structure trees)* y los corchetes etiquetados. Los primeros son un tipo de grafo (las redes de transición que se vieron en el apartado anterior eran otro tipo de grafo). En general, se suelen preferir los árboles a los corchetes etiquetados, pues las relaciones de dominio y precedencia se observan mejor.

(1)



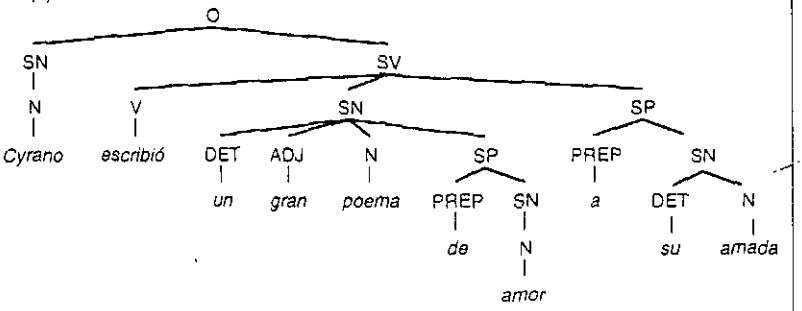
[O[SN[N[Calisto]]][SV[V[ama]][SP[PREP[a]][SN[N[Melibea]]]]]]

(2)



[O[SV[V[Fuimos]][SP[PREP[de]][SN[N[paseo]]]]]]

(3)



[O[SN[N[Cyrano]]][SV[V[escribió]][SN[DET[un]][ADJ[gran]][N[poema]][SP[PREP[de]][SN[N[amor]]]]][SP[PREP[a]][SN[DET[su]][N[amada]]]]]]

Figura 3.3. Algunas oraciones generadas por Gramática 1.

CUADRO 3.3. Gramática 1.

Reglas	*Lexicón*
O → *SN SV* O → *SV* *SV* → V *SV* → V SN *SV* → V SP *SV* → *V SN SP* SN → PRON SN → N SN → DET N SN → DET ADJ N SP SP → *PREP SN*	DET: {el | la | los | las |un |una |unos |unas | su} PRON: {yo | tú | él | ella | nosotros | vosotros | ellos} N: {Cyrano | Calisto | Melibea | corazón | nariz | amor | mentira | poema | paseo | amada} ADJ: {gran | roja} *PREP*: {a | de} V: {tiene | ama | habla | escribió |regaló | fuimos}

Las tres oraciones son ejemplos de estructuras básicas del español. A pesar de ello, nuestro análisis de (2) puede ser controvertido. Es habitual encontrar en descripciones teóricas del español que su estructura oracional es necesariamente bimembre (O → SN SV). Esto implica que para oraciones sin un sujeto expreso, hay que postular una categoría vacía (algo así como SN → *e*, donde e es un elemento sin significante). Esto supone una complicación extra en el programa, ya que se tiene que expresar en el análisis un elemento que no aparece en la cadena de entrada. Por tanto, se ha decidido escribir la regla O → SV, que permite reconocer oraciones gramaticales del español sin necesidad de postular un elemento sin realización fonética (o superficial). Desde el punto de vista teórico no es tan descabellado como pudiera parecer, dado que el sujeto está implícitamente representado en los morfemas de concordancia (persona y número) del verbo, no mediante un elemento vacío.

El ejemplo (2) ilustra varios puntos esenciales que se deben tener en cuenta a la hora de escribir gramáticas, ya sean teóricas o computacionales:

1. Toda gramática es una teoría acerca de una lengua: no hay descripciones neutrales. Una oración sencillísima y frecuente como "Fuimos de paseo" se puede analizar al menos de dos maneras diferentes. Cada una supone una posición distinta acerca de la pertinencia de las categorías vacías. Hay argumentos para defender cualquiera de las dos posturas, pero son incompatibles y tenemos que adoptar una decisión al principio, para que las nuevas reglas que se vayan añadiendo a la gramática sean coherentes con la postura adoptada. Mantener la coherencia de la gramática es una de las cuestiones más complejas

de conseguir, y trataremos de ello al hablar de consejos prácticos sobre cómo desarrollar gramáticas.

2. La gramática reconocerá como gramaticales aquellas combinaciones que estén recogidas en las reglas de su gramática, y les asignará la estructura especificada. Esto significa que se suelen producir "divorcios" entre las reglas equivalentes de una gramática teórica y las de una computacional. Hay tratamientos que son permitidos sin problema en una gramática teórica y en cambio se ponen muchas reservas en una gramática computacional, y viceversa. Por ejemplo, escribir un elemento vacío en un árbol de análisis no supone ninguna complicación para un lingüista teórico, pero sí para un computacional. En cambio, desde posturas teóricas se busca la regularidad y la simetría en los análisis, y por tanto dejar una ramificación con un único elemento hijo es algo "excepcional" en las representaciones estructurales. En este punto, los lingüistas teóricos han desarrollado análisis muy generalizadores y abstractos (por ejemplo la Teoría de la X'), con ramificación exclusivamente bimembre (es decir, de cada nodo madre dependen única y necesariamente dos nodos hijos). Sin embargo, los lingüistas computacionales prefieren análisis menos abstractos y lo más parecidos a la representación superficial de la cadena de entrada. Nuestra oración (3) es un ejemplo de esta estrategia: el SN "un gran poema de amor" presenta una estructura bastante plana, con cuatro ramas saliendo del nodo SN.

La gramática 1 es claramente un modelo insuficiente incluso para oraciones muy sencillas del español. Por ejemplo, no es capaz de reconocer varios adjetivos seguidos, lo que se consigue con el autómata finito de la sección anterior, gracias a la iteración. De hecho, la gramática 1 se podría expresar con un autómata de estados finitos.

Para dar cuenta de la repetición de una categoría se puede utilizar la convención de Kleene. Así X^* significa que el elemento X puede aparecer cero o más veces; X^+ significa que el elemento X puede aparecer una o más veces. De esta manera, podemos volver a escribir la regla del SN como

$$SN \rightarrow DET\ ADJ^*\ N\ SP^*$$

Para describir las construcciones recursivas que no pueden ser tratadas con gramáticas regulares tendremos que emplear una regla recursiva, es decir, una regla donde el elemento de la izquierda también aparezca en la derecha. Por ejemplo, supongamos que creamos un nuevo constituyente, SADJ, sintagma adjetivo:

$$SADJ \rightarrow ADJ\ SADJ$$

Esta regla también se expandiría infinitamente, ya que donde aparece SADJ en la parte derecha siempre se puede sustituir por ADJ SADJ. Naturalmente, para tratar los casos de un único adjetivo habrá que postular o bien que el SADJ puede estar vacío en algunos casos, o bien utilizar la opcionalidad, que se explicará en seguida, o añadir la siguiente regla para "terminar" la recursión.

$$SADJ \rightarrow ADJ$$

Ambas técnicas, la convención de Kleene o la recursividad en la parte derecha de la regla, permiten formalmente generar o reconocer infinitos constituyentes del mismo tipo. Obviamente, las oraciones de las lenguas naturales no son de longitud infinita por limitaciones de la actuación, no de la competencia (por ejemplo, nuestra memoria no permite recordar más allá de un número limitado de elementos incrustados). A la hora de escribir gramáticas computacionales a veces es recomendable no hacer uso del poder de la recursividad y utilizar otros métodos más controlados.

Además de la recursividad de constituyentes, también podemos introducir una serie de convenciones que mejorarán la capacidad expresiva de nuestra gramática al tiempo que su formulación es más compacta. Por ejemplo, podemos extender a las reglas la convención de la alternancia de elementos que hemos aplicado a las entradas léxicas. Análogamente, para indicar qué constituyentes están en alternancia utilizaremos las llaves, y los separaremos mediante la barra. Por ejemplo:

$$\{ ADJ \mid SP \mid OREL \}$$

OREL es la variable para "oración de relativo". Esta regla dice que esos tres constituyentes están en alternancia (o distribución complementaria), es decir, que cada vez que se aplique la regla hay que escoger entre uno de ellos, y en ningún caso se pueden dar más de uno a la vez.

Si lo que queremos es permitir los tres constituyentes, pero a su vez que no sean obligatorios, entonces introduciremos una nueva convención para la *opcionalidad:* los constituyentes opcionales irán entre paréntesis:

$$(ADJ)\ (SP)\ (OREL)$$

De esta manera, indicamos que cada uno puede aparecer o no, independientemente de los otros.

Podemos combinar la alternancia y la opcionalidad. Por ejemplo:

$$(\{ ADJ \mid SP \mid OREL \})$$

describe que hay un componente opcional (es decir, puede aparecer o no) y que puede ser uno de esos tres constituyentes.

Como se adelantó más arriba, la opcionalidad se puede utilizar para expresar la recursividad en la parte derecha de la regla, permitiendo cero o más apariciones del mismo elemento:

$$SN \rightarrow (DET)\ (SADJ)\ N\ (SADJ)$$
$$SADJ \rightarrow ADJ\ (SADJ)$$

Con estas dos reglas describimos el hecho de que un N puede estar modificado a la derecha y a la izquierda por cero o más adjetivos.

La gramática 2 mostrada en el cuadro 3.4 incluye ejemplos de recursividad, opcionalidad y alternancia. Con sólo una regla más, la gramática 2 genera muchas más oraciones que la gramática 1. (Téngase en cuenta que aunque se han añadido constituyentes nuevos como SADJ o OREL, varias reglas se han reunido en una sola.) La figura 3.4 muestra algunas oraciones que reconoce y genera esta gramática.

CUADRO 3.4. Gramática 2.

Reglas	*Diccionario*
O -> (SN) SV SV -> V SV -> V *SN* SV -> V SP SV -> V SN SP SV -> V SP SP SV -> V SADJ SN -> PRON SN -> (DET) (SADJ) N ((SADJ | SP | OREL)) SADJ -> (ADV) ADJ ((ADJ | SP)) SP -> PREP SN OREL -> PRONREL SV	DET: {el | la | los | las | un | una | unos | unas} PRON: {yo | tú | él | ella | nosotros | vosotros | *ellos*} N: {Cyrano | Calisto | Melibea | corazón | nariz | amor | mentira | poema | reflejos | hombre | parque | telescopio} ADJ: {gran | roja | rápido} PREP: {a | de | con | en} V: {tiene | ama | habla | regaló | es | está | vi} ADV: {muy} PRONREL: {que | quien}

La oración (6) es un ejemplo de sobregeneración (o sobreanálisis, en su caso): la gramática 2 es capaz de generar (y reconocer) muchas oraciones que son agramaticales en español. Naturalmente esto es indeseable desde cualquier punto de vista, teórico y práctico. Se podría argumentar que es altamente improbable que algún sistema se encontrara alguna vez con una oración semejante, pero el hecho es que nuestra gramática la reconocería como gramatical. Debemos, por tanto, imponer algún tipo de restricción a nuestra gramática sintagmática para que dé cuenta exclusivamente de las oraciones gramaticales del español.

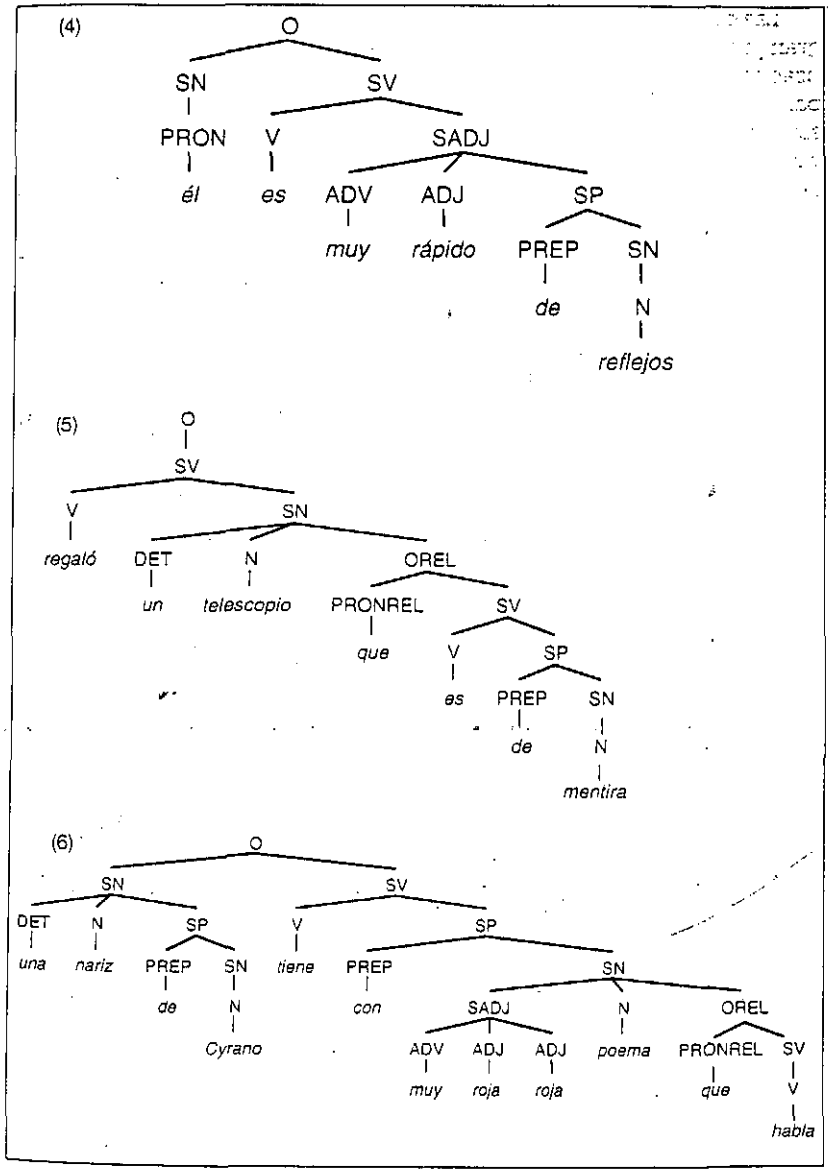


Figura 3.4. Algunas oraciones generadas por Gramática 2.

Las dos restricciones más comunes que aparecen en la mayoría de las gramáticas son las que tratan los fenómenos de concordancia y subcategorización. El primero consiste en que varios constituyentes de la oración comparten necesariamente una serie de rasgos morfosintácticos (número, persona, género, etc.). Entonces se dice que están "en concordancia". Las lenguas varían en cuanto a los rasgos que intervienen en la concordancia: se puede decir, en general, que depende de qué información gramatical se marca morfológicamente. Por ejemplo, las lenguas sin género morfológico, como el inglés, no exigen concordancia de género (aúnque en algunas situaciones se pueda marcar sintácticamente con los pronombres). En cambio, las lenguas con casos, como el alemán o el latín, exigen que los modificadores del nombre lleven el mismo caso que éste. La concordancia es, por tanto, un fenómeno que afecta a los niveles morfológico y sintáctico.

La subcategorización es una información que forma parte de las entradas léxicas y que especifica las propiedades de combinación de las palabras. En concreto, la subcategorización describe los requisitos sintácticos que impone un determinado elemento léxico (un verbo, un nombre, un adjetivo) sobre sus argumentos o complementos. Por ejemplo, un verbo como *clavar* exige tres argumentos: un sujeto, un objeto directo y, opcionalmente, un objeto instrumental *(alguien clava algo con algo)*. Un adjetivo como *rápido* exige un complemento con la preposición *de (rápido de reflejos)*, mientras que *apto* exige la preposición *para (apto para comer)*. La violación de tales requisitos implica la agramaticalidad, ya sea por omisión de un argumento obligatorio *(*Juan clava)* como por la aparición de un argumento no apropiado *(*apto de comer)*. La subcategorización es una información esencial para determinar la gramaticalidad y la asignación de estructura a una oración.

En la oración (6) se incumple tanto la concordancia (**roja poema*) como la subcategorización *(*tiene con ...)*. A pesar de que ambas son imprescindibles, no se expresan bien con gramáticas independientes del contexto. Esto se debe a que implican dependencia del contexto: hay que conocer las características del sujeto y del verbo principal para determinar si están en concordancia; la subcategorización se comprueba entre el núcleo y sus modificadores. En resumen, se necesita consultar la información de varios constituyentes.

Hay varias técnicas para resolver estos problemas, aunque ninguna es elegante ni completamente satisfactoria. Se presenta una para cada fenómeno y en el siguiente apartado (sobre las gramáticas de unificación y rasgos) se verá un tratamiento más apropiado para ambos casos.

La solución más conocida es multiplicar las categorías sintácticas, de manera que especifiquen información morfosintáctica relevante. En español, por ejemplo, convertiremos las categorías DET, N y ADJ en:

Masculino (M) singular (S)	DETMS	NMS	ADJMS
Femenino (F) singular (S)	DETFS	NFS	ADJFS
Masculino (M) plural (P)	DETMP	NMP	ADJMP
Femenino (F) plural (P)	DETFP	NFP	ADJFP

Paralelamente necesitaremos cambiar nuestra regla del SN. Supongamos, por motivos de simplicidad, que es:

$$SN \rightarrow (DET)\ (ADJ)\ N\ (ADJ)$$

Las nuevas reglas serían:

SNMS → (DETMS) (ADJMS) NMS (ADJMS)
SNFS → (DETFS) (ADJFS) NFS (ADJFS)
SNMP → (DETMP) (ADJMP) NMP (ADJMP)
SNFP → (DETFP) (ADJFP) NFP (ADJFP)

Además de la pérdida de legibilidad, se pierde la generalización que expresa la regla única para el SN. Téngase en cuenta que habrá que volver a modificar estas reglas para incluir la información de persona, necesaria para la concordancia con el verbo:

SNMP1 → PRONMP1 (pronombre mas. plural de 1.ª pers. "nosotros)
...
SNFP2 → PRONFP2 (pronombre fem. plural de 2.ª pers. "vosotras")
...
SNMS3 → (DETMS) (ADJMS) NMS3 (ADJMS) (SN mas.sing de 3.ª persona)
etc.

y modificar las reglas correspondientes del SV y de la oración:

O → SNMP1 SVP1 (concordancia: plural y 1.ª persona)
O → { SNMS3 | SNFS3} SVS3 (concordancia: singular y 3.ª persona)
etc.

Esta solución multiplica innecesariamente el número de reglas y de categorías. En una lengua como el inglés, donde la comprobación de la concordancia es muy limitada, esta técnica es factible. Por ejemplo, en los verbos plenos en presente la única distinción es entre "tercera persona del singular" y "no tercera persona del singular"; en otros tiempos ni siquiera se da. A medida que la complejidad morfológica aumenta, este procedimiento es completamente ineficiente.

Para la subcategorización podemos utilizar una técnica semejante: asignamos a cada combinación de complementos verbales un tipo de verbo diferente. Si tomamos las cinco reglas del SV en Gramática 2, podemos adaptarlas de la siguiente manera:

Gramática	*Diccionario*	
SV → V1	V1: {morir | correr }	% verbos intransitivos
SV → V2 SN	V2: {ver | tener}	% verbos con O. Dir.
SV → V3 SP	V3: {hablar}	% verbos con O. Prep.
SV → V4 SN SP	V4: {regalar}	% verbos con O. Dir e Indir.
SV → V5 SP SP	V5: {pegar}	% verbos con O. Indir. y Prep
SV → V6 SADJ	V6: {ser | estar}	% verbos copulativos

¿Cómo se trata la subcategorización de esta manera? Cada verbo está clasificado en el diccionario según un código (V1, V2, etc.) y, para que se aplique la correspondiente regla, es necesario que se den las otras condiciones estructurales. Así un verbo *tener*, que es tipo V2, no podría combinarse con un SP como ocurre en la oración (6). De esta manera, se controla la asignación de complementos para cada verbo. Esta estrategia, a diferencia de la que vimos en la concordancia, tiene ventajas:

1. No supone ningún aumento innecesario de las reglas gramaticales.
2. Estructura el diccionario en clases, capturando generalizaciones estructurales significativas.

Sin embargo, hay serios inconvenientes sobre este tratamiento puramente sintáctico de la subcategorización:

a) La subcategorización es un fenómeno básicamente lexico-semántico: la estructura oracional se predice en gran medida de la semántica del núcleo verbal. El tratamiento que hemos presentado sólo comprueba si el número de complementos o su categoría son correctos, pero

no controla las restricciones seleccionales que determinan la gramaticalidad semántica de una oración (por ejemplo, *Calisto ama a un telescopio* sería aceptada por la gramática).

b) No supone ningún ahorro de reglas gramaticales. En cambio, una aproximación lexicista (primacía del componente léxico sobre el gramatical) puede reducir substancialmente el número de reglas. Se verá en el siguiente apartado, gramáticas de unificación y rasgos.

Pero si para la concordancia y la subcategorización hay soluciones parciales, para otros fenómenos característicos de las lenguas naturales no hay manera de tratarlos con una gramática sintagmática. Nos referimos a los *constituyentes discontinuos:* aquellos constituyentes que se muestran en más de una posición estructural, es decir, que están separados por otros constituyentes. Por ejemplo, *¿qué libro regaló Luis a su madre?* Cualquier interrogación sobre algún constituyente supone un "movimiento" o reordenamiento de los constituyentes de la correspondiente oración afirmativa: *Luis regaló un libro a su madre.* En ambos casos, el O. Directo (el libro) forma parte de SV; en la interrogativa, el Sujeto (Luis) se intercala entre dos elementos constituyentes del SV, el V y el SP Objeto Indirecto. Por supuesto, hay una manera prolija de registrar este ordenamiento de constituyentes: escribir una regla para cada combinación posible. Por ejemplo:

$$O \rightarrow SQ\ V\ SN\ SP$$

(SQ sería un nuevo constituyente intermedio, Sintagma Q, para los sintagmas donde aparezca algún elemento interrogativo o relativo. Estos sintagmas están encabezados en español por pronombres que empiezan por Q: qué, quién, [c]uál, etc.). En las lenguas sin un orden estricto de constituyentes esta estrategia se hace impracticable. El español es una lengua moderadamente libre en cuanto al orden de sus elementos; pueden escribirse reglas para tratar muchas combinaciones, aunque el tamaño de la gramática afecta negativamente a la rapidez de procesamiento y dificulta el control de las restricciones necesarias.

La principal limitación de esta estrategia es que se pierde la generalización de la dependencia estructural entre los constituyentes oracionales, que es la misma en una oración afirmativa, en una interrogativa, o en una relativa. Este fue uno de los argumentos de Chomsky para defender la pertinencia de las transformaciones: son capaces de describir elegantemente los "movimientos" de los constituyentes discontinuos superficiales.

En resumen, las gramáticas independientes del contexto no son lo suficientemente expresivas para formalizar fenómenos importantes de las lenguas naturales. En la práctica, ningún sistema PLN de cierta cobertura utili-

za la versión pura de este tipo de gramáticas. Sin embargo, si se añaden ciertas extensiones formales, como el uso de rasgos, podemos dar cuenta del contexto sin aumentar significativamente el poder computacional de la gramática. Por eso, se reconoce casi unánimemente que las gramáticas de unificación y rasgos son el modelo computacional más completo y restringido al mismo tiempo conocido hasta la fecha. Se verá en el siguiente apartado.

3.2.5. Gramáticas de unificación y rasgos

Dentro de este término se agrupa una clase bastante amplia de formalismos y teorías gramaticales que se caracterizan por hacer complejas descripciones formales mediante el uso de rasgos y por utilizar una operación general para la combinación y comprobación de la información gramatical, conocida por *unificación*.

El origen de estos formalismos lo encontramos a finales de los años setenta y a principios de los ochenta, momento en el que confluyen distintas líneas de investigación. Por una parte, estaban los lingüistas computacionales como Martin Kay, que buscaban un método de codificación de información lingüística diferente a las redes de transición aumentadas de la época. Para ello se inspiró en algunas ideas y conceptos que se utilizaban ya en otras áreas de la computación, como el uso de rasgos y de la unificación (concepto aplicado por primera vez por Robinson en los años sesenta). Por otra parte, algunos lingüistas descontentos con el modelo transformacional, como Bresnan o Gazdar, empezaron a buscar alternativas menos poderosas formalmente que las transformaciones. La primera formulación de la Gramática de Unificación Funcional de M. Kay es de 1979. Posteriormente, en 1982, se publicó el libro fundacional de la Gramática Léxico-Funcional (LFG) y en 1985 la Gramática Sintagmática Generalizada (GPSG). Mientras tanto, se fueron creando entornos de programación para desarrollo de gramáticas de unificación, como PATR. La *Introducción a los formalismos gramaticales de unificación* de Shieber (1986) es el libro de referencia para esta primera época, escrito precisamente por uno de los creadores de PATR. El concepto de *unificación* se fue extendiendo a otros tipos de gramáticas formales, como las gramáticas categoriales o las gramáticas de adjunción de árboles. De todas las variantes, los dos modelos más populares en la actualidad son LFG y HPSG (Gramática Sintagmática Nuclear –Pollard y Sag, 1987 y 1994–). El término *unificación* parece algo gastado en los últimos años y algunos prefieren utilizar formalismos gramaticales "basados en restricciones" (Uzkoreit y Zaenen 1996): las posibles combinaciones de información están restringidas por medio de diferentes principios.

Se presentará en primer lugar los dos conceptos esenciales de estos formalismos:

1. *Estructuras de rasgos:* son el mecanismo básico de representación de la información. La idea es que las unidades lingüísticas son elementos de información. En ese sentido, podemos representar cualquier unidad lingüística, desde el fonema al discurso, mediante una estructura de rasgos. Una estructura de rasgos generalmente está formada por más de un rasgo. Un rasgo es un par compuesto por un atributo y un valor. El atributo lleva el nombre que identifica al rasgo. Ambas partes se distinguen por algún signo de puntuación, por ejemplo, "=" o ":" Por tanto,

número = plural

es un ejemplo de rasgo, donde indicamos que dicha estructura tiene "plural" como valor asignado al atributo "número".

Hay dos tipos de rasgos, en función de cómo sea el valor:

— *Valores atómicos,* es decir, símbolos que no se pueden descomponer porque no tienen más estructura, como "plural", "masculino" o "murciélago".
— *Valores complejos,* aquellos que a su vez son un rasgo o una estructura de rasgos.

La figura 3.5 muestra varios ejemplos de estructuras de ragos.

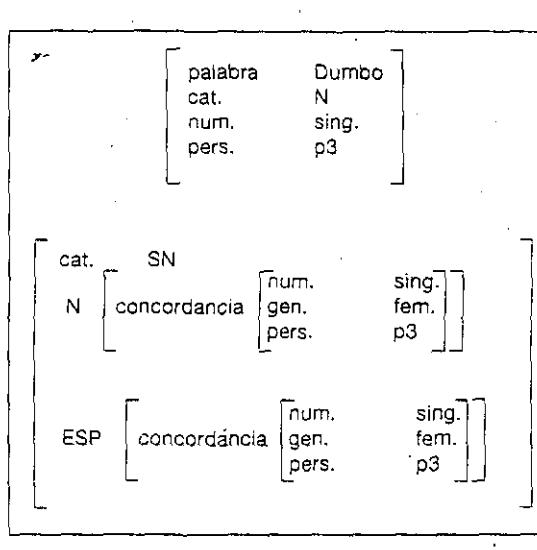


Figura 3.5. Algunas estructuras de rasgos.

2. La operación de *unificación:* la información contenida en distintas estructuras de rasgos se combina en una estructura nueva mediante la unificación. Para que la operación pueda producirse es necesario que las estructuras de rasgos tengan información compatible, pues en caso contrario no se unificarán (es decir, no formarán una estructura superior). La compatibilidad entre estructuras viene dada por la naturaleza de los rasgos que contengan, así como por la asignación de los valores a los rasgos. La idea clave es que dos estructuras de rasgos no pueden tener valores distintos cada una para el mismo rasgo. Por ejemplo, para que A y B unifiquen la estructura A no puede tener el rasgo "número = plural" y la estructura B "número = singular". Esto es una contradicción. Si interpretamos los rasgos como funciones parciales, entonces cada rasgo en un contexto dado puede tener un único valor o estar inespecificado, pero no tener más de un valor. Precisamente una de las características de las estructuras de rasgos es que, salvo en la estructura unificada final, siempre hay algún valor no definido, de ahí que se diga que las estructuras de rasgos son funciones parciales. Esto permite que diferentes estructuras informativas puedan ser combinadas coherentemente. Supongamos que todos los rasgos de cualquier estructura tuvieran asignado un valor concreto; esto es verdaderamente artificial ya que, por ejemplo, ¿qué valor asignaríamos al rasgo "número" de la palabra *crisis?*, ¿singular?, ¿plural? Lo más apropiado es dejarlo inespecificado, de manera que sea la información del nivel sintagmático la que asigne el valor. Concretamente, cuando se combinen las estructuras del artículo y del nombre, por ejemplo, *la crisis*, la información proporcionada por el artículo contendrá "número = singular", que podrá unificar con la del nombre "número = ?" (utilizamos el símbolo de interrogación para indicar que el valor es desconocido o inespecificado). La nueva estructura resultante, el SN, tendrá "número = singular".

   Aquellos rasgos que sólo aparezcan en una de las estructuras que se unifican, la nueva estructura los incorporará tal cual. Por tanto, la estructura unificada contendrá más información (será más específica) que las estructuras hijas, ya que combinará la información común entre ellas y también la información diferente.

Se ejemplificará con el tratamiento de la concordancia y la subcategorización. Como vimos en el anterior apartado, una gramática independiente del contexto maneja mal estos dos fenómenos. Para expresar las reglas de unificación utilizaremos el formalismo PATR-II, porque su uso está muy extendido en LC, es relativamente sencillo y además existen programas gratuitos que permiten practicar con pequeñas gramáticas (consúltense las referen-

cias en el capítulo 8). PATR-II es un lenguaje de programación que permite codificar información lingüística. Fue desarrollado originariamente por S. Shieber en SRI Stanford a principios de los años ochenta.

La concordancia es un caso muy indicado para ser tratado con rasgos. Así, por ejemplo, para la concordancia interna del SN en español necesitamos los rasgos de género y número. El valor para estos rasgos debe ser el mismo para todas las categorías con flexión nominal (nombre, adjetivo, determinantes, pronombres, demostrativos). La siguiente regla construye un SN. La primera línea define la estructura sintagmática y las otras líneas expresan restricciones sobre los valores de los rasgos.

SN → (DET) (ADJ) N { (ADJ) | (SP) }:
<SN conc> = <N conc>
<DET conc> = <N conc>
<ADJ conc> = <N conc>

Las restricciones se expresan mediante *paths* o *caminos* de rasgos. Es una convención notacional para referirse a fragmentos de las estructuras de rasgos. Un *path* es una secuencia de uno o más atributos (es decir, el nombre de los rasgos) encerrados entre ángulos (<....>). Por ejemplo, < DET conc> es un *path* que identifica el rasgo concordancia (conc) en la categoría DET. Hay, por tanto, una equivalencia entre ambas representaciones:

< DET conc> es equivalente a $\begin{bmatrix} \text{cat} = \text{DET} \\ \text{conc} = [\ \ ] \end{bmatrix}$

Los *paths* se utilizan tanto para establecer restricciones sobre las reglas como para crear macros de rasgos, de las que se hablará en el capítulo siguiente.

Regresando a la regla del SN, las dos últimas expresiones indican que el valor del rasgo concordancia del DET y el del ADJ debe ser igual al valor del rasgo concordancia del N. En español, el rasgo concordancia es un rasgo complejo compuesto por los rasgos número y género. Las siguientes entradas léxicas nos servirán de ejemplo:

Palabra *la*	Palabra *hermosa*	Palabra *molinera*
<cat> = DET	<cat> = ADJ	<cat> = N
<conc num> = sing.	<conc num> = sing.	<conc num> = sing.
<conc gen> = fem.	<conc gen> = fem.	<conc gen> = fem.
<lex> = el	<lex> = hermoso	<lex> = molinero

Si aplicamos la regla a estas tres entradas, la unificación de sus rasgos tendrá éxito y se construirá el SN. Si en lugar del determinante *la* hubiéramos tenido *el*, entonces la comprobación del *path* <conc gen> habría fallado y no se hubiera construido el SN. El SN que resulta de la aplicación de esta regla toma el valor de concordancia del N. Esto se expresa mediante <SN conc> = <N conc>. La copia de rasgos del núcleo a su proyección máxima es una característica de muchas teorías lingüísticas. La unificación probablemente es el mecanismo más sencillo y eficiente de implementar computacionalmente el Principio de Proyección.

Análogamente, podemos emplear la misma técnica para expresar las restricciones de concordancia entre el sujeto y el verbo principal.

O → SN SV:
<SN conc > = < SV conc >
<O suj> = <SN>
<O pred> = <SV>

La primera restricción establece que los valores de concordancia del SN tienen que ser iguales a los de concordancia del SV. Aquí hay que hacer una precisión: la concordancia externa comprueba los rasgos de número y persona. En la construcción del SN que acabamos de ver no se asigna el rasgo de persona. En cambio, en el verbo necesariamente tiene que aparecer. La unificación funciona porque cuando un rasgo no existe en algún constituyente no es incompatible con la información. Recordemos que la estructura unificada combina la información de todos los constituyentes, incorporando los rasgos que sólo estén especificados en alguno de ellos. Sin embargo, haría falta especificar "persona = tercera", en todos los SSNN, ya que su omisión permitiría que unificaran un verbo en primera o segunda persona con un SN sujeto en tercera: * *Mi hijo nací en ese hospital.* Como nos interesa controlar la sobregeneración, lo apropiado es buscar algún mecanismo que incorpore por defecto el rasgo "persona = tercera" automáticamente en cualquier SN que no tenga asignado dicho rasgo. Sólo en el caso de que el núcleo del SN sea un pronombre, el valor de "persona" estará especificado.

Las dos últimas restricciones de la regla O establecen el sujeto y el predicado de la oración, asignándoles respectivamente al SN y al SV.

La subcategorización es un fenómeno léxico que tiene una importancia esencial en la sintaxis porque especifica las posibilidades de combinación de las palabras. Este fenómeno se representa mediante marcos de subcategorización, que consisten en una lista de sintagmas. Cada sintagma ocupa alguna posición argumental respecto al elemento léxico que lo subcategoriza y se ordenan de forma que aparecen primero los que son obligatorios seguidos por los opcionales. En el caso del verbo, que es la categoría léxi-

ca que tiene más complementos subcategorizados, el argumento 0 (arg0) lo ocupa el sujeto, el arg1 el objeto directo, etc. En PATR se puede expresar la subcategorización de la siguiente manera en las entradas de diccionario:

Palabra *amar*:	palabra *dar*:
<cat> = V	<cat> = V
<arg0 cat> = SN	<arg0 cat> = SN
<arg0 función> = sujeto	<arg0 función> = sujeto
<arg1 cat> = SN	<arg1 cat> = SN ,
<arg1 función> = obj-dir	<arg1 función> = obj-dir
	<arg2 cat> = SP
	<arg2 valor-p> = a
	<arg2 función> = obj-indir

La regla que comprobará la subcategorización verbal y construirá el SV tendrá esta forma:

SV → V (X1) (X2):
<V arg1> = X1
<V arg2> = X2.

Esta regla contiene una notación especial: el símbolo X es una variable que se utiliza para referirse a cualquier elemento terminal o no terminal. Los números indican variables diferentes. Por ejemplo, en la regla se expresa que cualquier elemento que aparezca inmediatamente detrás del V deberá ser de la misma categoría y función que los valores del rasgo complejo arg1 del V. Análogamente, el segundo constituyente, X2, tendrá la misma categoría y función que las especificadas en el rasgo arg2 del V.

Las ventajas del uso de la variable X son evidentes, ya que permite capturar generalidades. En el caso de la subcategorización, por ejemplo, se necesita una única regla para todos los tipos, frente al tratamiento expuesto en la sección anterior. Sin embargo, las variables tienen que utilizarse con cuidado, ya que pueden ser sustituidas por cualquier elemento.

La regla de subcategorización expuesta es incompleta si la aplicamos a lenguas donde los constituyentes oracionales no siguen un orden estricto, como es el caso del español. Se tratará este problema en el siguiente capítulo al hablar del orden de constituyentes. Igualmente hablaremos de otros fenómenos muy relevantes en cualquier lengua, como la coordinación o las dependencias no acotadas, en la sección dedicada al procesamiento sintáctico. Ahora es el momento de recapitular lo expuesto sobre conocimiento lingüístico y dedicar algo de atención a la parte puramente informática.

3.3. La estructura de un sistema PLN simbólico

Cualquier sistema de PLN tiene dos grandes tipos de conocimiento almacenado:

1. *Conocimiento lingüístico*, en forma de gramática, lexicón y modelo conceptual del mundo. La gramática es simplemente una definición abstracta de un conjunto de elementos estructurados y bien formados. En términos psicolingüísticos sería el equivalente a nuestra competencia lingüística y pragmática.
2. *Programa* o *parser*, que contiene las instrucciones para procesar los datos lingüísticos. El parser es un algoritmo o conjunto de instrucciones que relaciona cadenas de símbolos con el conocimiento lingüístico almacenado. *Sería el equivalente al funcionamiento de nuestro* cerebro cuando produce y relaciona representaciones mentales.

Dentro del sistema, el conocimiento lingüístico son los datos sobre los que opera el programa para crear estructuras. Se parte de la base de que los objetos lingüísticos que se van a procesar son objetos estructurados, aunque su estructura no es manifiesta. El *parser* es el mecanismo computacional que infiere la estructura de las cadenas de palabras a partir del conocimiento almacenado en la gramática y diccionario, y establece si son cadenas gramaticales o agramaticales. Durante el procesamiento de los datos se crean muchas *estructuras de trabajo* que son temporales. Las estructuras finales son el resultado del análisis. El peso de ambas partes, el conocimiento lingüístico y el programa, depende del diseño general del sistema. En los primeros tiempos de la Inteligencia Artificial y el PLN predominó el *estilo procedural*, que da prioridad al programa sobre el conocimiento. Desde los años ochenta la tendencia es el *estilo declarativo*, que hace hincapié en la estructura del conocimiento y no en la forma de proceder con los datos.

3.3.1. Métodos de *parsing*

El problema que tiene que resolver cualquier parser es puramente sintáctico: reconocer las oraciones gramaticales y asignarles una estructura. Otros componentes del sistema se encargarán de su interpretación. Se han desarrollado muchas técnicas para tratar este problema con algún tipo de gramática generativa. La figura 3.6 muestra el esquema general del proceso combinando los dos componentes esenciales (conocimiento y programa) y su relación con las estructuras de trabajo y las estructuras analizadas resultantes.

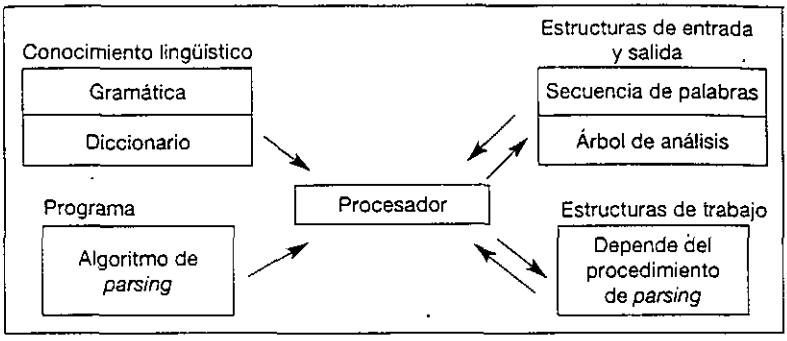


Figura 3.6. Esquema general del proceso de parsing (adaptado de Winograd 1983).

Las estructuras de entrada (una cadena de caracteres o palabras) son procesadas mediante la gramática y el diccionario siguiendo los pasos especificados en el algoritmo. Un algoritmo de *parsing* es un procedimiento que prueba diferentes maneras de combinar reglas gramaticales para encontrar una combinación que genere un árbol que pueda representar la estructura de la oración de entrada (Allen, 1987). Durante el procesamiento se construyen distintas estructuras intermedias o de trabajo, para finalmente producir un árbol de análisis estructural de la secuencia de entrada. Hay una correspondencia directa entre los árboles de análisis, las reglas de la gramática y las entradas del diccionario: cada nodo del árbol se corresponde con una regla, y cada elemento terminal del árbol con una entrada de diccionario.

Los algoritmos de parsing más utilizados están pensados para gramáticas independientes del contexto. Kartunnen y Zwicky (1985) lo atribuyen en parte a que se conocen bastante bien sus propiedades formales y en parte a que los lingüistas están interesados en utilizar gramáticas más restringidas. Ya se ha visto que las gramáticas *independientes del contexto aumentadas con rasgos son las más apreciadas*. La clave del diseño de los algoritmos de *parsing* consiste en aprovechar la correspondencia entre reglas y estructuras: se trata de confrontar los elementos que aparecen en la parte derecha de las reglas con los elementos sucesivos de las oraciones de entrada. Los algoritmos son los responsables de decidir qué reglas probar y en qué orden. Cada algoritmo suele combinar diferentes parámetros y diferentes estructuras de trabajo. La presentación que se expone se limitará a mostrar las combinaciones más usuales. Hay tres parámetros esenciales:

— *Análisis descendente/análisis ascendente:* la primera elección en el diseño de un algoritmo es si va a estar dirigido por la gramática o por los datos de entrada. El procesamiento descendente *(top-down)* bus-

ca primero en la gramática las reglas y va construyendo estructuras hasta que llega a completar las palabras de secuencia de entrada. El procesamiento ascendente *(bottom-up)* comienza por las palabras de entrada y busca reglas cuya parte derecha corresponda a combinaciones de palabras adyacentes. De esta manera, va construyendo estructuras hacia arriba hasta llegar al símbolo inicial, es decir, O.

— *Procesamiento secuencial/procesamiento en paralelo:* este parámetro tiene que ver con la manera de tratar las opciones de análisis que se pueden presentar en un determinado punto. El procesamiento secuencial prueba primero una opción hasta el final, y si falla regresa al punto inicial y prueba la siguiente, hasta que encuentra alguna con éxito o agota los caminos. El procesamiento en paralelo prueba diferentes posibilidades al mismo tiempo.

— *Procedimiento determinista/procedimiento no determinista:* un análisis puede estar dirigido de manera que sólo haya un camino para llegar a una configuración estructural particular (algoritmo determinista) o puede contener etapas donde haya que realizar alguna elección, es decir, a un análisis concreto se puede llegar por diferentes caminos (algoritmo no determinista). Los algoritmos deterministas son más eficientes, pero más limitados. Por otra parte, hay que señalar que el carácter determinista o no depende tanto de la gramática como del parser: si escribimos reglas sin opciones entonces obtendremos necesariamente un procesamiento determinista. El procesamiento psicolingüístico sugiere que las gramáticas de las lenguas naturales no pueden ser deterministas, dado que no podemos saber de antemano los análisis de una oración.

3.3.2. Algoritmos descendentes en serie con *backtracking*

Son el procedimiento más común (Grishman, 1986). Parten de O y van generando derivaciones sustituyendo el elemento no terminal más a la izquierda hasta llegar a la primera palabra que queda por reconocer. Veamos un ejemplo sencillo. El cuadro 3.5 muestra una gramática y un diccionario que se utilizarán para reconocer si la oración *Abelardo ama a Eloísa* es gramatical y cuál es su estructura.

Un parser descendente utiliza dos estructuras de trabajo, una es la *posición actual* de la palabra que se está reconociendo y la otra es el *resto* que queda por procesar. El resto sirve para llevar un registro de los elementos que faltan por reconocer. También se le llama *stack* o pila, donde los elementos se añaden y se eliminan en cada turno. Una vez completada la derivación de la palabra en la posición actual se prosigue con la primera palabra de la izquierda en el resto (es decir, en la posición más alta de la pila). En cada derivación se prueba una regla. La derivación de la oración se mues-

tra en el cuadro 3.6 (la derivación de una oración de acuerdo con una gramática dada es el recorrido que va desde la configuración inicial hasta la final en la cual todos los constituyentes terminales ya han sido analizados).

CUADRO 3.5. Gramática y diccionario de ejemplo.

Gramática	*Diccionario*
(1) O → SN SV	*a)* ART → el
(2) SN → ART N	*b)* N → gato
(3) SN → PRON	*c)* PRON → tú
(4) SN → NOM-PROPIO	*d)* NOM-PROPIO → Abelardo
(5) SV → V	*e)* NOM-PROPIO → Eloísa
(6) SV → V SP	*f)* V → ama
(7) SP → PREP SN	*g)* PREP → a

CUADRO 3.6. Derivación de Abelardo ama a Eloísa.

Posición en la secuencia	*Derivación*	
1. Abelardo	O	
1. Abelardo	O ⇒ SN SV	
1. Abelardo	O ⇒ SN SV ⇒ ART N SV	
1. Abelardo	O ⇒ SN SV ⇒ ART N SV ⇒ el N SV	Fallo: el ≠ Abelardo
1. Abelardo	O ⇒ SN SV ⇒ PRON SV	
1. Abelardo	O ⇒ SN SV ⇒ PRON SV ⇒ tú SV	Fallo: tú ≠ Abelardo
1. Abelardo	O ⇒ SN SV ⇒ NOM-PROPIO SV	
1. Abelardo	O ⇒ SN SV ⇒ Abelardo SV	Éxito
2. ama	O ⇒ SN SV ⇒ Abelardo V	
2. ama	O ⇒ SN SV ⇒ Abelardo ama	Fallo: la derivación ha acabado antes de llegar a la última posición
2. ama	O ⇒ SN SV ⇒ Abelardo V SP	
2. ama	O ⇒ SN SV ⇒ Abelardo ama SP	Éxito
3. a	O ⇒ SN SV ⇒ Abelardo ama PREP SN	
3. a	O ⇒ SN SV ⇒ Abelardo ama a SN	Éxito
4. Eloísa	O ⇒ SN SV ⇒ Abelardo ama a ART N	
4. Eloísa	O ⇒ SN SV ⇒ Abelardo ama a el N	Fallo: el ≠ Eloísa
4. Eloísa	O ⇒ SN SV ⇒ Abelardo ama a PRON	
4. Eloísa	O ⇒ SN SV ⇒ Abelardo ama a tú	Fallo: tú ≠ Eloísa
4. Eloísa	O ⇒ SN SV ⇒ Abelardo ama a NOM-PROPIO	
4. Eloísa	O ⇒ SN SV ⇒ Abelardo ama a Abelardo	Fallo: Abelardo ≠ Eloísa
4. Eloísa	O ⇒ SN SV ⇒ Abelardo ama a Eloísa	Éxito. FINAL

El procedimiento de *backtracking* (retroceder) se muestra en varias ocasiones. Por ejemplo, cuando prueba con las dos primeras reglas del SN y falla porque el elemento terminal no coincide con el de la posición, entonces vuelve a la última posición con éxito y prueba otra regla. El *backtracking* se utiliza para guardar árboles parciales, que se van sustituyendo cada vez que tiene éxito una derivación.

3.3.3. Algoritmos ascendentes en paralelo

Es el otro algoritmo más utilizado. Comienza buscando todas las categorías sintácticas posibles para cada palabra de la secuencia de entrada y después procede a combinarlas en todas las maneras posibles que sean consistentes con la gramática. Durante el proceso va construyendo estructuras parciales.

El procedimiento, en su especificación más básica, realiza innumerables computaciones que luego se descartan, bien porque las combinaciones producidas no corresponden a elementos adyacentes en la secuencia inicial, bien porque no hay ninguna regla en la gramática para la combinación que se está probando. Además, cada vez que se añade un nuevo constituyente a la estructura parcial éste se vuelve a analizar en cada ciclo posterior. Se han propuesto muchas formas de modificar este algoritmo de manera que evite reanálisis, tanto los que han tenido éxito como los que no. Lo veremos un poco más adelante al hablar de los analizadores con chart. Pero antes se hará un resumen comparativo de las dos aproximaciones. El cuadro 3.7 muestra las ventajas y desventajas de cada método.

CUADRO 3.7. Comparación de estrategias de *parsing*.

	Algoritmos descendentes (top-down)	*Algoritmos ascendentes (bottom-up)*
Ventajas	El *backtracking* consume mucho menos espacio. Toma en cuenta el contexto *izquierdo, lo que reduce las* pruebas de reglas.	Sólo considera las palabras que aparecen en la cadena de entrada. Sólo construye un análisis parcial de la *misma estructura.*
Desventajas	La mayor parte de la búsqueda *se hace en el nivel* de los elementos léxicos. Considera palabras y categorías que no aparecen en la secuencia de entrada. Repite análisis con el mismo *símbolo si éste* aparece en contextos distintos.	El procesamiento en paralelo consume *mucho espacio al* almacenar todos los análisis parciales. No tiene restricciones contextuales y prueba muchas combinaciones para las que no hay reglas.

Se puede comprobar, efectivamente, que los métodos descendentes sacan partido de su "conocimiento" de las reglas gramaticales, mientras que los ascendentes lo hacen de su "conocimiento" de los elementos léxicos pertinentes para la oración que está siendo analizada. A partir de los años ochenta se han impuesto procedimientos híbridos que combinan las ventajas de ambos. Es decir, aprovechan por una parte la capacidad del algoritmo descendente para construir selectivamente análisis parciales basados en el contexto izquierdo, y por otra parte la capacidad del algoritmo ascendente de crear cada análisis una sola vez. Por tanto, la clave del *parsing* eficiente está en cómo almacenar los resultados intermedios para evitar la redundancia en el espacio de búsqueda.

3.3.4. Algoritmos con chart

Los parsers con chart (tablas de almacenamiento) consiguen ese objetivo. Un *chart* es una estructura de datos que almacena resultados parciales de manera que el trabajo no tenga que ser duplicado. El chart lleva el registro de todos los constituyentes derivados en un punto del análisis, así como aquellas reglas que han sido aplicadas con éxito parcial pero que todavía no están completas. Estas estructuras se suelen representar mediante arcos. Los arcos activos son aquellos a los que les falta algún constituyente por reconocer. La operación básica de un parser basado en chart es combinar un arco activo (es decir, un constituyente incompleto, por ejemplo un SN sin modificador postnominal) con un constituyente completado (por ejemplo, un SP complemento del nombre). El resultado será o bien un nuevo constituyente (en nuestro ejemplo, un SN completo) o un nuevo arco activo que es una extensión del anterior (un SN al que le sigue faltando algún postmodificador). Todos los constituyentes que están completos se guardan en una lista hasta que son requeridos por un chart. Cuando el último arco activo se completa, termina el reconocimiento.

En cuanto a eficiencia, los parser basados en charts se consideran más eficaces que los que se basan sólo en la búsqueda ascendente o descendente, debido a que un mismo constituyente nunca se construye más de una vez (Allen, 1987). En cualquier caso, como advierte Winograd (1983:114), "en toda la discusión acerca de la eficiencia, es importante recordar que la eficiencia formal y la eficiencia práctica no son la misma cosa". De nuevo tenemos la misma situación que con la adecuación formal de las gramáticas. Aunque teóricamente se sepa que determinado algoritmo es más eficiente que otro, los resultados prácticos dependerán de la forma en que se ha implementado el procedimiento, mediante qué estructuras de datos, con qué lenguaje de programación y en qué máquina. Además, buena parte de la efi-

cacia del sistema depende de la gramática: el número de reglas y el grado de "selección" y "detalle" en la discriminación de candidatos.

3.4. Ideas principales del capítulo

Los modelos simbólicos son el paradigma predominante en LC, tanto desde la perspectiva lingüística como computacional. El repertorio de conceptos, métodos y aproximaciones es riquísimo y ha sido aplicado y experimentado sobre múltiples y variados problemas y lenguas.

Los fundamentos teóricos descansan en la teoría de las gramáticas formales, los autómatas y los algoritmos de *parsing*. Sin embargo, no hay un acuerdo unánime sobre cuáles son los más adecuados para el tratamiento computacional de las lenguas naturales. Como en cualquier ciencia aplicada, hay cierta separación entre lo esperable teóricamente y los resultados prácticos. El funcionamiento real de los sistemas depende de muchas y complejas variables que contradicen (o al menos no siguen fielmente) las demostraciones teóricas. Corolario: hay diversas maneras teóricas de abordar un problema y la cuestión es encontrar la máxima eficacia con los medios que tenemos. Una estrategia que ha tenido éxito con una determinada aplicación y/o lengua no implica que vaya a conseguir resultados equivalentes al cambiar de aplicación o de lengua.

Los modelos simbólicos más empleados en LC son los autómatas de estados finitos (por su sencillez y eficiencia de procesamiento) y las gramáticas sintagmáticas (por su poder expresivo para dar cuenta de fenómenos lingüísticos). Entre las últimas destacan las gramáticas de unificación, modelo que utiliza rasgos como instrumento descriptivo general, lo que permite dar cuenta de manera computacionalmente eficiente de restricciones basadas en la comprobación de características en distintos constituyentes.

3.5. Ejercicios

Para la mayoría de los ejercicios que se proponen estaría indicado utilizar algún programa pedagógico o algún intérprete de Prolog que cuente con DCG. En el capítulo 8 proporcionamos algunas sugerencias, así como direcciones para conseguirlos.

1. Hágase una lista de fenómenos y cuestiones que no se pueden representar ni explicar con una gramática formal.
2. Escribir una pequeña red de transición para tratar oraciones tomadas de un texto real, por ejemplo, un cuento para niños.

3. Escribir una red de transición que sea capaz de generar o reconocer las mismas oraciones que la Gramática 1.
4. Hágase un listado más o menos exhaustivo de las oraciones agramaticales del español que puede generar/reconocer la Gramática 2.
5. La opcionalidad y la alternancia no son más que convenciones para hacer más compacta y manejable la gramática. Desarrollar reglas sintagmáticas sencillas que son necesarias para dar cuenta de la regla:

$$(SDET)\ (SADJ)\ N\ \{\ SADJ\ |\ SP\ |\ OREL\}$$

6. Las dos técnicas expuestas para expresar la posibilidad de constituyentes repetidos (la iteración y la recursividad) implican diferencias menores en la interpretación. Por ejemplo:

$$SN \rightarrow (DET)\ (SADJ)\ N\ (SADJ)$$

y

$$SN \rightarrow (DET)\ ADJ^*\ N\ ADJ^*$$

son equivalentes en cuanto a capacidad generativa (reconocen los mismos SN), sin embargo asignan árboles de estructura diferentes. Dibújense los árboles y defínase la interpretación de cada análisis. ¿Hay argumentos para preferir un tratamiento sobre el otro?
7. Extiéndase la *Gramática 2* para dar cuenta de más casos en el SN ("todos los hombres" o "estos cinco discos"). Ayuda: se puede construir un SDET parecido al SADJ. Además, descríbanse nuevas reglas para tratar adverbios, oraciones pasivas y existenciales ("hace mucho calor", "hay tres libros en la mesa").
8. Escríbase una pequeña gramática de unificación para tratar la concordancia en latín. Hágase lo mismo con alguna lengua de distinto tipo y familia lingüística y compárense las gramáticas.

4.

Modelos simbólicos II: el conocimiento lingüístico

En este capítulo trataremos de los problemas generales que aparecen en cada nivel lingüístico, ejemplificándolos con el español. Hay que tener en cuenta que muchas de las técnicas y estrategias que se conocen no se aplican universalmente a las lenguas del mundo. El parecido tipológico entre el español y el inglés, así como con otras lenguas indoeuropeas, permite utilizar sin grandes dificultades y cambios los mismos recursos computacionales. Comprobaremos, sin embargo, que en algunos casos la eficiencia del procesamiento no es equivalente.

4.1. Gramáticas computacionales

Cualquier sistema PLN necesita una gramática que especifique cómo se forman las oraciones a partir de sus partes constituyentes (sintaxis) y cómo se deriva la información asociada con cada oración (es decir, su interpretación) de la información de sus partes. Una cuestión esencial es que dicha gramática sea capaz de tratar oraciones no vistas o conocidas por el sistema previamente. Esto implica alguna generalización con respecto a los datos que han servido de base a la gramática. Dicha generalización permite hacer predicciones sobre la gramaticalidad de nuevas oraciones. Como sabemos, las gramáticas generativas, en sus diferentes tipos, son capaces de definir una sintaxis y una semántica, así como de realizar predicciones sobre gramaticalidad.

4.1.1. Precisión frente a cobertura

Como destaca Pereira (1996), la elección de una gramática para una aplicación particular implica tomar decisiones sobre dos requisitos que están en conflicto: precisión y cobertura. La *precisión* mide el grado de acierto de la gramática en cuanto a procesamiento sintáctico y semántico. Lógicamente, se espera que la gramática haga lo mejor posible su tarea, pero la experiencia y los datos muestran que el rendimiento está lejos de ser excelente. Por otra parte, la *cobertura* gramatical mide la proporción de oraciones que reciben tratamiento (al menos de manera aceptable) con respecto a un conjunto de oraciones de evaluación. Ambas propiedades son necesarias: cuanto más precisa es una gramática, mejor es la calidad de sus análisis; por otra parte, también interesa la variedad de estructuras tratadas por la gramática. El conflicto entre ambos requisitos se presenta cuando queremos aumentar el rendimiento de alguno de ellos.

Para mejorar la precisión hay que incorporar más restricciones a la gramática, con lo que se tiende a perder cobertura, ya que las nuevas restricciones suelen rechazar algunas oraciones que son más o menos aceptables para los hablantes. Pereira aduce que esto se debe a que las restricciones más poderosas son en realidad idealizaciones de la actuación real de los hablantes. Es decir, algo de sobra conocido: la actuación es mucho más permisiva que la competencia.

Por otra parte, si queremos mejorar la cobertura habrá que aumentar el número de reglas. Cuando una gramática alcanza un tamaño aceptable cada vez se hace más difícil de controlar y extender: las reglas nuevas entran en complejas interacciones con las anteriores, oraciones que antes no presentaban problemas producen varios análisis equivocados, aumenta la ambigüedad; en suma, decrece la precisión.

4.1.2. Tres tipos de gramáticas computacionales

Pereira (1996) clasifica las gramáticas computacionales en tres clases:

1. *Gramáticas lingüísticas:* son versiones computacionales de teorías lingüísticas. En cuanto a cobertura, su principal diferencia con las gramáticas teóricas es que, además de los aspectos interesantes desde el punto de vista teórico, las gramáticas computacionales tienen que tratar fenómenos de uso real (fechas, expresiones de medida, formas de nombrar, estilos de puntuación, etc.). Otro aspecto importante es que en la última década ha crecido el interés por teorías muy lexicalizadas, es decir, donde el componente léxico tiene más importancia

que las reglas gramaticales. Esto se explica porque hay evidencia de que buena parte de la ambigüedad estructural depende de elementos léxicos (verbo, nombre, adjetivo). Por tanto, en lugar de utilizar gramáticas muy orientadas a la sintaxis se prefiere tratar los fenómenos con mecanismos léxicos. Por último, la mayor parte de las teorías lingüísticas requiere procedimientos de computación demasiado complejos para ser eficientes. Por lo tanto, este tipo de gramáticas computacionales no son lo suficientemente rápidas para algunas aplicaciones que exigen interacción (por ejemplo, una interfaz hombre-máquina con un sistema de procesamiento de habla).

2. *Gramáticas orientadas a tareas:* para aplicaciones como recuperación y extracción de información, o reconocimiento de habla, una gramática de cobertura lingüística amplia no suele ofrecer una eficacia interesante. En estos casos, se suelen utilizar lo que se conoce por *gramáticas semánticas*: gramáticas que están diseñadas para buscar conceptos y relaciones mediante palabras clave y reglas que especifican posibles estructuras con conceptos pertinentes. Toda información que es considerada no relevante semánticamente no se analiza, con lo que la eficiencia aumenta considerablemente. Lo que se prima en este tipo de gramáticas es la manera de guiar al parser para que resuelva de la forma más eficiente su tarea. Normalmente este tipo de aplicaciones (como el reconocimiento de habla o el resumen del contenido de textos) tiene dominios temáticos muy limitados, donde se conoce con bastante seguridad la información relevante. La principal limitación de estas gramáticas es que no son fáciles de adaptar a nuevos dominios ni a tareas diferentes, y su cobertura y precisión es menor que las gramáticas computacionales de uso general.
3. *Gramáticas orientadas a los datos:* a diferencia de los dos tipos anteriores que imponen fuertes restricciones a las reglas, las gramáticas orientadas a los datos relajan bastante las restricciones para conseguir una cobertura más amplia. Para compensar la precisión se recurre a la probabilidad: se escoge la derivación que sea más probable en ese contexto. El nombre de *gramáticas orientadas a los datos* les viene de que para computar las estadísticas necesarias hace falta recurrir a un procedimiento de aprendizaje o entrenamiento sobre un corpus de datos lingüísticos. De las gramáticas probabilísticas se hablará en el siguiente capítulo.

Se han visto los dos criterios básicos para diseñar la elaboración de una gramática computacional. Se tratan ahora cuestiones específicas de cada nivel lingüístico.

4.2. Procesamiento morfológico

La necesidad de tratar la información morfológica se hizo evidente cuando se empezaron a desarrollar los primeros sistemas para lenguas con morfología rica. Hasta entonces, por influencia del inglés (que tiene un reducido inventario de formas flexionadas para cada lexema), la estrategia que se utilizaba era incluir todas las formas en el diccionario, es decir, lo que se ha hecho en todas las gramáticas presentadas como ejemplo en el capítulo anterior. En una gramática computacional se considera prioritario tratar la estructura sintáctica, por lo que el reconocimiento de las estructura morfológica se dejó para una etapa posterior. Para gramáticas que utilicen un léxico reducido, que incluya sólo las formas más habituales de los verbos (por ejemplo, las terceras personas), este tratamiento es más o menos satisfactorio. Pero si queremos aplicar nuestro sistema a texto real, estas "gramáticas de juguete" son insuficientes. La estrategia del listado completo *(full listing)* no sirve para el español: cualquier verbo tiene más de 50 formas flexionadas; la mayoría de los nombres y adjetivos cuenta con dos o cuatro formas, dependiendo de si tiene marca de género *(gato, gata, gatos, gatas)* o no *(luna, lunas)*. Para lenguas con casos y otros tipos de categorías gramaticales marcadas mediante morfemas esta aproximación sencillamente no se puede plantear ni para una gramática de juguete. En finés, por ejemplo, un verbo puede llegar a tener más de 12.000 formas.

Es bien sabido que la estructura de una palabra se puede dividir en componentes más pequeños, dotados de significado: los morfemas. Las combinaciones de morfemas muestran generalizaciones que se pueden expresar en forma de reglas. Si estas reglas se codifican en algún componente de la gramática, el diccionario se verá libre de formas redundantes y solo contendrá la información para cada lema. El procesamiento morfológico reconocerá la estructura interna de las palabras, proporcionando al componente sintáctico la información morfosintáctica (número, persona, género, tiempo, aspecto, etc..) y la información léxica (el lema, la subcategorización) asociada a cada palabra.

Los dos problemas fundamentales de cualquier procesador morfológico son:

1. *Reglas de formación de palabras* (o morfotáctica): los morfemas no se combinan libremente. Por ejemplo, una raíz verbal no se puede unir con un sufijo nominal. Es necesario establecer las combinaciones válidas de morfemas. Las reglas de formación de palabras se pueden dividir en dos tipos: las reglas de la flexión y las reglas de creación de léxico (que incluyen la derivación y la composición).
2. *Reglas de alomorfia*: los morfemas pueden presentar variantes en alternancia para distintos contextos. Por ejemplo, el lema verbal CONTAR

tiene dos realizaciones superficiales (o alomorfos): /cont-/ y /cuent-/ como en *contamos* y *cuentan*.

En cuanto al primer problema, la mayoría de los sistemas se conforman con reconocer la flexión, dado que la derivación y la composición plantean múltiples problemas, sobre todo a nivel semántico, no resueltos desde un punto de vista teórico. En contraposición, la flexión en cualquier lengua es un fenómeno claramente delimitado y finito. A diferencia de la sintaxis, donde no se sabe con certeza el número y tipo de estructuras que se pueden dar en una lengua, en la morfología flexiva se conoce el número de formas que componen un paradigma flexivo y se pueden clasificar todos los tipos de regularidades e irregularidades. Bien es cierto que hay vacilación en casos concretos sobre cómo flexionar una forma (¿cuál es el plural de *pub*, o la conjugación de *abolir?*), pero no es comparable a la incertidumbre sobre la gramaticalidad de muchas construcciones sintácticas. Por lo tanto, el primer problema es tratable de partida.

El problema de la alomorfía también es tratable, dado que el número de alomorfos para cada lema se conoce. Podemos, por tanto, resolver el tratamiento de la flexión con una descripción pormenorizada de la morfología de una lengua dada. Para los casos de incertidumbre, se puede o bien decidir por una opción o permitir las diferentes posibilidades. Por tratarse de un número muy reducido, los casos inciertos no son un problema importante. Este hecho puede explicar la afirmación de Karlsson y Kartunnen (1996) de que "en los últimos 10-15 años la morfología computacional ha avanzado hacia aplicaciones de la vida real mucho más que los otros subcampos del PLN".

La estrategia de diseño de un procesador morfológico debe combinar dos requisitos:

— Encontrar el procedimiento computacional que sea el más eficiente (es decir, que reconozca más rápidamente la información contenida en las palabras).
— Integrar el conocimiento morfológico dentro del lexicón de una manera que pueda ser fácilmente ampliado.

En cuanto a la eficiencia, parece que hay un gran consenso acerca de que el uso de autómatas de estados finitos es el modelo más eficiente para la morfología computacional (Karlsson y Kartunnen, 1996). Sin embargo, desde la perspectiva de su integración en el componente léxico parece más apropiado un modelo basado en unificación y rasgos, al menos para lenguas con morfología de tipo fusionante como el español, alemán o ruso. En este punto, expondremos argumentos desarrollados en trabajos nuestros previos (Moreno, 1991, Goñi *et al.*, 1997, Moreno en prensa).

4.2.1. El modelo de dos niveles de K. Koskenniemi

Koskenniemi revolucionó la morfología computacional con su tesis de 1983, donde propone un modelo "universal" para el tratamiento de cualquier tipo de fenómeno, ya sea flexivo, derivativo o composicional. Su sistema utiliza el mismo tipo de conocimiento para análisis y generación, se puede aplicar a cualquier lengua y es muy eficiente porque está basado en autómatas de estados finitos. Dicho de una manera simple, parece el método definitivo. Para demostrarlo, Koskenniemi lo aplicó al finés, lengua que no había podido ser tratada de manera satisfactoria hasta la fecha. Este modelo, conocido como *Two-Level Morphology*, se ha aplicado a numerosas lenguas, algunas de ellas muy distintas del finés desde un punto de vista tipológico, como el árabe, el alemán o el español.

El hecho de que haya sido aplicado a múltiples lenguas (cosa que no se ha conseguido con otros métodos) ha llevado a pensar de manera generalizada que es el método "universal" para la morfología computacional. Sin embargo, no es el más satisfactorio ni desde el punto de vista teórico (Sproat, 1992) ni desde punto de vista de lenguas concretas (Moreno en prensa). Pero antes de ver sus limitaciones, conozcamos las características generales de los autómatas de estados finitos aplicados a la morfología.

Supongamos que contamos con una descripción completa de las formas de cada categoría sintáctica y de cada tipo de alternancia alomórfica. Esto es una tarea ardua pero posible pues, como hemos señalado, el inventario es finito. Por ejemplo, un diccionario de unos 40.000 lemas en español se expande en unas 500.000 formas. Esto permite desarrollar una gramática regular, donde se codifiquen las transiciones para todas las formas. Es decir, se construye una red gigantesca de estados y para reconocer una palabra se busca la transición o transiciones que recorren exactamente todos los caracteres que forman la palabra. Al llegar al estado final se muestra la información morfosintáctica asociada a la palabra. La figura 4.1 muestra un ejemplo de una parte mínima de la red, para algunas formas del lema CONTAR: *cuento, cuentas, cuenta, contamos, contáis, cuentan*. (El símbolo Ø significa estado final.) Como se puede observar, el lexicón está estructurado en forma de árbol de letras. El reconocedor va moviéndose por las ramas del árbol en función de la letra que va recibiendo en cada momento. Si llega a un estado en el que no puede continuar porque el carácter de entrada no coincide con ninguna de la ramificaciones posibles en ese estado, entonces retrocede hasta un punto donde pueda tomar otra transición. Si no hay posibilidad en la red de encontrar otro camino, entonces el autómata se para e informa de que la cadena de caracteres no es gramatical. Tzoukermann y Liberman (1990) aplicaron esta técnica al español. Su funcionamiento es muy sencillo y eficiente, pero su elaboración es muy monótona y pesada, aunque se desarrollen procedimientos automáticos para la construcción de la red.

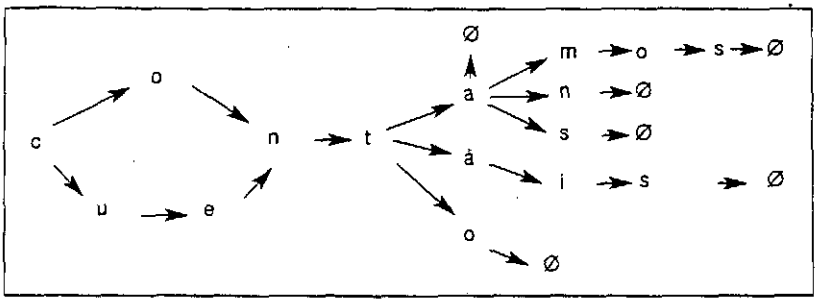


Figura 4.1. Fragmento de un árbol de letras.

Este modelo de autómatas secuenciales es el más sencillo, pero muy poco natural para que un lingüista computacional lo desarrolle y amplíe. Koskenniemi propuso un modelo más complejo y más atractivo para el "escritor del componente morfológico". Se basa en transductores de estados finitos, es decir, dos autómatas que trabajan en paralelo: un autómata va analizando la cadena de caracteres superficial y otro autómata la va contrastando con otra cadena donde se recoge una representación léxica de los morfemas. Por tanto, la tarea del transductor es reconocer si las dos cadenas, superficial y léxica, se corresponden. En caso afirmativo, el analizador devuelve la información asociada a cada morfema. La otra tarea posible es generar la forma superficial apropiada a partir de las representaciones léxicas. La figura 4.2 muestra el esquema general de los principales componentes del procesador morfológico en dos niveles más conocido, PC-KIMMO (Antworth, 1990: 3).

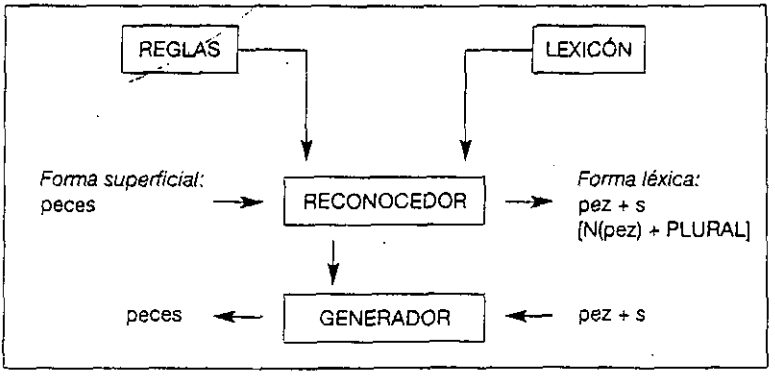


Figura 4.2. Principales componentes de PC-KIMMO.

Como es habitual, el conocimiento lingüístico se describe en un fichero de reglas y en un lexicón. Este último contiene la *representación léxica* para cada elemento léxico, además de especificar las restricciones morfotácticas mediante *clases de continuación* (es decir, qué morfemas se combinan para formar una palabra) y, por último, proporciona una *glosa* del análisis de cada elemento léxico. Dos ejemplos de entrada léxica serían:

Forma léxica	*Clase de continuación*	*Glosa*
pez	PLURAL	"N(pez)"
+s	*FRONTERA*	"+PLURAL"

La forma léxica tiene un estatus equivalente al de morfema en Lingüística Teórica, y la glosa es la información que porta cada morfema. "N(pez)" se interpreta como categoría Nombre, lexema *pez*. "+PLURAL" significa número plural.

Lo más peculiar de este modelo es que las reglas de combinación de morfemas no se especifican en el componente de reglas, sino en el lexicón. Efectivamente, las restricciones sobre el orden en que se deben concatenar *los morfemas se codifican mediante lo que Koskenniemi denomina clases de* continuación o sublexicones: el diccionario está dividido en clases léxicas compuestas por morfemas que se comportan de la misma manera en cuanto a restricciones de orden. En la entrada léxica de cada morfema se especifica el nombre del sublexicón que contiene morfemas que pueden adjuntarse a continuación. En nuestro ejemplo de entradas léxicas, el morfema *pez* puede ser seguido por algún morfema de la clase PLURAL. La clase FRONTERA *(BOUNDARY)* contiene un símbolo, normalmente "#", que indica que la cadena ha llegado a un morfema de cierre y, por tanto, la palabra ha sido *reconocida*.

Cada sublexicón es una autómata que recoge todos los morfemas de una determinada clase, por ejemplo, *PREFIJOS-NOMINALES, RAÍCES-VERBALES, PLURAL, DESINENCIAS-VERBALES*, y cualquier otra clase morfológica que pueda necesitarse según las características de la lengua tratada. Estos sublexicones a su vez se organizan en un autómata de estados finitos que recoge las combinaciones de morfemas permitidas. La figura 4.3 muestra una red de sublexicones para los verbos del español (las mayúsculas indican clases de continuación y las minúsculas sublexicones).

El estado inicial, etiquetado con "inicio", *representa el comienzo* de la palabra. Hay dos arcos que salen de él, uno etiquetado con PREFIJOS-V, y el *otro con 0. En ambos casos el estado siguiente es el sublexicón de* "Prefijos verbales", que contiene morfemas como in+, des+, re+, etc. Un arco etiquetado con 0 (cero) en un autómata finito significa que dicho arco pasa has-

ta el siguiente estado sin consumir ningún carácter de entrada. En otras palabras, es una forma de representar la posible ausencia de un prefijo verbal y, en general, es la manera de reflejar la opcionalidad de un constituyente en una red de transición. El segundo estado es el de "Raíces verbales". Dado que no hay ningún arco cero, se trata de un estado obligatorio. Lo mismo ocurre con el siguiente estado, "Desinencias verbales", hasta que llegamos al estado final.

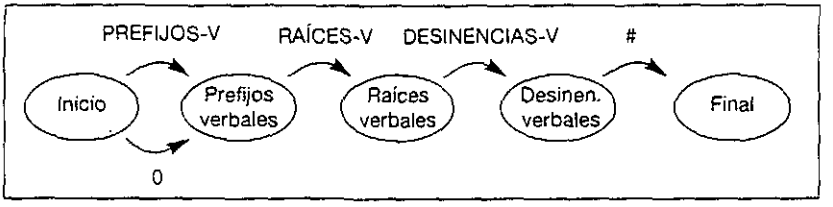


Figura 4.3. Red de sublexicones para verbos del español.

El componente de *reglas* en un modelo de dos niveles trata el segundo problema de todo procesador morfológico: la alternancia de formas superficiales del mismo morfema, los alomorfos. Por tanto, este componente en realidad sólo consiste en reglas de alomorfía. Koskenniemi se inspiró en el primer modelo de fonología generativa (Chomsky y Halle, 1968). En él las reglas de alomorfía se expresaban mediante reglas contextuales. Utilizaremos un ejemplo de cambio ortográfico, ya que su tratamiento computacional es equivalente, a efectos prácticos, a una regla fonológica:

$$z \rightarrow c \ / \ \_\_ + e$$

Esta regla expresa que el carácter *z* es sustituido por *c* en el contexto linde de morfema (+) seguido de *e*. Serviría para tratar la alternancia gráfica de *pez / pec-es*. La mencionada regla se puede expresar de la siguiente manera con una regla de dos niveles:

$$z : c \Rightarrow \_\_ + : 0 \ e$$

Sin embargo, ambas reglas no tienen exactamente la misma interpretación. En la regla contextual, el elemento de la izquierda es sustituido por (o transformado en) el de la derecha. Además es una regla secuencial que una vez aplicada no se puede deshacer, y se aplica en orden determinado: delante de unas reglas y detrás de otras reglas. La novedad de las reglas en dos

niveles es que son completamente declarativas: establecen correspondencias entre dos caracteres, no sustituyen uno por otro. De esa manera, la misma regla puede utilizarse para reconocimiento y generación. Además su aplicación es en paralelo, lo que supone que no hay niveles intermedios de representación entre la forma léxica y la superficial. De ahí toma su nombre. Los dos niveles se suelen representar de la siguiente manera:

RL (Representación léxica): p e z ~ 0 s
RS (Representación superficial): p e c 0 e s

Para conseguir la correspondencia entre ambas representaciones necesitamos dos reglas, una para el cambio ortográfico y otra para la epéntesis (inserción) de una e en la cadena superficial:

R1 (cambio z/c) z : c ⇒ __ + : 0 e
R2 (epéntesis) 0 : e ⇔ + : 0 __ s#

Estas dos reglas nos muestran la estructura general de cualquier regla de dos niveles:

$$L : S \Rightarrow C$$

— *Correspondencia:* es la parte izquierda de la regla; en ella se especifican un par de caracteres, por ejemplo, z : c, donde el primer carácter corresponde al nivel léxico y el segundo al superficial. Ambos pueden ser variables, por ejemplo, P : B podría utilizarse para expresar una correspondencia entre las consonantes oclusivas sordas en el nivel léxico y las consonantes oclusivas sonoras en el superficial.
— *Operador:* expresa el tipo de relación que se da entre el par de caracteres y el contexto. Hay cuatro tipos de operadores y su interpretación es la siguiente:

⇒ L se realiza sólo como S en el contexto C, pero no en otros contextos.
⇐ L se realiza siempre como S en el contexto C.
⇔ L se realiza siempre y sólo como S en el contexto C, y en ninguna otra parte.
/ ⇐ L nunca se realiza como S en el contexto C.

— *Contexto:* expresa la situación en que se debe verificar la correspondencia. Se puede tener en cuenta tanto el contexto anterior como el posterior, y ambos a la vez.

No se entrará en más detalles por limitaciones de espacio, pero es necesario avisar de que la construcción de una gramática en dos niveles supone una complicación mayor de la que pudiera parecer a simple vista. A medida que aumentan las reglas, comienzan las interacciones complejas y la necesidad de diferenciar las que se aplican con carácter general (lo que se conoce como *alomorfía condicionada fonológica u ortográficamente*, como el cambio z/c o los plurales -s/-es) de las que tienen un contexto más restringido *(alomorfía condicionada morfológicamente*, como las diptongaciones de las raíces verbales sólo en ciertas formas, por ejemplo, cont- / cuent-). En este último ejemplo hay que ser muy fino en la descripción, pues el contexto inmediato es el mismo: c __ nt. El peligro de las reglas de dos niveles es que son muy poderosas y pueden provocar correspondencias insospechadas.

Una vez elaboradas las distintas reglas fonológicas hay que traducirlas manualmente a tablas de estados. Ésta es, sin duda, la tarea más engorrosa, problema reconocido incluso por sus más ardientes defensores: "las tablas de estados no son la notación más feliz para las reglas fonológicas" (Antworth, 1990:11). Este autor dedica buena parte del libro a explicar cómo compilar las reglas en tablas. Sin embargo, en los últimos años ha mejorado mucho la situación: la nueva versión de PC-KIMMO (2) incluye un compilador automático de reglas. Además se pueden utilizar otros compiladores de acceso gratuito, como el de Xerox.

Se expondrá a continuación algunas de las limitaciones de este modelo, así como soluciones que se han propuesto.

1. *La morfología no concatenativa:* con este nombre se conoce a los procesos morfológicos donde los elementos involucrados no están en estricto contacto local. En español el caso más abundante es el de la modificación de la raíz verbal ante determinadas desinencias verbales. Ejemplo: /cont-/ diptonga en /cuent-/ sólo cuando se le adjunta alguno de estos morfemas: -o, -as, -a, -an, -e, -es, -en. Sin embargo, con otros sufijos como -amos, -áis, -aba, -emos, -éis, -aré, etc., no se produce la diptongación. Esta situación obliga a definir muy detalladamente el contexto. Se ha demostrado que la mayoría de los fenómenos no concatenativos conocidos se pueden expresar mediante autómatas de estados finitos. Al fin y al cabo son procesos que se dan sobre cadenas de una longitud finita abarcable. El resultado son unas tablas de transición enormes que se procesan muy lentamente. Como reconoce Antworth (1990:12), "la aproximación de estados finitos a la morfología no concatenativa peca tanto de adecuación notacional como de eficiencia computacional". En los años noventa se han propuesto varias estrategias para tratar fenómenos morfológicos tan complejos como los patrones triconsonánticos de las lenguas semíticas. La solu-

ción consiste en crear descripciones en varios niveles y encadenar varios transductores. Obviamente, de este modo se superan las limitaciones del modelo en dos niveles, pero el modelo pierde su seña de identidad.

2. *Las representaciones no lineales:* la representación fonológica de este modelo es estrictamente lineal. Por ello, elementos de carácter suprasegmental como el acento, la longitud o el tono no pueden representarse adecuadamente. Sin embargo, la solución de tener varios niveles de representación puede valer también para estos casos.
3. *La morfotáctica:* es un hecho claro que el modelo de dos niveles trata mucho más satisfactoriamente la parte fonológica (es decir, la alomorfia) que la parte sintáctica y léxica (las reglas de formación de palabras). Las reglas en la morfología en dos niveles únicamente especifican las correspondencias fonológicas, y toda la información morfotáctica se incorpora directamente en las entradas léxicas. Entre otras cuestiones, esta estrategia impide el tratamiento apropiado de morfemas discontinuos como los de *en+roj+ecer* o *a+grand+ar*, donde en+...+ecer y a+ ...+ar son partes discontinuas del mismo sufijo: derivan verbos a partir de adjetivos y son inseparables, pues no existen verbos como *rojecer o *grandar. Para dar cuenta de estos fenómenos, en el diccionario se tienen que duplicar las raíces adjetivales, con un coste importante en capacidad de almacenamiento. Otro problema relacionado con la estrategia de codificar la formación de palabras en las entradas léxicas es que la información proporcionada no es muy útil para las gramáticas sintácticas: únicamente se da una lista de las glosas de cada uno de los morfemas que se han reconocido. Esto supone que a veces no se indica ni siquiera la categoría de la palabra. Dicho de una manera clara, este modelo permite reconocer morfológicamente las palabras, pero si se quiere aprovechar para el posterior tratamiento sintáctico hay que convertir la información en un formato de estructuras de rasgos. Eso es precisamente lo que se ha desarrollado en la segunda versión de PC-KIMMO, que incluye la posibilidad de integrar información en forma de rasgos, y su salida se puede tomar directamente como entrada para una gramática de unificación, PC-PATR.

En resumen, podemos decir que la evolución del modelo en dos niveles ha ido en el sentido opuesto a sus orígenes y señas de identidad: dos únicos niveles de representación y utilización de autómatas finitos. Para algunos fenómenos complejos estos modelos se quedan ciertamente cortos y, aunque puedan tratarlos de alguna manera, su eficiencia se resiente tanto que pierden gran parte de su atractivo. Con todo ello, creemos que la utilización de este modelo y, en general, los autómatas de estados finitos para el tratamiento morfológico, puede ser lo más idóneo si la lengua en cuestión cuen-

ta con una alomorfía muy regular (es decir, condicionada fonológicamente) y no tiene abundantes casos de fenómenos no concatenativos y suprasegmentales. A este respecto, el español se queda a medio camino. En el siguiente apartado veremos otra aproximación que insiste más en la parte sintáctica que en la fonológica.

4.2.2. Morfología basada en la unificación y rasgos

Esta aproximación a la morfología destaca por la importancia de las reglas de combinación de morfemas y de la codificación de la información morfosintáctica. Para ello utilizan una gramática independiente del contexto aumentada con rasgos, exactamente igual que las que se utilizan para la sintaxis. De hecho, uno de los atractivos de este modelo es poder utilizar el mismo formalismo para tratar todo tipo de unidades lingüísticas significativas, desde el morfema hasta el discurso. Por tanto, en esencia utiliza los mismos recursos que los expuestos en el apartado 3.2.5.

El tratamiento de la morfología mediante gramáticas de rasgos cuenta con varios modelos, entre ellos AMPLE y STAMP. En la exposición seguiremos nuestro propio modelo, GRAMPAL, aplicado a la morfología del español (Moreno, 1991; Moreno y Goñi, 1995; Goñi *et al.*, 1997).

GRAMPAL está formado por una gramática con reglas morfológicas y un lexicón. Las entradas del diccionario son alomorfos, en lugar de palabras flexionadas. Cada alomorfo incluye dos tipos de información, utilizando rasgos como mecanismo descriptivo:

1. *Información lingüística:* compuesta tanto por información gramatical (número, persona, género, tiempo, aspecto...) como por información léxica, en concreto el lema y la subcategorización (para las categorías léxicas pertinentes: verbo, nombre y adjetivo). Toda esta información será utilizada posteriormente por las reglas sintácticas de la gramática para seguir procesando la cadena de entrada.
2. *Información contextual:* una parte de los rasgos de la entrada léxica se encarga de expresar la morfotáctica. Por ejemplo, la conjugación es una información puramente combinatoria: establece que las raíces verbales de determinada conjugación sólo puede concatenarse con sufijos verbales de la misma conjugación. Esta información sólo sirve para restringir la combinación de morfemas, pero no aporta ninguna información gramatical o léxica que pueda ser utilizada en etapas siguientes del procesamiento lingüístico. Por tanto, la información contextual morfotáctica no se transmite hacia el nivel superior (la palabra) al producirse la unificación.

Siguiendo esta estrategia, las entradas léxicas para PEZ serán dos, una para cada alomorfo:

pez	pec
morfo-cat = raíz-n	morfo-cat = raíz-n
sint-cat = n	sint-cat = n
lex = pez	lex = pez
conc gen = masc	conc gen = masc
tipo-plu = no	tipo-plu = plu2
tipo-gen = inherente	tipo-gen = inherente

Los morfemas de número y género cuentan con dos y tres entradas, respectivamente:

Alomorfos de número		*Alomorfos de género*		
s	es	o	e	a
morfo-cat = suf-n	morfo-cat = suf-n	morfo-cat = suf-n	morfo-cat = suf-n	conc num = sing
conc num = plu	conc num = plu	conc gen = masc	conc gen = masc	conc gen = fem
tipo-plu = plu1	tipo-plu = plu2	conc num = sing	conc num = sing	morfo-cat = suf-n
		tipo-gen = mas1	tipo-gen = mas2	tipo-gen = fem

El cuadro 4.1 muestra la interpretación de los distintos rasgos, organizados por clases de información.

CUADRO 4.1. Interpretación de los rasgos.

Información gramatical	*Información morfoléctica*
Rasgos sintácticos: sint-cat (categoría sintáctica)	Rasgos en el verbo: conj (conjugación) tipo-raíz (tipo de raíz) tipo-des (tipo de desinencia)
Rasgos morfológicos: morfo-cat (categoría morfológica) conc gen (concordancia género) conc num (concordancia número) conc pers (concordancia persona) infov tiempo (inform. verbal de tiempo) infov modo (inform. verbal de modo)	Rasgos en la flexión nominal (nombre, adjetivo, determinante, pronombres, etc): tipo-plu (tipo de plural) tipo-gen (tipo de género)
Rasgos léxicos: lex (unidad léxica)	

Los valores posibles para los rasgos gramaticales son fácilmente deducibles. Por ejemplo, los valores posibles para *conc gen* (concordancia de género) son *masc* y *fem*. En cambio, los valores de los rasgos contextuales necesitan una exposición más detallada.

El rasgo *tipo-plu*, por ejemplo, se utiliza para distinguir los alomorfos de la raíz nominal que se concatenan con el alomorfo -s (*plu1*) de los que se unen con -es (*plu2*). El tercer caso, aquellos nominales que no tienen morfema explícito de plural como *crisis* o *virus*, llevan asignado *tipo-plu = no*. (También llevan *tipo-plu = no* los alomorfos que no pueden llevar ningún morfema de plural, como pez, que es la forma del singular.) De esta manera, toda raíz nominal lleva especificado el alomorfo flexivo que le corresponde. La regla morfológica, mediante la operación de unificación, comprobará si raíz y morfema flexivo tienen los mismos valores para los rasgos contextuales pertinentes.

Un ejemplo de rasgo contextual en el verbo es *tipo-raíz*. Con él se identifica cada una de las formas que componen un paradigma verbal en español (57 en total). Para ello se utiliza un código numérico. Por ejemplo, la decena se utiliza para el presente del indicativo y, dentro de las unidades, el 1 indica primera persona del singular, hasta el 6 que es la tercera persona del plural. Por tanto, 14 será el código para primera persona del plural (4) del presente de indicativo (1). Este rasgo tiene una función esencial, ya que con él podemos identificar qué alomorfos de la raíz van con qué alomorfos de la desinencia: cada alomorfo de la raíz lleva una lista con los códigos de las desinencias que se pueden concatenar con dicha raíz. Veamos un ejemplo con las entradas de /cont-/ y /cuent-/, y dos morfemas verbales:

cont		cuent	
morfo-cat	= raíz-v	morfo-cat	= raíz-v
sint-cat	= v	sint-cat	= v
lex	= contar	lex	= contar
conj	= cjg1	conj	= cjg1
tipo-raíz	= [14 15 21 22 23 24 25 26 31...]	tipo-raíz	= [11 12 13 16 51 52 53 56 82]
tipo-des	= reg	tipo-des	= reg

amos		o	
morfo-cat	= des	morfo-cat	= des
conc pers	= p1	conc pers	= p1
conc num	= plu	conc num	= sg
infov tiempo	= pres	infov tiempo	= pres
infov modo	= ind	infov modo	= ind
conj	= cjg1	conj	= {cjg1 cjg2 cjg3}
tipo-raíz	= 14	tipo-raíz	= 11
tipo-des	= reg	tipo-des	= reg

De acuerdo con esto, -amos sólo se puede concatenar con cont-, dado que su código 14 aparece en la lista de dicho alomorfo; lo mismo ocurre con -o y cuent-. Hay una restricción importante en cuanto a las listas: únicamente los rasgos contextuales pueden llevar como valor una lista. Por ejemplo, el rasgo de conjugación del alomorfo -o tiene como valores posibles una lista con las tres conjugaciones: am-o, tem-o, part-o. Por otra parte, los rasgos gramaticales sólo pueden llevar valores únicos. Debido a esta restricción, hay que duplicar algunas entradas cuando el alomorfo de la desinencia tiene un único significante pero dos significados (por ejemplo, -amos identifica al presente y al pretérito indefinido), ya que en rasgos como el tiempo verbal no se pueden utilizar las listas.

La gramática está compuesta por muy pocas reglas: 2 para conjugar todos las formas verbales sintéticas y 4 para la flexión nominal. Se muestran las dos reglas de la flexión verbal, en formato PATR:

Regla para los verbos regulares	*Regla para los verbos irregulares*
palabra → raíz-v des	palabra → raíz-v des
<X1 conj> = <X2 conj>	<X1 conj> = <X2 conj>
<X1 tipo-raíz> = 100	<X1 tipo-raíz> = <X2 tipo-raíz>
<X1 tipo-des> = <X2 tipo-des>	<X1 tipo-des> = <X2 tipo-des>
<X0 sint-cat > = <X1 sint-cat>	<X0 sint-cat> = <X1 sint-cat>
<X0 conc> = <X2 conc>	<X0 conc> = <X2 conc>
<X0 infov> = <X2 infov>	<X0 infov> = <X2 infov>
<X0 lex > = <X1 lex>	<X0 lex> = <X1 lex>

Ambas reglas son idénticas salvo en la comprobación del rasgo *tipo-raíz*. Dado que los verbos regulares sólo tienen un alomorfo de la raíz, era redundante e innecesario tener la lista de las 57 formas en el valor de tipo-raíz. Por eso, se prefirió partir la regla en dos y asignar un código genérico (100) a todas las raíces regulares.

Las reglas deben interpretarse de la siguiente manera: una palabra (entre otras posibilidades) se forma por la concatenación de una raíz verbal y una desinencia. Para que se produzca dicha concatenación tienen que unificar todos los rasgos. En primer lugar, unifican los rasgos contextuales *(conj, tipo-raíz* y *tipo-des);* después se transmite la información al nodo palabra (X0). La categoría sintáctica y el lema se toman de la raíz verbal (X1), mientras que la información verbal y la concordancia se heredan de la desinencia (X2).

Análogamente, la flexión nominal se especifica mediante 4 reglas, siguiendo la misma estrategia: una regla sintagmática concatena una raíz-nominal y

un sufijo flexivo (número y/o género, según los casos). Hay dos rasgos contextuales, *tipo-gen* y *tipo-plu* que controlan la combinación de los alomorfos apropiados; y la información léxica se hereda de la raíz y la morfosintáctica del morfema flexivo.

De nuevo comprobamos que los sistemas basados en la unificación se caracterizan por un componente gramatical muy reducido y por un lexicón muy prolijo en información, en ocasiones bastante redundante. Pero si comparamos la tarea de escribir reglas y entradas de diccionario en un modelo de estados finitos y en un modelo de unificación, parece claro que el segundo presenta características más cómodas para el morfólogo computacional. En el proyecto ARIES, por ejemplo, se han desarrollado unas herramientas para elaborar un diccionario de más de 40.000 lemas basándose en el modelo de GRAMPAL. En concreto, hay dos programas muy útiles: uno es un clasificador de verbos y otro un generador automático de entradas léxicas. El primero lo que hace es tomar un verbo no registrado en el lexicón y lo clasifica en alguno de los paradigmas del sistema. Una vez clasificado, el generador automático infiere los alomorfos de la raíz y crea las entradas de diccionario completas. Esta automatización permite ampliar con esfuerzo mínimo cualquier verbo que se encuentre en un corpus y no esté en el diccionario. Todo ello ha sido posible gracias a la exhaustiva clasificación de los verbos en paradigmas. Tarea, aunque costosa, tratable, ya que, como se mencionó anteriormente, con unas 40.000 entradas se puede recoger buena parte del léxico básico de una lengua como el español. Goñi *et al.* (1997) presentan la descripción más detallada y reciente de la plataforma léxica ARIES.

4.2.3. Comparación entre la morfología en dos niveles y la morfología basada en la unificación

Se empezará con los puntos en común:

1. Ambas son básicamente concatenativas: sus fundamentos formales (los autómatas de estados finitos y las gramáticas independientes del contexto) suponen una limitación inherente en el tratamiento de fenómenos no locales o discontinuos. El uso de diferentes niveles de representación y de rasgos contextuales, respectivamente, permite manejar los procesos morfológicos no concatenativos, aunque no de una manera eficiente. Éste es sin duda uno de los puntos débiles de ambos modelos.
2. Son declarativas: esta característica es muy interesante, pues permite la bidireccionalidad (análisis y generación) sin modificaciones en la gramática.

3. Cobertura (casi) completa de los fenómenos morfológicos de muchas lenguas: ambos modelos han demostrado que permiten desarrollar completas descripciones en diferentes lenguas. Para las formas muy irregulares se recurre a la lexicalización directa, de manera que todas las formas se pueden tratar. En cuanto a la universalidad, el modelo en dos niveles parece aplicable a un rango mayor de lenguas, aunque no necesariamente de manera más satisfactoria. El modelo de unificación no es práctico en lenguas con una alomorfía muy abundante, porque el tamaño del lexicón sería inmanejable.
4. Eficiencia computacional: ambos modelos reconocen y generan palabras en tiempo real y su funcionamiento es robusto. Probablemente, la morfología en dos niveles es más eficiente por emplear autómatas de estados finitos, aunque no siempre es cierto: las complejas interacciones en sistemas con muchas reglas y tablas de transición extensas hacen disminuir la eficiencia del modelo. Por otra parte, los sistemas basados en la unificación pueden recurrir a muchas técnicas para mejorar la búsqueda en el diccionario.

En cuanto a las diferencias, podemos señalar una fundamental: el modelo de dos niveles da cuenta básicamente del *contexto fonológico* y el modelo basado en rasgos trata únicamente el *contexto morfológico*. Las lenguas suelen tener una mezcla de alomorfía condicionada fonológica y morfológicamente. La proporción de una y otra puede dar la pista de cuál de los dos es el más apropiado para una lengua dada. Incluso la aplicación de métodos mixtos puede ser la solución óptima, como han propuesto algunos autores (Ritchie *et al.*, 1992). En cualquier caso, parece que los sistemas de uno u otro tipo tienden a incorporar características del método opuesto (Antworth, 1994).

Para terminar esta sección, que se ha alargado por la importancia que supone la morfología en el tratamiento computacional del español, se señalará una serie de fenómenos que no se tratan de manera satisfactoria por estos dos modelos. Son los fenómenos que tienen que ver con el *contexto paradigmático*: las irregularidades fuertes, las duplicaciones, las formas defectivas y otros tipos de excepciones que normalmente se incluyen directamente en el diccionario. Se verá precisamente en la sección dedicada a la lexicografía computacional un marco formal para representar la información tanto regular como excepcional de una manera compacta.

4.3. Procesamiento sintáctico

En este apartado se verán algunos fenómenos sintácticos generales y también algunos particulares del español. El objetivo es plantear los problemas básicos que se deben abordar en cualquier gramática computacional.

4.3.1. Dependencias no acotadas o a larga distancia

Ya se expuso en la sección dedicada a las gramáticas independientes del contexto (3.2.4) las dificultades con que éstas se encontraban para procesar constituyentes discontinuos: oraciones interrogativas, relativas, extraposiciones, etc. En una gramática transformacional estos fenómenos se tratan con transformaciones de movimiento de un constituyente a su posición superficial. Como las gramáticas transformacionales no se emplean habitualmente como modelo para construir una gramática computacional, no se entrará en más detalles. Otra posibilidad son las gramáticas de unificación y rasgos, que es el modelo más extendido en la actualidad.

Cada gramática puede aportar una solución diferente, pero muchas de ellas están basadas en el tratamiento que de estos fenómenos se hace en GPSG. La idea de partida es que en un constituyente falta un elemento esencial para que esté completo. Ese elemento que falta deja una marca en forma de rasgo en el constituyente del que ha salido: este rasgo suele denominarse *slash*. Por ejemplo, SV/SN quiere decir que a ese sintagma verbal le falta un sintagma nominal para estar completo. Ese rasgo es, en realidad, la definición de una categoría que incluye sus rasgos de concordancia. En el proceso de análisis, se utilizan las características de este rasgo *slash* para reconocer al elemento desplazado y asignarlo a su posición normal. Observemos cómo se aplica este tratamiento a una oración del tipo *¿Qué lee Eloísa?* El árbol de análisis sería como el mostrado en la figura 4.4. Para llegar hasta él hay que añadir el rasgo *slash* a las reglas de la gramática:

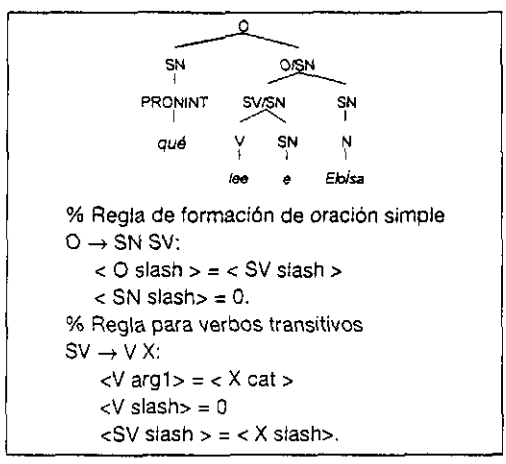


Figura 4.4. Ejemplo de dependencia a larga distancia.

Estas reglas deben interpretarse de la siguiente manera: la regla para verbos transitivos indica que un verbo como *leer* toma un argumento del tipo arg1 perteneciente a la categoría X. Para permitir que ese argumento pueda encontrarse desplazado fuera del SV, como ocurre en la oración interrogativa del ejemplo, se añaden las expresiones <SV slash> = <X slash>, que al establecer que el valor del rasgo *slash* en el SV es igual al valor de *slash* en la categoría X, pasa las características del argumento ausente (X cat) al nivel superior para que sean comprobadas cuando se encuentre dicho argumento. Por su parte, el valor de *slash* en el verbo es nulo (<V slash> = 0), es decir, el elemento verbal no pasa sus características al nivel superior, puesto que ya se encuentra presente en este sintagma.

La regla de formación de la oración establece que el valor de *slash* en la oración será el mismo que tuviera en el SV, mientras que el SN no pasará las características de su rasgo *slash* al nivel superior (indicado por <SN slash> = 0). Para que la oración quede completa, hay que encontrar un elemento de categoría X cuyos rasgos coincidan con el valor de *slash* en O.

Otra estrategia diferente consiste en dar un tratamiento sintagmático superficial a todos los constituyentes y establecer algún procedimiento que proyecte la estructura superficial de constituyentes en una estructura más abstracta, donde los elementos principales hayan "recuperado" su posición canónica. Esta estrategia estaría inspirada en los tratamientos de la gramática transformacional o en LFG. Por ejemplo, la gramática del proyecto PROTEUS de la Universidad de Nueva York emplea un tratamiento de este tipo: las reglas sintagmáticas de la gramática tienen incorporadas unas extensiones (escritas en Lisp) que proyectan la estructura de constituyentes en una estructura regularizada similar a la estructura funcional de LFG.

4.3.2. La coordinación

Las estructuras coordinadas aparecen con mucha frecuencia en aplicaciones con texto real. Por tanto, para que una gramática computacional sea robusta, debe tener previsto un tratamiento bastante extenso de este tipo de estructuras. De nuevo, las gramáticas sintagmáticas independientes del contexto tienen pocos recursos para afrontar el problema, ya que la solución es duplicar las reglas de todas las categorías. Por el contrario, en una gramática de unificación y rasgos sólo sería necesario añadir a la gramática una regla del tipo:

$$X0 \rightarrow X1\ C\ X2:$$
$$\langle X0\ cat\rangle = \langle X1\ cat\rangle$$
$$\langle X0\ cat\rangle = \langle X2\ cat\rangle$$

donde se establece que un elemento X que pertenece a determinada categoría puede estar constituido por la coordinación de dos elementos de esa misma categoría, siendo C (conjunción) el nexo de unión entre ellas. Esta regla da cuenta del caso más simple, es decir, cuando sólo hay dos términos coordinados, por ejemplo *Abelardo ama a Eloísa y Calisto adora a Melibea*. Sin embargo, nada impide que se pueda aplicar sucesivas veces, permitiendo el análisis de coordinaciones de varios términos: *Abelardo ama a Eloísa, Calisto adora a Melibea y Cyrano no encuentra pareja*. De esta forma, se aumenta considerablemente la capacidad de análisis de la gramática, pero seguimos teniendo varios problemas:

1. *Asignación de la estructura interna del elemento coordinado.* Una regla como la que se ha añadido da lugar a un estructura en cascada en la que cada elemento que se suma está coordinado solamente con el elemento inmediatamente anterior y éste a su vez con el anterior, etc. En muchos casos, este análisis no es correcto, ya que los elementos coordinados entre sí deben formar una estructura plana y estar situados al mismo nivel, pues todos ellos realizan conjuntamente la misma función respecto al constituyente del que dependen. Una posible solución es permitir la repetición dentro de la misma regla.
2. *Elipsis de elementos.* La regla anterior no puede dar cuenta de una estructura tan frecuente como *Abelardo ama a Eloísa y Calisto a Melibea* donde, al coordinar las oraciones, se hace desaparecer el elemento repetido. La omisión del verbo, como ocurre en este ejemplo, es el caso más extremo y difícil de tratar, pues sólo teniendo en cuenta que nos encontramos ante una estructura coordinada podemos reconocer que el segundo elemento es una oración. Por el contrario, oraciones como *Abelardo ama a Eloísa y le escribe poemas* son más sencillas de analizar, ya que admiten al menos dos soluciones. Lo más simple es asumir que se trata de dos SV coordinados entre sí compartiendo el mismo sujeto. Para este análisis, la regla que hemos enunciado más arriba sería suficiente. Otra posibilidad, si la gramática ya está preparada para analizar sujetos omitidos, es construir primero una oración completa en el segundo elemento y luego coordinarla con la primera. Tampoco sería necesario añadir reglas, pero quedaría pendiente la cuestión de si el segundo sujeto es o no el mismo en ambas oraciones. En resumen, la solución pasa por reconstruir el constituyente incompleto a partir de su antecedente. Por último, hay otra estrategia de resolución de elementos elididos basada en la semántica. En este caso se busca una entidad en la forma lógica que pueda ser el antecedente del elemento elidido. Es decir, la información del constituyente omitido se recupera en la representación semántica y

no en la sintáctica. En todos los casos el problema central es reconocer el antecedente.

3. *Ambigüedad sintáctica.* Consideremos la oración *Los padres de los soldados y los mandos asistieron a la ceremonia.* La regla propuesta nos proporcionaría al menos dos análisis:

   a) Coordinaría entre sí los dos sintagmas nominales inmediatos *(los soldados y los mandos)* y luego construiría con ambos un sintagma preposicional dependiente de *los padres.*
   b) Construiría el SN *los padres de los soldados* y luego lo coordinaría con *los mandos.*

   En este ejemplo, el primer análisis da lugar a una interpretación muy poco probable, pero tampoco se puede imponer una restricción que excluya esa posibilidad de la gramática, ya que en ocasiones será la que asigne la estructura deseada. Por ejemplo, si modificamos un poco la oración anterior, *Los padres de los alumnos y las alumnas asistieron a la ceremonia*, vemos que el análisis que se acaba de desechar por incorrecto es ahora claramente preferible a la segunda posibilidad.

En resumen, la introducción de reglas para tratar la coordinación añade un alto grado de ambigüedad a la gramática y presenta aspectos muy difíciles de abordar (elisión de elementos clave).

4.3.3. El orden de constituyentes

Muchas de las gramáticas existentes en la actualidad se han desarrollado para una lengua como el inglés en la que el orden de los constituyentes oracionales (SVO) es prácticamente fijo. Sin embargo, al intentar desarrollar sistemas para otras lenguas, entre ellas el español, el lingüista computacional tropieza con una dificultad importante: hay más posibilidades de ordenación de los constituyentes oracionales, situándose en el extremo algunas lenguas en las que el orden es libre. Para el español, en concreto, además del orden SVO, habría que permitir al menos las siguientes posibilidades:

VSO: *Han venido los chicos a comer.*
VOS: *Estudiaron el problema los delegados.*
OVS: *El libro lo ha escrito Juan.*

Las reglas de la gramática que se propusieron anteriormente (en el apartado dedicado al tratamiento de elementos discontinuos) aplicadas a la ora-

ción *Come el perro el hueso* no producirían un análisis, ya que el orden esperado es:

$$O \rightarrow SN\ SV$$

Sería necesario, por tanto, añadir otra regla:

$$O \rightarrow SV\ SN$$

Con esta adición, la gramática sería capaz de producir un análisis, pero el problema no estaría resuelto. Al aplicar la regla de construcción del SV transitivo se asignaría la función de objeto directo (arg-1) al primer SN encontrado a la derecha del verbo *(el perro)*. Posteriormente, la regla que acabamos de escribir asignaría la función de sujeto (arg 0) al SN encontrado en último lugar *(el hueso)*, de manera que la asignación de funciones sería errónea.

Es evidente, por tanto, que el tratamiento sintagmático estrictamente superficial es inapropiado para tratar la variedad de posibilidades de orden de constituyentes. (Chomsky utilizó en 1957 esta inadecuación del análisis en constituyentes inmediatos para proponer su gramática transformacional.)

El tratamiento del orden de constituyentes en LT generalmente se ha basado en presentar una representación superficial más o menos plana y otra representación más abstracta, donde los constituyentes aparecen en el orden lógico. Por una parte, tenemos la versión transformacional, que asume una configuración abstracta sintagmática igual para todas las lenguas (la teoría de la X'). Por otra parte, tenemos la versión relacional, donde las relaciones superficiales se proyectan en relaciones de dependencia núcleo-complemento o en funciones gramaticales. Ésta es la estrategia de las gramáticas de dependencias y las gramáticas relacionales y de alguna teoría basada en la unificación como LFG. Sin embargo estas estrategias, aunque atractivas desde el punto de vista teórico, suponen un coste evidente en el procesamiento computacional, ya que exigen construir un segundo nivel de representación sintáctico.

A continuación se verán dos estrategias computacionales para dar cuenta del orden variables que tienen como punto de partida un análisis sintagmático superficial, donde se recogen las posibles combinaciones. Por ejemplo: $O \rightarrow SN\ SV\ SN$; $O \rightarrow SV\ SN$; $O \rightarrow SV$; $O \rightarrow SP\ SN\ SV$, etc.:

1. *Asignación de función sintáctica y cancelación de constituyentes.* Si utilizamos una gramática sintagmática como la mostrada más arriba, nuestra principal preocupación es cómo asignar correctamente las

funciones sintácticas (sujeto, objeto directo, etc.), dado que esta asignación típicamente depende de la posición estructural: el SN que está detrás del V es el objeto directo, etc. ¿Cómo podemos asignar correctamente dicha función sin recurrir al contexto sintagmático? Hay dos medios: uno, el más seguro, es recurrir a los casos morfológicos; el otro, ayudarse de partículas (generalmente preposiciones) que marcan los constituyentes. La opción de los casos morfológicos sólo es posible, obviamente, en aquellas lenguas que cuenten con ellos. Para el español, por tanto, no es una solución válida. Sólo nos queda el marcado por medio de una preposición. Este método es eficaz en español para distinguir los complementos verbales que son sintagmas preposicionales: el objeto indirecto, los instrumentales, locativos, temporales, etc. Por tanto, podemos asociar determinadas preposiciones con sus correspondiente funciones. La función de objeto directo, sin embargo, es problemática ya que, salvo en los objetos humanos que van precedidos de la preposición *a*, no hay partícula que los identifique *(vio Lucía a su novio con otra*, frente a *vio Lucía el libro con una mancha)*. Todavía tenemos la posibilidad de recurrir a la concordancia con el verbo para reconocer el sujeto: *explicó los problemas el profesor, estudiaron el problema los expertos*. Pero la ambigüedad estructural sigue produciéndose en los casos donde sujeto y objeto directo tienen el mismo valor para número: *estudió el problema el alumno*.

Una vez asignada una función a cada sintagma, el siguiente paso es ir cancelando argumentos de la lista de subcategorización del verbo. Para ello, habría que adaptar la estrategia del parser:

a) Habría que empezar por reconocer el verbo principal entre todos los componentes.
b) A partir del sintagma más a la derecha del verbo, se comprobaría qué funciones sintácticas cumpliría dentro del marco de subcategorización. Cada sintagma reconocido se cancelaría de la lista.
c) Una vez procesados todos los sintagmas a la derecha, se continuaría por el primero a su izquierda, si es que existe, hasta agotar todos los constituyentes.

Esta estrategia de *parsing* se utiliza habitualmente en los sistemas que cuentan con una morfología con casos, como el alemán. Desafortunadamente, la mayoría de los programas de uso público cuentan con un algoritmo de *parsing* basado en el orden estricto de las gramáticas independientes del contexto (véase 3.2.4.). Además, esta estrategia tampoco resuelve toda la ambigüedad estructural del espa-

ñol. Como se ha visto, oraciones del tipo *resolvió el problema el alumno* producirían siempre dos análisis (uno con sujeto *el problema* y el otro con *el alumno).*

2. *Utilización de restricciones semánticas.* Cualquier hablante competente de español sabe distinguir el sujeto de la oración *resolvió el problema el alumno:* un verbo como "resolver" exige que su agente sea una entidad dotada de razonamiento (por ejemplo, el *ordenador resolvió el problema planteado por el campeón de ajedrez).* Es decir, cada verbo restringe semánticamente el tipo de argumentos que puede llevar. Esto se conoce por *restricciones seleccionales* (terminó que acuñó Chomsky en 1965). Así como la subcategorización es un tipo de selección (o restricción) sintáctica, las restricciones seleccionales implican una selección semántica. Inspirada en esta estrategia cognitiva, otra forma de tratar el orden libre de constituyentes es recurrir a la semántica para resolver ambigüedades estructurales. Para ello tenemos que construir un modelo semántico para cada verbo, donde se establezca una relación entre funciones sintácticas (sujeto, etc.) y funciones semánticas (agente, paciente, etc.). Si contamos con dicho modelo semántico, entonces la desambiguación estructural se puede realizar en dos momentos:

   a) Después del análisis sintáctico: en este caso llegarán varios análisis posibles y el modelo permitirá seleccionar el apropiado. En el ejemplo anterior, elegirá el análisis donde *el estudiante* es el sujeto. La desventaja de esta estrategia es que la resolución de la ambigüedad se deja para el último componente, de manera que se generan diferentes análisis que finalmente se tienen que descartar.
   b) Durante el análisis sintáctico: esta opción consiste en que la información semántica se consulta en el momento de construir la estructura sintáctica. Por ejemplo, al intentar asignar la función de sujeto a *el problema,* comprobaría si es el tipo de agente que admite el verbo *resolver.* Esta estrategia es sin duda la que aporta mejores resultados en cuanto a precisión, ya que generalmente los análisis finales suelen ser apropiados. En su contra tiene que es la más costosa en términos de eficiencia computacional, pues consulta *sistemáticamente el modelo semántico, y en muchas ocasiones no* es necesario, ya que no hay ambigüedad estructural que resolver.

Esta estrategia tiene un inconveniente adicional: se basa en una especificación muy fina y exhaustiva de cada concepto semántico. Esto es factible para dominios temáticos muy restringidos, pero con modelos semánticos de cierto tamaño la eficiencia se resiente enor-

mente. Por supuesto, para texto de dominios no restringidos los modelos semánticos producen un resultado pobre.

Hay una tercera opción para tratar el orden de constituyentes en lenguas como el español: utilizar algún tipo de heurística o bien una gramática probabilística. Es sabido que ciertas configuraciones son más frecuentes que otras: *habló Juan con su madre* es más habitual que *con su madre Juan habló*. Con ese argumento, podemos tratar de dar prioridad a una configuración sobre las otras posibles. Esta preferencia se puede basar en una estimación más o menos subjetiva (equivalente a la que se acaba de hacer con el ejemplo) o en una estimación empírica. En el segundo caso, tendremos que consultar un corpus y calcular las estadísticas que nos permitan tomar una decisión fundamentada en la frecuencia de aparición de las distintas configuraciones de constituyentes. Se discutirá esta técnica en el siguiente capítulo.

4.3.4. *Elementos nulos o vacíos*

La existencia de constituyentes nulos o vacíos de sustancia fónica es uno de los temas más controvertidos de la LT, sobre todo en morfología y sintaxis. Se han dado argumentos de todo tipo para defender su pertinencia empírica. No se entrará en detalles pues no es el espacio apropiado. Simplemente se situarán las dos líneas de pensamiento:

- *En favor de las categorías vacías:* la principal ventaja es que proporcionan una representación simétrica. Por ejemplo, en una coordinación como *Los anticuarios compran y venden muebles antiguos*, hay suficiente evidencia para proponer que el objeto directo elidido de *compran* es *muebles antiguos*. Una estructura puramente superficial no daría cuenta de la interpretación correcta, mientras que la que contiene posiciones vacías sí (véase la figura 4.5). En general, la mayoría de los lingüistas aceptan la pertinencia de algún elemento vacío en la representación cuando está muy justificado por la interpretación. Sin embargo, muchos están en contra de su uso generalizado para proponer estructuras abstractas muy artificiales. Como se ha señalado en numerosas ocasiones, la búsqueda de la armonía y la elegancia teóricas a veces se vuelve contra el carácter empírico de la ciencia.
- *En contra de las categorías vacías:* la postura más radical argumenta que si el signo lingüístico está compuesto de un significante y un significado, ¿cómo se justifica la existencia de un significado sin significante? Esta aproximación entiende que sólo se puede evaluar la pertinencia de un constituyente si tiene una realización manifiesta. La

principal crítica a esta postura es que su carácter fundamentado en la observación directa impide capturar generalizaciones abstractas.

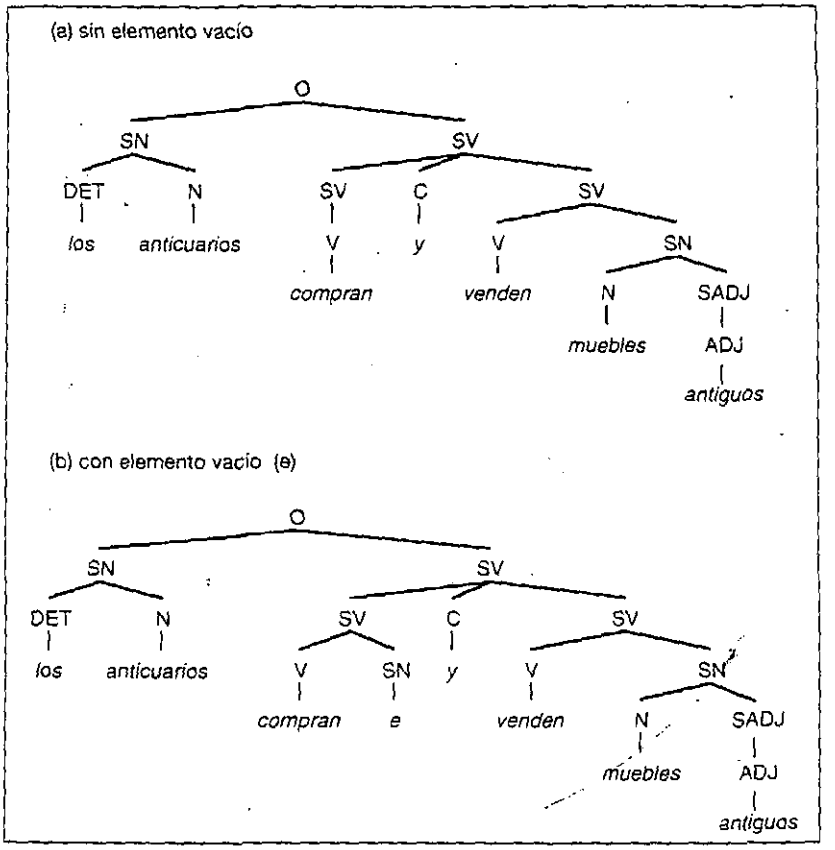


Figura 4.5. Ejemplo sin elemento vacío y con elemento vacío.

Sirva esta introducción al problema para dejar constancia de que diferentes modelos computacionales han adoptado una y otra postura. Es perfectamente posible introducir elementos vacíos en el procesamiento computacional, aunque no estén en la cadena de entrada. Sin embargo, por motivos de eficiencia es deseable hacer un uso lo más restringido posible de elementos nulos. Se verá un ejemplo de la complejidad que introducen en el *parsing*.

El español es una lengua donde habitualmente se omite el sujeto si está implícito en el contexto o se conoce por la concordancia del verbo. De hecho, la forma no marcada en español es la de la omisión del sujeto frente a la oración con sujeto explícito: *fuimos al zoo, nosotros fuimos al zoo*. La segunda oración es redundante y sólo se emplea cuando se quiere resaltar la información sobre el agente de la acción. Si adoptamos la estrategia de registrar los sujetos en todas nuestras representaciones necesitaremos alguna manera de expresar los sujetos nulos. Normalmente se utiliza algún procedimiento especial para "insertar" un símbolo que represente el elemento elidido, por ejemplo un 'Ø'. Por tanto, permitimos que el Ø aparezca en nuestras reglas:

$$O \rightarrow SN\ SV$$
$$SN \rightarrow \{DET\ N \mid \emptyset\}$$

La interpretación de la regla sería: "sustitúyase el SN por DET N o por Ø". En el segundo caso, no habría que buscar en el lexicón un elemento terminal para finalizar la derivación.

Esta técnica es razonablemente eficiente si hay pocos contextos donde se permitan elementos vacíos, pero desgraciadamente para el escritor de gramáticas computacionales del español la elisión del sujeto es muy frecuente. Téngase en cuenta que si permitimos un elemento nulo opcional, cada vez que se aplique la regla probará también esa opción. Eso, al menos, duplica los intentos en la construcción de estructuras durante el *parsing*. Por supuesto, una manera de controlar los efectos indeseados del elemento vacío es especificar restricciones muy precisas. Por ejemplo, poner un "penalización" a la opción Ø, o simplemente no permitirla cuando ha tenido éxito otra opción anterior.

En español la situación se complica con el orden variable de constituyentes: ¿debemos incluir la posibilidad del sujeto nulo en todas las combinaciones posibles? Esto aumentaría aún más la combinatoria de elementos vacíos. La experiencia indica que para el español es preferible escribir gramáticas que no tengan sujetos vacíos.

En resumen, el problema fundamental de los elementos nulos en el procesamiento computacional es que siempre tienen éxito (y por tanto construyen alguna estructura), ya que por definición no tienen que ser contrastados con ningún elemento terminal.

4.3.5. Algunos fenómenos típicos del español

Se ha visto en apartados anteriores que ciertos fenómenos sintácticos característicos del español parecen necesitar un algoritmo de procesamiento algo diferente al utilizado para el inglés. Al igual que se comprobó en el aná-

lisis morfológico, algunos métodos se adaptan mejor a unos fenómenos que a otros. Las gramáticas sintagmáticas y los algoritmos de *parsing* más habituales reflejan la estructura sintáctica del inglés (algo que se ha criticado en numerosas ocasiones por parte de corrientes tipológicas). Como consecuencia de todo ello, aunque sea bastante más atractivo y económico hablar de un método universal para todas las lenguas, es más eficiente emplear distintos métodos que se adapten a las características particulares de cada lengua.

Se dará un ejemplo para defender esta hipótesis. Nuestra experiencia en la construcción de la gramática del español para el proyecto PROTEUS nos indica que utilizando un modelo inspirado en el inglés se puede conseguir una cobertura similar a la gramática de esta lengua. Sin embargo, hay una gran diferencia en cuanto a la eficiencia: para la misma oración, la gramática del español prueba muchas más derivaciones debido a la multiplicación de reglas para dar cuenta de los posibles órdenes y elementos elididos.

Una propuesta de investigación interesante para el español sería explorar las posibilidades de un procesamiento basado en el verbo. Esta aproximación podría sacar partido de la rica información de todo tipo que contiene el verbo: no sólo sobre estructura eventiva o argumental, o sobre información temporal y aspectual, sino también sobre el sujeto. Esta estrategia no debería ser incompatible con una gramática sintagmática en los casos donde las relaciones locales de dominio y precedencia sean claramente aplicables, por ejemplo, en la estructura interna de los sintagmas. Curiosamente, se han propuesto diferentes combinaciones de métodos simbólicos y estadísticos, pero no es habitual encontrarse con propuestas híbridas dentro de los modelos simbólicos, como la que se ha sugerido de combinar una estrategia sintagmática para constituyentes dentro de la proyección máxima del núcleo (es decir, sintagmas) con otra estrategia relacional para los constituyentes superiores.

4.4. Interpretación

El nivel interpretativo es el más universal, de modo que se pueden aplicar los mismos métodos para cualquier lengua, lo que no sucede con los fenómenos morfológicos y sintácticos. Por ello, a pesar de la importancia del componente interpretativo para el procesamiento del lenguaje natural, no se le dedicará el mismo espacio que a los componentes mencionados. Cualquier modelo semántico formal se puede adaptar sin mayor problema al tratamiento del español. Existen, por una parte, excelentes manuales de semántica formal en español, como el de Garrido (1988). Por otra parte, la exposición del procesamiento semántico y discursivo en Grishman (1986) parece insu-

perable en cuanto a sencillez. Por último, el libro de Allen (1987; 1995) contiene la presentación más extensa escrita hasta la fecha sobre las cuestiones interpretativas. Por tanto, aquí se esbozarán los aspectos básicos de interpretación en LC.

4.4.1. Consideraciones previas

En la mayoría de las aplicaciones PLN no es suficiente con obtener un análisis estructural de las oraciones mediante alguna de las técnicas que se han visto hasta ahora. El objetivo final es determinar el significado del texto que se está tratando. En otras palabras, necesitan incorporar un componente de interpretación. Dentro de este componente, se pueden distinguir tres niveles:

- *Semántica oracional:* la primera tarea del componente interpretativo es asignar un significado a cada una de las oraciones analizadas, pero la noción de significado es en sí misma poco específica. En realidad, lo que la mayoría de los sistemas PLN hacen es determinar las condiciones bajo las cuales una oración es verdadera. En este componente se resuelven las cuestiones que hacen uso del contexto inmediato.
- *Semántica del discurso:* en este nivel se almacena el conocimiento que permite relacionar entre sí el significado de las oraciones aisladas e integrarlo para formar unidades mayores. En él se realizan algunas operaciones que son fundamentales para la interpretación, como la resolución de la anáfora y la asignación de referentes a los elementos elididos. Este componente es necesario para que el sistema tenga conocimiento del contexto comunicativo en el que se están produciendo los mensajes y tiene en cuenta aspectos pragmáticos como las intenciones del emisor y del receptor.
- *Conocimiento del mundo:* casi todos los sistemas PLN disponen de un modelo que intenta representar formalmente el conocimiento del mundo real empleado por los hablantes para interpretar correctamente las emisiones lingüísticas. Este es el componente en el que la relación con otros campos de la Inteligencia Artificial es más clara.

4.4.2. Semántica oracional

Casi todos los sistemas PLN entienden la interpretación semántica como la proyección de las estructuras de la lengua natural en una lengua formal que permita al ordenador hacer un uso directo de la información. En la mayor

parte de los casos, la lengua en la que se proyectan las estructuras es alguna variante de la lógica formal. Ésta se basa en dos conceptos básicos: la composicionalidad del lenguaje y el significado entendido como condiciones de verdad.

El *principio de composicionalidad* se fundamenta en la famosa máxima de Frege: el significado del conjunto es el resultado del significado de las partes. De acuerdo con él, el significado de una oración se obtiene a partir del significado de los constituyentes que la componen. A su vez, el significado de éstos deriva del de los sintagmas que se encuentran dentro de ellos y así hasta llegar al significado de las unidades léxicas. (Incluso este análisis composicional puede llegar hasta el nivel de los morfemas: el significado de *librerías* se puede deducir de /libr-/ /-ería/ /-s/.) Este principio permite descomponer la tarea y hacerla practicable, aunque no resuelve otras cuestiones fundamentales como cuáles son las unidades mínimas de expresión del significado o de qué forma se combina el significado de las distintas partes.

El segundo concepto intenta resolver el problema de qué es el significado. Para definirlo necesitamos tener una referencia. La lógica clásica se basa en la *búsqueda de un referente* (que puede ser el mundo real o un mundo inventado) y *la comprobación del valor de verdad*, es decir, si el enunciado es cierto o falso.

A) Traducción a la forma lógica

Como se decía al comienzo, el problema se reduce a la traducción del enunciado en lengua natural a una lengua formal que cumpla las siguientes condiciones: no ser ambigua, tener reglas simples de interpretación e inferencia y tener una estructura lógica determinada por la estructura de la oración.

Normalmente, cuando se construye un sistema PLN, lo que buscamos es que el sistema realice algún tipo de acción en respuesta a nuestros mensajes: que nos devuelva los datos que hemos pedido, que mueva un robot en determinada dirección, que produzca una traducción de la oración, que cree un registro en una base de datos, etc. Esto supone traducir el mensaje en lengua natural a la lengua formal que entiende la base de datos o el robot al que estamos dando órdenes. Las lenguas formales utilizadas por esas aplicaciones deben, en principio, cumplir los requisitos que hemos enunciado. Para presentar este proceso de traducción de un modo general, se asumirá que estas lenguas son equivalentes a un formalismo lógico, ya sea éste la lógica de predicados, la lógica proposicional o cualquier extensión de ellas.

Las ventajas de utilizar la lógica de predicados de primer orden como lenguaje de representación semántica es que permite tanto asignar una denotación como inferir información (utilizando algún mecanismo deductivo). Sin

embargo, la lógica de predicados clásica no es capaz de expresar todos los tipos de información semántica, como por ejemplo la modalidad, los cuantificadores generalizados, información contradictoria, etc. Para tratar estos y otros aspectos se han desarrollado extensiones de la lógica de predicados como la semántica de los mundos posibles o la lógica no monótona.

El proceso parte del análisis sintáctico de la oración. Como se vio en apartados anteriores, en los modelos simbólicos lo más frecuente es que la gramática no se limite a crear una estructura de constituyentes, sino que reconoce y asigna las posiciones argumentales. Por tanto, la salida del análisis sintáctico es una estructura en la que se han determinado las funciones argumentales de los distintos constituyentes. Las estructuras funcionales resultantes están ya prácticamente formalizadas. De esta manera, el último paso, es decir, la traducción al formalismo lógico, es casi directa y puede realizarse mediante unas pocas reglas simples. Se muestra el esquema de este proceso con un ejemplo:

Lengua natural	*Abelardo ama a Eloísa*
Estructura de constituyentes	[O[SN Abelardo] [SV [V ama][[SP [P a] [SN Eloísa]]]]
Estructura funcional	(PRED ama SUJETO Abelardo OBJ-DIR Eloísa)
Forma lógica	amar (Abelardo, Eloísa).

Otra forma de abordar la cuestión es realizar el análisis sintáctico y el semántico simultáneamente. En esta aproximación, conocida como la hipótesis regla-por-regla *(rule-to-rule hypothesis)*, cada regla de la gramática lleva asociada una regla de interpretación semántica. Siguiendo la notación empleada por Gazdar y Mellish (1989), se pueden añadir reglas de este tipo a la gramática:

O → SN SV:
<O significado> = VERDADERO si <SN significado> ∈ <SV significado>
= FALSO en otro caso.

Una alternativa al uso de formas lógicas es utilizar algún tipo de representación gráfica de las estructuras semánticas, como por ejemplo las *redes semánticas*. Estas redes se utilizan en todos los niveles de la interpretación, por lo que se hablará más detalladamente de ellas en el apartado dedicado a *Conocimiento del mundo*.

B) Información semántica mediante rasgos

Los sistemas que emplean rasgos ofrecen la posibilidad de codificar información semántica en el lexicón para que esté disponible en cualquier momento en que pueda ser necesaria. Para ello, se puede utilizar un conjunto de rasgos que haya sido definido por una teoría lingüística o utilizar rasgos *ad hoc* que sean pertinentes en el dominio de aplicación. En cualquier caso, debe tratarse de un conjunto finito de atributos y valores.

La gramática puede entonces incorporar en sus reglas restricciones selecionales. Estas restricciones interrumpen un gran número de análisis que son posibles desde el punto de vista sintáctico, pero semánticamente incorrectos. De otra forma, el parser completaría esas estructuras que deben ser finalmente desechadas, con la consiguiente pérdida de tiempo. Por tanto, las restricciones seleccionales pueden incrementar la eficiencia del sistema al tiempo que mejoran la precisión de los análisis.

4.4.3. Discurso

Una de las características más sobresalientes del lenguaje humano es que hay muchas cosas que no se dicen de forma explícita, sino que van implícitas en el discurso. Considérese el siguiente ejemplo:

He ganado bastante con la venta de unas acciones. De momento, voy a dejarlo en el banco.

Cualquier hablante es capaz de entender que en la segunda oración, el banco no se refiere a un objeto para sentarse, sino a una entidad de crédito y que el pronombre clítico "lo" se refiere al dinero ganado con la venta de las acciones. Para llegar a estas conclusiones, los hablantes tienen que establecer la conexión entre la primera y la segunda oración, es decir, hacen uso del contexto. Además, son capaces de inferir que el pronombre hace referencia a un "dinero" que no se ha mencionado explícitamente, es decir, que está elidido.

Estas características, uso del contexto discursivo, resolución de las referencias y alusión a elementos elididos, hacen que las lenguas naturales sean extremadamente complicadas de entender para un programa de ordenador. Sin embargo, un sistema PLN que intente emular la capacidad lingüística de los hablantes tiene que disponer de recursos para hacer frente a estos problemas. Se verán, por ejemplo, las estrategias más frecuentes para resolver las referencias anafóricas.

Considérese la siguiente oración:

Juan se compró tres camisas, pero devolvió una porque le estaba grande.

La entidad a la que se refiere el pronombre (o el sintagma) anafórico es lo que se conoce por *antecedente*. En el ejemplo, el pronombre "una" se refiere a una de las tres camisas que Juan compró. La resolución de la anáfora consiste en localizar dicho antecedente para posteriormente colocarlo en el lugar del material anafórico que se refiere a él y así completar el contenido de la oración. Al intentar llevar este proceso a la práctica hay que plantearse una serie de cuestiones.

La primera de ellas sería ¿cuándo hacer la resolución de la anáfora? Si partimos de la forma superficial de la oración, el número de un sintagma (singular o plural) depende del alcance de los cuantificadores y no siempre coincide con la definición de esa entidad en la forma lógica. Además, en ocasiones el referente no se corresponde con ningún constituyente del árbol sintáctico. Por ejemplo, en *Juan compró varias camisas. Éstas costaron 8.000 ptas.* Si sustituimos *éstas* por su antecedente obtendremos *Varias camisas costaron 8.000 ptas.*, lo cual no es del todo exacto. Su interpretación correcta sería "las camisas que Juan compró costaron 8.000 ptas.". Por tanto, es mejor partir de la forma lógica, en la que a cada sintagma nominal se le ha asignado ya su correspondiente descripción lógica que no presenta ambigüedad. De esta forma, a medida que se procesan las oraciones, se va elaborando una lista en la que figuran las *entidades del discurso*, que son definiciones del conjunto asociado con cada sintagma nominal aparecido en el texto. Los posibles antecedentes se seleccionan entre esa lista de entidades. Típicamente, los sintagmas nominales indefinidos como *un coche* introducen entidades nuevas en el discurso, mientras que los sintagmas definidos como *el coche* (incluidos los pronombres) se refieren a entidades que han aparecido anteriormente. A cada entidad generada se le asigna un nombre, una constante a la que se puede hacer referencia en el proceso posterior. La generación de entidades del discurso debe tener en cuenta si el sintagma se encuentra dentro del alcance de un cuantificador universal con interpretación distributiva. En ese caso, aunque se trate de un sintagma en singular, hay que considerarlo como un conjunto. Por ejemplo (Grishman, 1986):

Cada estudiante compró un libro.

Cuya forma lógica es:

$(\forall\ e \in$ *estudiantes*$)\ (\exists\ l \in$ *libros*$)$ comprar (e, l)

La entidad de discurso que se refiere sólo al subconjunto de libros que compraron los estudiantes sería:

$\{l \in$ *libros* $\mid \exists\, e \in$ *estudiantes*) comprar $(e,l)\}$

Una vez establecidas las entidades del discurso que son candidatas a referentes, hay que elegir entre ellas, par lo cual hay que contar con las restricciones sintácticas y semánticas. Algunas son bastante obvias, como la coincidencia de género y número, pero además de éstas, los teóricos de la semántica han estudiado en qué posiciones sintácticas puede un SN ser el referente de otro y en cuáles no (precedencia y mando-c). Si la sintaxis no es suficiente, habrá que recurrir a la semántica (clases de palabras).

Las siguientes cuestiones son dónde hay que buscar el referente, cuántas oraciones hay que remontarse y qué orden se debe seguir. De acuerdo con Hobbs (1976), hay que buscar por este orden:

1. En la misma oración, dentro de las posiciones sintácticamente posibles.
2. En la oración precedente. El árbol se recorre primero en extensión, de abajo a arriba y de izquierda a derecha. Es decir, el primer SN que se suele considerar con este sistema es el sujeto como antecedente más probable. Si no se encontrara el referente, se pasaría a la oración anterior.

Además de la anáfora otras cuestiones relevantes en el tratamiento lógico del discurso son la elipsis, el tiempo y el aspecto. Su naturaleza compleja no aconseja una presentación simplificadora. Por lo tanto, hay que limitarse a señalar que algunas teorías semánticas formales recientes han dedicado gran atención a estos temas. En concreto, la Teoría de las Representaciones Discursivas (DRT es su acrónimo en inglés), desarrollada por Hans Kamp y cuyo libro de referencia es Kamp y Reyle (1993), y también la Semántica de Situaciones (Barwise y Perry, 1983; Devlin, 1991). Ambas suponen un intento de ampliar las posibilidades de la lógica de primer orden y de superar la semántica basada en la oración, insistiendo en el hecho de que la interpretación de las oraciones aisladamente depende mucho más del discurso de lo que habitualmente se ha defendido.

4.4.4. Conocimiento del mundo

Para entender una lengua los hablantes cuentan, además de su competencia lingüística, con conocimiento general acerca del mundo. Este conocimiento les permite inferir el nexo causa-efecto que une las oraciones: *El árbitro tuvo una mala noche. El presidente del equipo casero ha pedido la anulación del encuentro.*

La *coherencia discursiva* es uno de los conceptos clave: un discurso es coherente cuando se puede determinar fácilmente qué relación existe entre unas oraciones y otras. Los teóricos de la comunicación han desarrollado distintos conceptos como las *implicaturas*, la *relevancia*, las *expectativas*, las *máximas de la conversación*, que tratan precisamente de definir las características de la coherencia discursiva. Dentro de la LC, estos conceptos se han aplicado sobre todo en la generación de oraciones, para planificar la producción de un discurso inteligible.

A) Modelos, marcos, guiones

Una forma bastante habitual de organizar el conocimiento del mundo en sistema PLN es elaborar unos modelos que especifican los elementos que intervienen en ciertas situaciones y las relaciones entre ellos. Este sistema fue iniciado por Minsky (1975), quien llamó marcos *(frames)* a estos patrones.

La unidad básica de conocimiento es el marco. Estos marcos están clasificados en:

— *Marcos prototípicos:* son los que describen clases de objetos o situaciones. Un ejemplo de este tipo de marco es el que describe un tipo de habitación como el dormitorio, la cocina o el baño, o bien una situación como hacer la compra, adquirir unas acciones o cometer un atentado. Se trata, por tanto, de los marcos más generales, aunque dentro de ellos también se pueden jerarquizar. Así, el modelo "habitación" es más general que el modelo "cocina". Normalmente, existen mecanismos de herencia, ya que los modelos más concretos heredan ciertas propiedades de sus padres, que no es necesario volver a especificar.
— *Marcos de aparición:* son los que describen objetos o situaciones individuales, es decir, son un caso particular del marco prototípico. Siguiendo los ejemplos anteriores, un marco de aparición describiría la cocina del mago Merlín o el atentado contra Kennedy.

Cada marco o modelo tiene un conjunto de posiciones etiquetadas que especifican las propiedades del objeto o de los participantes de la situación que se representa. Las posiciones pueden estar ocupadas por un número, una secuencia de caracteres o pueden hacer referencia a otro marco.

El modelo semántico que se muestra como ejemplo fue utilizado en el sistema de extracción de información Proteus (Olmeda y Moreno, 1992), cuyo dominio eran los atentados terroristas en Sudamérica. En concreto, se trata

de una descripción del verbo "saldar (se)" que, partiendo del análisis de la oración, proyecta las funciones sintácticas en papeles semánticos. El modelo consta de un identificador (:id); un modelo "padre" (:parent), del que se heredan algunas características; la especificación de la palabra o clase de palabras a las que se aplica (:constraint); y el modelo en sí mismo, que está formado por una lista de adjuntos. Cada adjunto se identifica por su función sintáctica (:marker) y por la clase semántica dentro de una jerarquía de conceptos (:class) y el modelo le asigna un papel dentro de la estructura semántica (:case).

```
(add-clause-model :id 'clause-saldar-se
                  :parent 'clause-any
                  :constraint 'saldar
                  :ADJUNCTS (list (make-specifier
                                     :marker 'subject
                                     :class 'suceso
                                     :case 'cause)
                                  (make-specifier
                                     :marker 'se-marker
                                     :class 'se
                                     :case 'ignore)
                                  (make-specifier
                                     :marker 'con
                                     :class 'suceso
                                     :case :effect
                                :essential-required 'required)))
```

Al aplicar este modelo se puede establecer que el sujeto de la oración *El atentado de Sendero Luminoso se saldó con cinco muertos* no es en realidad un actor, sino una causa que ha producido un efecto. Como se observa en el modelo, el papel asignado al objeto preposicional introducido por "con" remite a su vez al modelo que describe un "efecto" (:effect). Además, el modelo toma en cuenta que tanto el sujeto como el objeto deben pertenecer a la categoría semántica "suceso", que se les habrá asignado haciendo uso de una jerarquía conceptual como se verá enseguida.

Estos marcos permiten, por tanto, interpretar de una forma bastante precisa los mensajes. El problema es que sólo pueden aplicarse a dominios muy restringidos. Para analizar textos generales habría que elaborar tal cantidad de marcos de todo tipo que la tarea se vuelve impracticable.

Otra variante de estos modelos, que sirve sobre todo para analizar textos de tipo narrativo, son los *guiones* (Schank y Abelson, 1975, 1977). Estos guiones dan cuenta del conocimiento de las personas sobre secuencias de sucesos esteoreotipadas, como puede ser una visita al médico. En el guión se codificará lo que suele ocurrir en una de estas visitas (espera en la sala, llamada, saludos, preguntas típicas, examen clínico, diagnóstico, receta...), guiando de esta forma la interpretación de las distintas oraciones que aparezcan en la narración.

B) Redes semánticas y jerarquías conceptuales

Una idea bastante extendida, tanto en Psicología como en Inteligencia Artificial, es que en la mente humana los conceptos se encuentran relacionados entre sí formando una red. Utilizando esta imagen, cada concepto constituiría un nodo de la red que se conecta con otros nodos mediante nexos de distinta naturaleza. Los nexos establecen el tipo de relación: algunos de los más frecuentemente empleados son el nexo que indica pertenencia a una clase ("es un tipo de"), el de meronimia ("es una parte de"), el nexo de sinonimia ("es igual que"), el de función, o el de co-localización ("está en"). El conjunto de nodos y nexos se conoce como *red semántica*. Las redes semánticas han sido empleadas por numerosos sistemas PLN para codificar y almacenar el conocimiento del mundo empleado en la interpretación. La figura 4.6 muestra un ejemplo simple de red semántica, donde los conceptos están en cursiva y las relaciones en redonda.

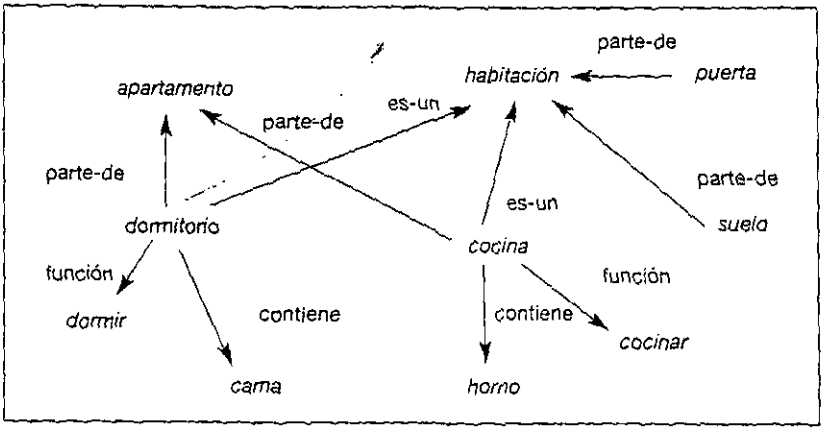


Figura 4.6. Ejemplo de red semántica.

Las redes semánticas más sencillas conectan las palabras en sí mismas, pero lo más frecuente es utilizar una red semántica más abstracta, que utiliza conceptos organizados jerárquicamente y en la que sólo los elementos terminales se corresponden con palabras de la lengua en cuestión. En principio, esto permite utilizar la misma jerarquía de conceptos para varias lenguas teniendo que modificar sólo esos elementos. Los conceptos que se encuentran en lo más alto de la jerarquía son muy abstractos como, por ejemplo, *suceso* o *entidad*. A medida que se desciende en la jerarquía, los conceptos son más concretos y se pueden introducir distinciones tan finas como se considere necesario para el buen funcionamiento del sistema. La principal relación en estas jerarquías es la de hiperónimo ("ser un" o "pertenecer a la clase de"), aunque también se utiliza la sinonimia y la meronimia. La característica que distingue estas jerarquías de una red simple es que utilizan la herencia de propiedades de las categorías superiores a las inferiores. De este modo, cualquier modificación que se realice en un nodo superior, se extiende a todas sus instancias. El ejemplo más conocido hasta la fecha de relaciones semánticas entre palabras es WordNet, proyecto desarrollado en Princenton originariamente para el inglés y que actualmente se está extendiendo a varias lenguas europeas (EuroWordNet).

Las jerarquías semánticas se pueden utilizar con varios fines en un sistema PLN. Por un lado, se pueden emplear para guiar el proceso de análisis sintáctico. La asignación de una clase semántica a los distintos sintagmas presentes en la oración permite descartar análisis erróneos o, al menos, establecer preferencias entre las distintas posibilidades. Por ejemplo, si tenemos un verbo que en la jerarquía conceptual está clasificado dentro de los verbos de movimiento, podemos inferir que su sujeto estará clasificado dentro del grupo de seres animados.

Por otro lado, tanto las redes como las jerarquías se utilizan en el proceso de interpretación. Algunos sistemas establecen directamente la relación semántica entre los distintos elementos haciendo uso de los correspondientes nexos de la red. En otros sistemas, se emplea la asignación a una clase conceptual para luego poder trabajar con modelos semánticos como acabamos de ver con los marcos.

La dificultad que plantean estos modelos es la delimitación de los distintos conceptos o nodos que intervienen en la red. Aunque hay bastante acuerdo en que la mente trabaja con conceptualizaciones, todavía estamos muy lejos de poder establecer cuáles son los conceptos básicos y de asignarles un contenido fijo. A diferencia de la sintaxis, que trabaja con unas categorías sobre las que existe un acuerdo casi unánime (nombre, verbo, SN, SV), no hay por el momento un inventario de conceptos o de primitivos semánticos universalmente aceptado, lo que hace que cada investigador y cada sistema utilice el suyo, aunque haya muchas coincidencias entre ellos.

4.5. Lexicones computacionales

Cualquier sistema PLN necesita un lexicón rico en información sobre propiedades morfológicas, sintácticas y semánticas de las palabras de una lengua. Como en el caso de las gramáticas computacionales, una descripción previa del léxico es una ayuda esencial para el lexicógrafo computacional. Desgraciadamente los diccionarios impresos no son fáciles de trasladar a un entorno computacional, y ni siquiera los que han sido adaptados a una versión electrónica son utilizables directamente. La razón es que los diccionarios tradicionales no están lo suficientemente formalizados ni estructurados. Chuchuy y Moreno (en prensa) analizan las características y las deficiencias estructurales de cuatro diccionarios de español en formato CD-ROM.

4.5.1. Estructura de la información

Los diccionarios computacionales se caracterizan por una clara división de los tipos de información. Además su formato habitual son estructuras de rasgos, o estructuras de datos de lenguajes de programación. Para sistemas PLN sencillos lo normal es desarrollar el lexicón desde el principio e ir aumentándolo progresivamente con nuevas entradas añadidas a mano. Éste suele ser suficiente para sistemas que tratan dominios acotados, pues con lexicones de menos de 1.000 lemas pueden funcionar perfectamente. El problema surge cuando se quiere manejar texto no restringido temáticamente. Entonces se requieren diccionarios de varias decenas de miles de lemas. Esta necesidad se ha acentuado especialmente en la década de los noventa y ha inducido a buscar técnicas y herramientas que faciliten la adquisición de información léxica a partir de recursos ya existentes, como los diccionarios en formato electrónico, y también a partir de corpus textuales. Por ejemplo, una forma de adquirir léxico es revisar un texto para comprobar si hay palabras que no están codificadas en nuestro diccionario o ampliar las existentes con nueva información (por ejemplo, distintas subcategorizaciones o diferentes sentidos). Otra fuente de adquisición han sido las bases de datos terminológicas, a menudo de carácter multilingüe, que contienen información acotada semánticamente y muy bien estructurada. Por último, también se (re)utilizan teorías y descripciones sobre el conocimiento del mundo (por ejemplo, los tesauros y otros tipos de clasificaciones). Todos estos recursos combinados han permitido compilar diccionarios computacionales de gran número de entradas en pocos años.

Repasaremos muy rápidamente las cuestiones esbozadas. Los diccionarios tradicionales son una fuente valiosa de información para cualquier lexicógrafo computacional. El problema fundamental de su reutilización consis-

te en cómo traducir una información expresada generalmente de manera no formalizada (las clásicas definiciones en lengua natural) en información formalizada y estructurada.

Boguraev y Levin (1993) repasan algunos de los problemas de la extracción automática de información léxica de recursos en formato electrónico, llegando a la conclusión de que hay mucha información implícita en los diccionarios tradicionales que es muy difícil de descubrir automáticamente. En concreto, la información semántica acerca de diferentes sentidos de una palabra.

El volumen editado por Boguraev y Briscoe (1989) recoge una serie de artículos sobre el desarrollo de lexicones computacionales a partir de las cintas magnéticas del LDOCE (Longman Dictionary of Contemporary English).

4.5.2. Acceso a la información léxica

Otro aspecto esencial en un lexicón computacional, además de contener una gran cantidad y variedad de información, es la manera en que se accede a dicha información. El acceso a las entradas léxicas se hace a través de etiquetas o *claves* que están asociadas a cada entrada. Esto es equivalente a los lemas que encabezan las entradas de los diccionarios tradicionales. Por tanto, lo primero que hace cualquier sistema es buscar por las claves para recuperar luego toda la información asociada. Los métodos de búsqueda léxica eficiente son múltiples y no es el propósito de un libro como éste detenerse en su exposición. Un diccionario computacional no es más que un subtipo de estructura de datos y las técnicas informáticas que se aplican son similares a las que se utilizan en otros campos. A título orientativo, se mencionan algunas de las más empleadas:

1. Tipos de estructuras de datos:
   - *Listas lineales*: es la forma más elemental de estructurar información, pues simplemente coloca los ítems uno detrás de otro y su principio organizativo es el orden en el que se han ido añadiendo las entradas. Por ejemplo, en las pequeñas gramáticas que hemos expuesto en secciones anteriores el léxico no estaba organizado por ningún principio, salvo el de ir añadiendo palabras para comprobar el funcionamiento de las reglas. Este tipo de estrategia sólo es válida para gramáticas de prueba.
   - *Estructuras jerarquizadas:* consiste en organizar la información en distintos niveles. El ejemplo más clásico es el *árbol*, formado por un nodo raíz del cual dependen varios nodos subsidiarios, con sus relaciones de dominio y precedencia. El número de ramas del

árbol, conocido como *factor de ramificación*, es una medida sobre la complejidad de la búsqueda y el tiempo que consumirá. El *árbol binario* es una de las variantes preferidas. Como su nombre indica cada nodo tiene como máximo dos ramificaciones, lo que reduce la elección al mínimo y mejora la eficiencia de búsqueda. Sin embargo, muchas veces es indeseable o imposible presentar únicamente dos opciones.

— *Redes:* es un tipo de estructura donde los nodos están unidos bien directamente o bien indirectamente por medio de nodos intermedios. Un aspecto interesante es que entre dos nodos puede haber varios caminos posibles. Por lo tanto, las redes representan piezas de información que pueden no estar unidas por un único patrón (como ocurre con los árboles) sino por varios. Éste es un caso muy habitual en la información semántica, como se vio al tratar las redes semánticas.

2. Tipos de técnicas de búsqueda léxica:

— *Búsqueda binaria en una lista alfabética:* es el método más antiguo pero sigue siendo muy eficiente. El algoritmo lo describió Mauchely en los primeros tiempos de los ordenadores. Consiste en encontrar la palabra que aparece justo en la mitad de la lista, comprobar si la que estamos buscando está por encima o por debajo de ella y repetir la operación de escoger la palabra en el medio hasta que encontremos la que estamos buscando. En cada paso, vamos reduciendo a la mitad el tamaño de la lista. Este método es más eficiente cuanto más grande sea la lista donde se busca. Smith (1991) calcula que para una lista de 47 palabras hacen falta 6 pasos; para un diccionario de 60.000 lemas se necesitarán 17 pasos; y para encontrar a una persona en un censo de 6.000 millones de habitantes se requerirían solamente 35 comparaciones. La limitación de la búsqueda binaria es que hay que mantener ordenada alfabéticamente la lista. Cada añadido o borrado supone un nuevo reordenamiento de la lista. Si los cambios son muy frecuentes y la lista muy grande, esta técnica puede no ser la más eficiente.

— *Árboles de búsqueda:* esta estrategia se aplica lógicamente a estructuras de datos organizadas en forma de árbol. En el árbol binario en cada nodo hay una palabra y de él salen como máximo dos punteros, uno a la palabra precedente y otro a la siguiente. El método es muy parecido al de búsqueda binaria en listas: en cada paso se compara la palabra que se busca con la del nodo vigente; si no coinciden entonces se toma el camino de la derecha o el de la izquierda en función de si la palabra que se busca precede o sigue alfa-

béticamente a la del nodo vigente. La ventaja de esta técnica reside en la organización del diccionario, ya que añadir nuevas palabras al árbol consiste en encontrar el nodo madre apropiado y ajustar los punteros, de manera que sólo se modifica una pequeña porción de la estructura y el resto del árbol se mantiene. Con frecuencia los árboles se convierten en asimétricos (es decir, hay más ramas en unas partes que en otras) cuando se introducen muchos cambios. En ese punto, el método pierde su eficacia. Para resolver este problema, se utilizan los árboles equilibrados *(B tree*, en inglés, por *balanced tree)*. Se trata de un árbol no binario o de múltiples ramas. Es decir, en lugar de dos punteros se almacenan más claves. Para mantener la eficiencia se restringe el número de claves y factor de ramificación de cada nodo. Cuando el límite se excede, automáticamente se reorganiza el árbol de manera que nunca pierda su equilibrio. Otra técnica de árboles de búsqueda muy empleada es la de los *árboles binarios con pesos*. Está basada en cálculos estadísticos sobre la frecuencia de aparición de las palabras. Para reducir al mínimo el número de comparaciones necesario para encontrar una palabra de uso frecuente se crean subárboles con distintos pesos en sus transiciones. Naturalmente, para crear esta estructura es necesario conocer las estadísticas de las palabras. Por último y probablemente uno de los más utilizados últimamente, tenemos a los *trie*, que son árboles cuyos nodos están formados por caracteres en lugar de por palabras. El ejemplo de árbol de letras mostrado al explicar la aplicación de autómatas a la morfología es un trie. La búsqueda es muy eficiente, pues sólo compara caracteres en lugar de palabras completas, aunque la eficiencia se degrada cuando el factor de ramificación aumenta.

— *Hashing*: esta técnica de búsqueda utiliza una tabla en lugar de listas o árboles. Cada elemento léxico tiene una clave numérica en la tabla. Dicha clave se calcula mediante una función que también se utiliza para buscar la palabra. Por ejemplo, supongamos que la función asigna el valor 35 a la palabra *helicóptero*, 4 a *aeropuerto*, 21 a *aeroplano*, etc. La función está basada en fórmulas que utilizan distintas variables, como por ejemplo valor de la primera letra y número total de letras. La idea es la misma que la de la función que asigna la letra al final de los dígitos en el NIF. Al buscar una determinada palabra, se aplica la función y da el valor único, que identifica directamente al elemento en la lista. La búsqueda es rapidísima porque ni siquiera tiene que hacer comprobaciones como en el caso de los métodos binarios. Además, tampoco hay que ordenar los elementos. Para que esta técnica funcione, tiene que

haber más posiciones en la tabla que elementos por colocar. El gran problema que tiene esta técnica son las llamadas *colisiones*: cuando dos o más elementos reciben el mismo valor por la función y por tanto la misma dirección en la tabla. Hay distintos algoritmos con funciones de *hashing* que garantizan la mínima colisión para listas de determinado tamaño, pero aun así el problema de la colisión no está resuelto completamente.

4.5.3. Tipos de lexicones computacionales

El lexicón siempre depende de la gramática: las entradas léxicas no son más que los elementos terminales que se insertan en las reglas gramaticales. Los lexicones más sencillos son los de las gramáticas sintagmáticas independientes del contexto. A modo de recuerdo, se copia parte del diccionario de una de las gramáticas expuestas en este capítulo:

```
DET: (el | la | los| las | un | una | unos |unas)
PRON: (yo | tú | él | ella | nosotros | vosotros | ellos)
N: (Cyrano | Calisto | Melibea | corazón | nariz | amor | mentira | poema | reflejos | hombre )
```

Naturalmente, la simplicidad de estas entradas no es útil para ningún sistema. La única información que se aporta es la cadena de caracteres y su categoría sintáctica. Las entradas léxicas de los modelos morfológicos que presentamos en la sección 3.4.2. son más informativas, pues contienen información morfosintáctica. Por otra parte, la unidad básica no es la palabra, sino el morfema:

Forma léxica	*Clase de continuación*	*Glosa*
pez	PLURAL	"N(pez)"
+s	FRONTERA	"+PLURAL"

Un diccionario para una gramática de cierta complejidad debe incluir información sintáctica y semántica. Serían las entradas para *alcalde* y *comprar* en el lexicón del proyecto PROTEUS:

```
(noun :masc-sing    ALCALDE    :attributes (HUMAN))
(verb :baseform COMPRAR
      :attributes (reflexive)
      :objlist (DIRECT-OBJ DITRANS DIROBJ-PN (PVAL (PARA))))
```

En estos ejemplos podemos ver información morfológica (:masc-sing y :baseform son dos etiquetas para compactar información léxica, de la que se hablará enseguida); información sintáctica (:reflexive y :objlist; el último aporta la lista de objetos subcategorizados por el verbo) e información semántica (:human, que restringe la regla de objetos directos con la preposición *a*).

En la última década se ha extendido la utilización de formalismos gramaticales basados en la unificación y eso ha supuesto la generalización del uso de estructuras de rasgos y de la herencia de información en las entradas del lexicón. Por ejemplo, la entrada para *pez* en el sistema GRAMPAL:

pez	pec
morfo-cat = raiz-n	morfo-cat = raiz-n
sint-cat = n	sint-cat = n
lex = pez	lex = pez
conc gen = masc	conc gen = masc
tipo-plu = no	tipo-plu = plu2
tipo-gen = inherente	tipo-gen = inherente

Efectivamente, las estructuras de rasgos proporcionan una estructura de datos ideal para codificar información léxica compleja (Sanfilippo, 1996). Además, el mismo formalismo permite especificar morfemas, palabras y sintagmas, lo que supone una considerable ventaja sobre modelos previos de almacenamiento léxico. Se reproducen de nuevo las entradas utilizadas para ejemplificar la subcategorización:

palabra amar:	palabra dar:
<cat> = V	<cat> = V
<arg0 cat> = SN	<arg0 cat> = SN
<arg0 función> = sujeto	<arg0 función> = sujeto
<arg1 cat> = SN	<arg1 cat> = SN
<arg1 función> = obj-dir	<arg1 función> = obj-dir
	<arg2 cat> = SP
	<arg2 valor-p> = a
	<arg2 función> = obj-indir

Por su parte, el concepto de *herencia* permite ordenar las estructuras de rasgos en una jerarquía de manera que no sea necesario especificar nada más que una vez la información que se repite en subconjuntos de entradas.

Para ello se utilizan *plantillas* o *macros léxicas*. Todos aquellos rasgos que presenten subclases de verbos transitivos como *amar*, o ditransitivos como *dar*, pueden ser definidos mediante macros como las siguientes:

Macro verbos-trans:	Macro verbos-ditrans:
<cat> = V	verbos-trans
<arg0 cat> = SN	<arg2 cat> = SP
<arg0 función> = sujeto	<arg2 valor-p> = a
<arg1 cat> = SN	<arg2 función> = obj-indir
<arg1 función> = obj-dir	

Las macros tienen un símbolo que las identifica y que abrevia el conjunto de especificaciones de rasgos. En la macro verbos-ditrans observamos que se incluye la macro verbos-trans. Esto ha de interpretarse como que los rasgos asociados a esa macro se deben incluir en la primera macro. De esta forma se consiguen entradas léxicas muy compactas y sin redundancia:

lexema amar: verbos-trans.	lexema leer: verbos-trans.	lexema dar: verbos-ditrans.	lexema ofrecer: verbos-ditrans.

Por consiguiente, la información se hereda de los niveles superiores, y esto ha "transformado" el trabajo del lexicógrafo computacional. La manera habitual es proporcionar la máxima especificación en cada entrada particular. La utilización del concepto de herencia obliga a concentrarse en el establecimiento de clases o tipos que comparten información, y en organizar jerarquías de herencia. Una vez realizada esa labor, la ampliación del diccionario con nuevas entradas se reduce a clasificar cada unidad léxica en su clase apropiada.

La unificación es el mecanismo que se utiliza para integrar la información inherente y heredada a través de macros en las estructuras de rasgos que se utilizan en procesamiento (Sanfilippo, 1996). Una característica habitual de la unificación es su *monotonía* o *monotonicidad*: toda información, ya sea especificada o heredada, se conserva. Este requisito se impone para que una entrada válida no contenga rasgos conflictivos. Como ya se dijo, la unificación de información incompatible no se puede producir. Por lo tanto, el carácter monótono de la unificación clásica es una herramienta para comprobar la consistencia de las entradas léxicas: si por equivocación o error se han asignado valores incompatibles a la misma entrada por medio de la heren-

cia y de la asignación directa, la unificación detectará la inconsistencia y permitirá al lexicógrafo computacional corregir el error, bien en la jerarquía de herencia, bien en la entrada concreta.

El desarrollo de la teoría de las estructuras de rasgos tipificadas (Carpenter, 1992) ha permitido incluir comprobaciones de consistencia mucho más estrictas. En concreto, la tipificación especifica exactamente los atributos y los valores apropiados para una determinada estructura de rasgos.

Al mismo tiempo que se desarrollaban estas técnicas, ha surgido una aproximación no monótona a la unificación. Esta versión está inspirada en el trabajo en representación del conocimiento. La *no monotonía* es el uso de reglas por defecto que pueden tener excepciones. Dicho de otra manera, supongamos que asignamos los rasgos A, B, C y D a todos los verbos de tipo X, pero resulta que hay un verbo que, cumpliendo los otros requisitos, no tiene el rasgo A. ¿Qué hacemos? La solución clásica es crear una definición específica para él, pero esto no es satisfactorio porque se pierde la generalización de que el verbo es del tipo X en todo menos en un rasgo. La lógica clásica nos dice que la excepción invalida la regla, justo lo contrario que el sentido común: la excepción confirma la regla. Precisamente con la lógica no monótona lo que se pretende capturar es la inferencia de sentido común, mediante la relajación de la condición de monotonicidad, de manera que se puedan recoger regularidades a pesar de las excepciones.

La aplicación de la herencia por defecto a fenómenos lingüísticos que ha tenido más éxito hasta la fecha ha sido la que se da en la morfología. Es el momento de recoger el hilo de la discusión que se dejó en la sección 3.4.2.

Todas las lenguas tienen verbos irregulares, por supuesto, unas en mayor cantidad que otras. Una observación importante es que la mayoría de los llamados verbos irregulares se comportan básicamente como verbos regulares salvo en que se desvían en unos pocos rasgos. Por eso Gazdar y Evans (1996) prefieren hablar de subregularidades, en lugar de irregularidades. Estos dos autores han desarrollado el lenguaje de representación léxica DATR, que incorpora como característica definitoria los conceptos de *herencia por defecto* y *sobreescritura*.

El método es como sigue: en primer lugar, establecemos una clasificación jerárquica de las clases léxicas (por ejemplo, el verbo con sus distintos paradigmas regulares e irregulares). Esta clasificación se organiza en nodos, y cada uno reúne una colección de información compartida por los miembros de la clase. Esta información se representa por medio de rasgos. En el contexto de la descripción léxica, los nodos de la red pueden ser, de menor a mayor, una forma flexionada, un lexema o un tipo de lexemas. Los nodos en posiciones inferiores en la jerarquía heredan los rasgos por defecto de los nodos superiores, excepto cuando el nodo inferior ya tiene asignado un valor para un rasgo concreto. En este caso, el valor por defecto *se sobreescribe*.

Veamos un ejemplo clásico, adaptado de Fraser y Corbett (1995): en una clasificación del concepto de AVE, podemos especificar como rasgos definitorios "plumas = sí" y "puede volar = sí". Por lo tanto, cualquier concepto que dependa de este nodo heredará estos rasgos. Así, si definimos GAVIOTA y GALLINA podemos establecer que heredan las características de AVE, además de especificar alguna propiedad que las diferencie de otras unidades, por ejemplo, "hábitat marino = sí" para GAVIOTA y "ave doméstica = sí" para GALLINA. Ninguno de estos rasgos contradice la información heredada de AVE. Supongamos que queremos incluir el concepto PINGÜINO. Sabemos que es un AVE, que tiene plumas, pero *no puede volar*. Esta información, siendo cierta, es incompatible con nuestra definición de AVE. En un modelo monótono habría que eliminar el rasgo que presenta excepciones, "puede volar = sí", con lo cual nuestra descripción pierde una característica muy importante de las aves. Sin embargo, en un modelo no monótono esta información puede mantenerse, y sólo hay que especificar en la entrada de PINGÜINO "puede volar = no" (figura 4.7).

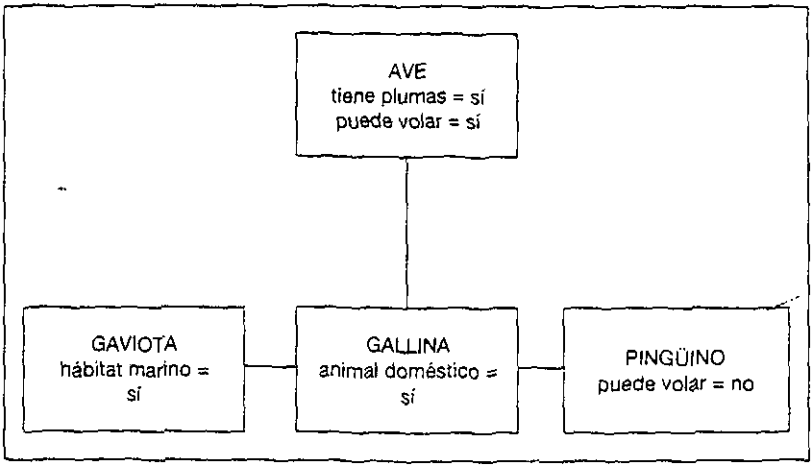


Figura 4.7. Jerarquía para el concepto de AVE.

Esta manera de organizar la información léxica presenta tres característica interesantes:

1. Permite expresar generalizaciones una sola vez, evitando la redundancia.
2. Se emplea un procedimiento simple y general tanto para las regularidades como para las irregularidades.
3. Las excepciones se marcan como tales, pero incluidas dentro del modelo, no como una nota a pie de página.

Las jerarquías de herencia se aplican especialmente bien a sistemas donde aparecen bastantes excepciones, en concreto el léxico y la morfología. Corbett y Fraser han desarrollado la *Network Morphology* inspirándose en este modelo (Corbett y Fraser, 1993; Fraser y Corbett, 1995). Moreno Sandoval (en prensa) presenta una aplicación a la morfología del verbo español, que se resumen a continuación como ejemplo de las posibilidades de este método.

La conjugación verbal del español está compuesta por 55 formas (se descartan las formas obsoletas del futuro imperfecto del subjuntivo). A pesar de parecer un número ciertamente elevado con respecto a otros sistemas morfológicos, lo cierto es que hay varias *homonimias paradigmáticas*: formas diferentes utilizan el mismo morfema flexivo. Por ejemplo, la primera y la tercera personas del singular del presente de subjuntivo o, más significativamente, los verbos de la segunda y tercera conjugaciones, comparten las mismas desinencias para los tiempos pretéritos del indicativo y presente y pretérito imperfecto del subjuntivo. Claramente, hay mucha redundancia en los paradigmas, lo que probablemente esté motivado por razones psicolingüísticas: facilidad de aprendizaje, facilidad de almacenamiento. Parece además que es un fenómeno universal en las lenguas que tienen una morfología de tipo paradigmático (Carstairs, 1987). Las descripciones tradicionales del verbo español consisten en largas enumeraciones de distintos modelos de irregularidades. Sin embargo, la morfología verbal del español (salvo los verbos muy irregulares como *ser, estar, ir,* etc.) se puede representar en dos jerarquías de herencias por defecto. La primera recoge la estructura de la organización de los sufijos verbales y la segunda la organización de los alomorfos de la raíz. Las figuras 4.8 y 4.9 muestran ambas redes de herencia.

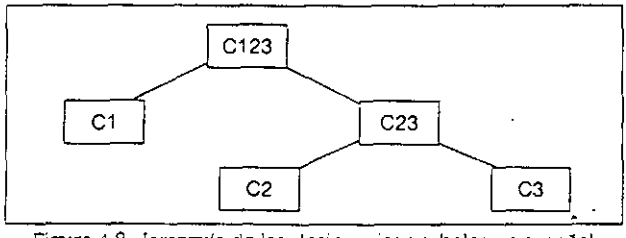


Figura 4.8. Jerarquía de las desinencias verbales en español.

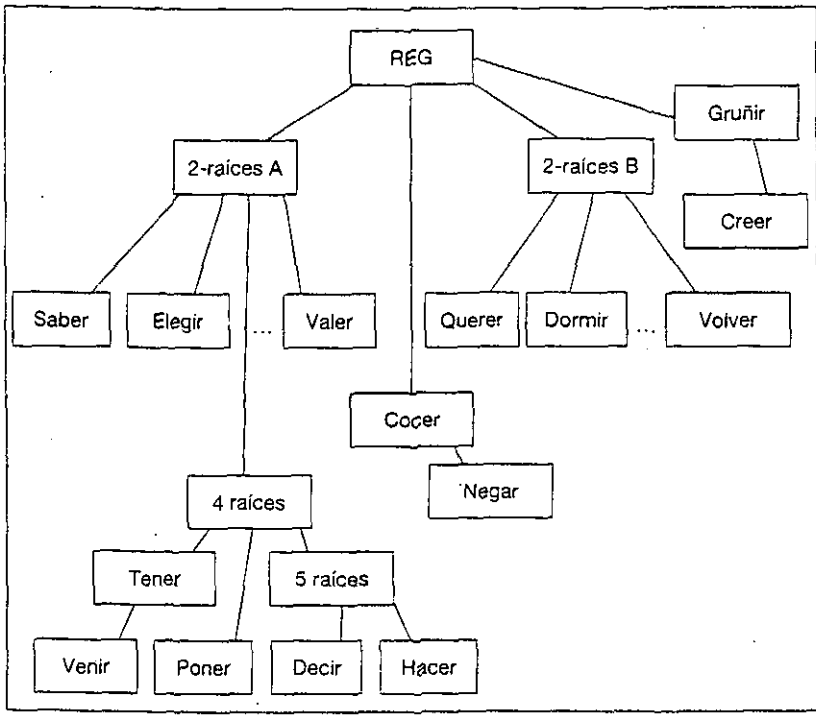


Figura 4.9. Jerarquía de las raíces verbales en español.

La jerarquía de las desinencias es fácil de interpretar: las tres conjugaciones, C1, C2 y C3, están organizadas en una red, donde el nodo C23 reúne todos los sufijos verbales compartidos entre C2 y C3; C123 es el nodo que agrupa la información comparida por todas la conjugaciones. En concreto, el rasgo de categoría sintáctica y la desinencia de primera persona del singular del presente del indicativo. -o *(amo, temo, parto)*.

La jerarquía de raíces presenta la organización jerárquica de los distintos paradigmas irregulares. Frente a la estructura plana de las clasificaciones convencionales, esta jerarquía aporta información muy significativa:

1. Todos los paradigmas irregulares heredan cierta información del paradigma regular (REG).
2. La mayoría de los paradigmas irregulares se organizan en torno a dos clases, que llamamos 2-raíces A y 2-raíces B. Se caracterizan porque

sólo tienen dos alomorfos de la raíz, y cada uno tiene una distribución complementaria de formas
3. Los paradigmas más irregulares (los que tienen cuatro o más raíces) heredan de modelo 2-raíces A.
4. Los paradigmas que contienen el nombre de un verbo representan pequeñas variaciones sobre el paradigma del que heredan.

En resumen, esta organización de la morfología estructurada en torno a los conceptos de paradigma y de herencia por defecto permite dar cuenta de una manera muy económica (en el sentido minimista) de toda la flexión verbal del español en comparación con los modelos basados en reglas que se vieron en el apartado 3.4.2. Para las cuestiones de implementación, consúltese el mencionado artículo o el de Cahill y Gazdar(1997).

Para terminar esta sección, hay que decir que aunque la herencia por defecto en el lexicón es atractiva y deseable para conseguir simplicidad y eliminar la redundancia en las descripciones, lo cierto es que puede ser muy problemática si se usa de una manera generalizada y sin restricciones. Por ejemplo, si se permite la herencia múltiple es muy probable que se produzcan situaciones con información muy conflictiva y contradictoria heredada de distintos nodos. Estos casos son difíciles de resolver y de momento no se ha podido conseguir una formulación satisfactoria de la unificación no monótona (Thomason, 1997) aplicable de manera generalizada a todos los niveles lingüísticos. Piénsese, por ejemplo, que la operación básica de la semántica es la composicionalidad, fenómeno inherentemente monótono; o en las reglas sintácticas y fonológicas, que en general no presentan excepciones.

4.6. Cuestiones prácticas

En las secciones anteriores se han visto problemas particulares de cada nivel lingüístico. En las páginas que siguen se hablará de cuestiones prácticas de carácter general.

4.6.1. Consejos para escribir una gramática

Al enfrentarse a la tarea de escribir una gramática para un sistema PLN, el lingüista computacional se irá encontrando con una serie de problemas prácticos. En los apartados siguientes se enumeran algunos de estos problemas junto con algunos consejos basados en la experiencia.

A) Toda gramática es incompleta

"No hay gramática sin goteras", la famosa frase de Sapir, refleja el hecho de que no es posible conseguir una formulación completa de los fenómenos gramaticales de una lengua. Efectivamente, ningún gramático puede afirmar que su gramática es capaz de proporcionar un análisis correcto de todas las oraciones de una lengua. Esto es una consecuencia del hecho de que las oraciones son infinitas y no podemos conocer con exactitud qué estructuras se pueden dar en la lengua *(incertidumbre sintáctica)*.

La lección práctica que se sigue de esto es que debemos acotar los fenómenos lingüísticos que queremos tratar antes de empezar a desarrollar la gramática. El procedimiento habitual consiste en elaborar una lista de las construcciones que nos proponemos tratar, a la que se suele llamar *especificaciones*.

Otra técnica para delimitar la cobertura de nuestra gramática es partir de un conjunto de oraciones modelo, que pueden ser ejemplos reales tomados del dominio temático de la aplicación.

En cualquiera de los dos casos, la gramática debe ser capaz de procesar como mínimo las estructuras que figuran en las especificaciones o que aparecen en el conjunto de prueba. De esta manera, se guía el desarrollo de la gramática.

B) Inspirarse en los datos

En la última década se ha popularizado en LC la utilización de grandes colecciones de datos que sirven tanto de modelo para construir la gramática como de campo de prueba para evaluar los resultados. Cuando se trabaja en aplicaciones prácticas, es fundamental no perder de vista los datos. A continuación, se verá una serie de diferencias entre el trabajo "orientado por una teoría concreta" y el trabajo "orientado por un corpus":

— Algunas construcciones muy frecuentes en los corpus, como fechas, precios, expresiones de medida y horarias, números de teléfono, direcciones postales, fórmulas, etc., no suelen estar tratadas en ningún manual de gramática porque se consideran poco interesantes desde el punto de vista teórico. De igual forma, oraciones que ocupan muchas páginas en los manuales teóricos aparecen raramente o nunca en los textos reales. En resumen, las descripciones teóricas suelen ser menos útiles que los datos.
— Controlar la recursividad de los constituyentes: las descripciones teóricas normalmente no especifican un número máximo de constituyentes anidados, por ejemplo cuántas oraciones de relativo puede llevar un sin-

tagma nominal. Sin embargo, la actuación marca unas limitaciones. Para mantener la eficacia de una gramática computacional es preferible restringir el número de elementos anidados basándose en datos extraídos de la actuación que dejar la gramática abierta a todas las combinaciones posibles en teoría.

— Nombres propios: trabajando con texto real es muy habitual encontrarse con nombres propios que pueden bloquear el procesamiento si no están incluidos en el lexicón. Para favorecer la robustez del sistema, hay que dotarle de la capacidad de reconocer este tipo de palabras, ya que no parece deseable ni posible introducir todos los nombres propios en el diccionario. La solución más habitual es utilizar un procedimiento que convierte cualquier palabra que no esté en el diccionario en un nombre propio y tratarlo mediante las reglas correspondientes de la gramática.

En resumen, en LC los datos tienen prioridad sobre las teorías.

4.6.2. Problemas generales: la ambigüedad, la cobertura y las excepciones

Los tres problemas han salido en algún momento a lo largo de la exposición. A continuación, haremos un resumen de los aspectos más importantes.

A) La ambigüedad

Muchos lingüistas computacionales opinan que es el problema más grave de los modelos simbólicos. Cualquier sistema PLN tiene que tratar el problema de seleccionar el análisis semántica y pragmáticamente correcto para una oración determinada de entre un número (con frecuencia grande) de análisis sintácticos posibles. Veamos el ejemplo clásico. La oración *Vi a un hombre en el parque con un telescopio* proporciona, al menos, los cuatro análisis estructurales que se muestran en la figura 4.10. Cada uno de ellos tiene una interpretación diferente, en función de su dependencia estructural:

a) Yo he visto a un hombre que se encuentra en un parque (el que se encuentra en el parque es el hombre, y posiblemente el emisor).
b) Yo he visto en el parque donde hay un telescopio a un hombre (el que se encuentra en el parque es el emisor, pero también el hombre).
c) Yo he visto con un telescopio a un hombre que estaba en el parque (el emisor puede o no encontrarse en el parque).
d) Yo me encontraba en el parque mirando con un telescopio y vi a un hombre (este hombre podría estar o no en el mismo parque).

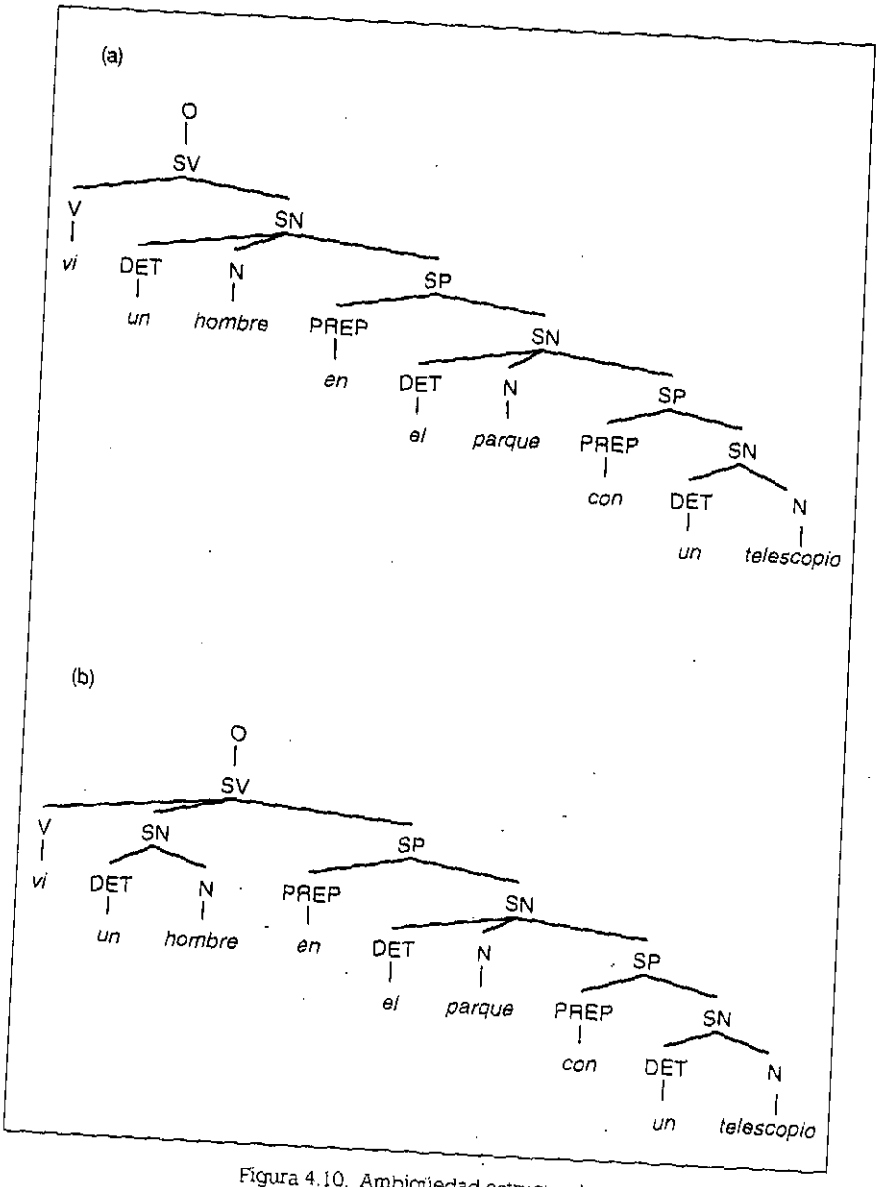


Figura 4.10. Ambigüedad estructural.

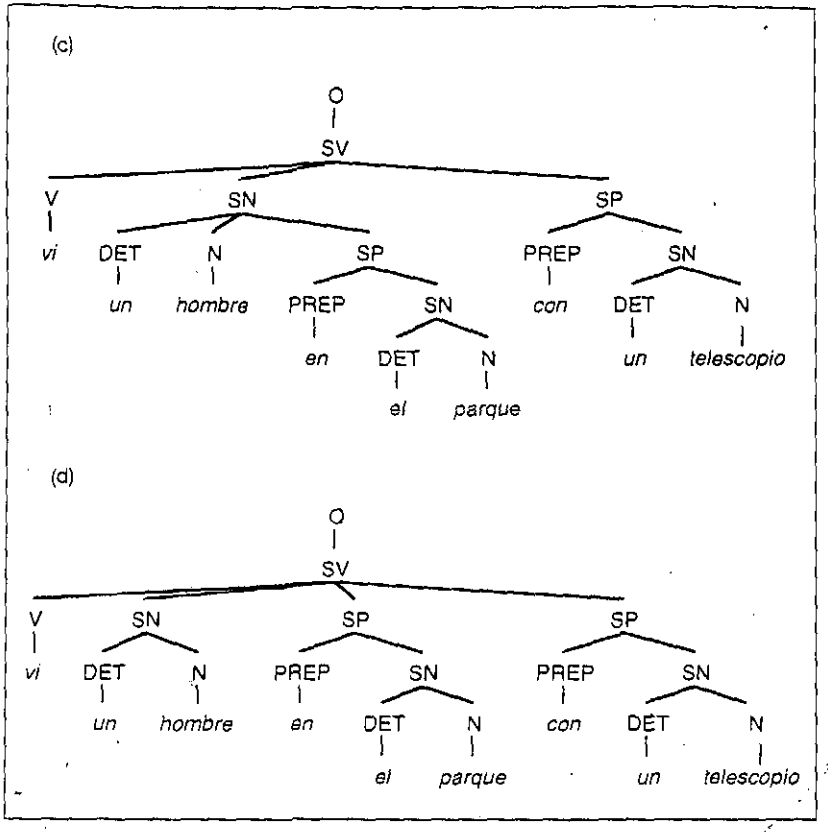


Figura 4.10. (*Continuación*)

Además, si permitiéramos constituyentes discontinuos, podríamos interpretar al menos otro análisis en el cual el hombre tuviera el telescopio. Es decir, *con un telescopio*, al ser el sintagma preposicional más a la derecha, puede asignársele al sujeto (yo), al objeto directo (el hombre) y al adjunto locativo (el parque). *En el parque*, a su vez, puede ser asignado al objeto directo o al sujeto. Algunas interpretaciones, por ejemplo *a)* y *b)* son prácticamente equivalentes, pues es muy posible que el emisor y el hombre se encuentren en el mismo parque. Sin embargo, cada análisis asigna estructuralmente el locativo a un argumento diferente. ¿Cuál de todas ellas es la interpretación más correcta? Sólo lo podemos decidir conociendo el con-

texto donde se ha emitido la oración: ¿quién tiene el telescopio?, ¿el hombre, el parque o el emisor?

La técnica habitual de desambiguación en un modelo simbólico es especificar de manera muy detallada la información semántica. Otra técnica muy empleada es utilizar una estrategia heurística: la experiencia nos dice, por ejemplo, que en los ejemplos observados, la adjunción del SP se da más en el caso del V que en el del SN. Por lo tanto, podemos decidir que en los casos de varios análisis se escoja el que se parezca más al del ejemplo *d)*. Estas soluciones sólo funcionan bien en dominios muy limitados, donde se puede predecir que la asignación del SP tienen a una u a otra configuración estructural, pero en análisis de texto sin restricción temática la asignación correcta es mucho más compleja. Se verá un estrategia de desambiguación con modelos probabilísticos en el apartado 5.5.2.

Aunque no se traten aquí, hay que mencionar otros fenómenos que generalmente son una fuente de ambigüedad estructural: la coordinación, las oraciones relativas y los adjuntos oracionales. En el caso de la coordinación, el problema surge cuando hay constituyentes elididos o son más de dos coordinados. En las oraciones relativas la ambigüedad se produce cuando hay varios candidatos a ser el antecedente. Con los adjuntos oracionales ocurre lo mismo que con los sintagmas preposicionales: pueden ser asignados a distintos núcleos, especialmente cuando la oración es compleja.

B) La cobertura

La falta de cobertura, o *infrageneración*, es el otro gran problema de los modelos simbólicos. La meta de cualquier gramática computacional es generar o analizar la mayor cantidad posible de oraciones gramaticales de una lengua o de un dominio temático. Obviamente, éste es un objetivo ideal que ninguna gramática computacional ha conseguido hasta la fecha. Briscoe (1996) sugiere que parece que hay un límite superior por encima del cual no se puede mejorar la cobertura: un proyecto durante tres años elaboró una gramática para reconocer oraciones del dominio restringido de manuales de ordenador, con un vocabulario también restringido a 3.000 palabras. A pesar de ello, el sistema no fue capaz de analizar el 4% de las oraciones que le presentaron como conjunto de prueba. Aunque el sistema reconoció el 96%, hay que tener en cuenta que a pesar del esfuerzo y de la reducción de la complejidad de la tarea no se consiguió llegar a la cobertura completa. Naturalmente, las cifras de cobertura de la mayor parte de los sistemas menos restringidos temáticamente son mucho peores. Uno de los atractivos de los modelos probabilísticos es que han conseguido elevar el grado de cobertura en algunos casos. Se verá en el siguiente capítulo.

Un último comentario: la infrageneración (o infraanálisis) sólo es realmente un problema para los sistemas de análisis o reconocimiento. Análogamente, la sobregeneración (o sobreanálisis) de una gramática representa un serio problema para los sistemas de síntesis o generación. En situaciones prácticas, a veces es deseable relajar las restricciones y permitir oraciones no estrictamente gramaticales, pero sí aceptables y, sobre todo, usadas por los hablantes.

C) Las excepciones

Como señala Thomason (1997), "cada área de la Lingüística presenta generalizaciones que tienen excepciones". Esto justifica la investigación de métodos que permitan dar cuenta de ellas. Dentro de los modelos simbólicos, las herramientas más apropiadas hasta la fecha son la lógica no monótona y las jerarquías con herencia por defecto. Ya se han visto sus aplicaciones al tratamiento computacional de la morfología y del léxico, niveles lingüísticos donde se han investigado más las propiedades de estas técnicas. El uso de lógica no monótona para dar cuenta de excepciones a reglas sintácticas o fonológicas, sin embargo, no parece tan fácil de adoptar como en los mencionados campos. Igualmente, se ha intentado desarrollar una estrategia de unificación por defecto de estructuras de rasgos, pero hasta la fecha no se ha conseguido ningún sistema que dé resultados satisfactorios. Según Thomason, esto se debe a las complejidades de las definiciones de herencia.

En cualquier caso, parece que la búsqueda de tratamiento de las excepciones será un campo de investigación recurrente en los sistemas simbólicos, ya que por lógica este tipo de aproximaciones no admite de manera natural las excepciones a las reglas. En ese sentido, los modelos estadísticos ofrecen tratamientos más globales: por ejemplo, los universales estadísticos del lenguaje cuentan con afirmaciones como "el 99% de las lenguas estudiadas en el experimento tienen alguna consonante nasal". Es decir, se han encontrado algunas lenguas que no cuentan con nasales. Esta observación es más ajustada empíricamente que decir que las consonantes nasales son universales, o que no lo son porque hay unas pocas excepciones a la regla. La estadística es un buen método de aproximarse a la realidad cuando no se pueden establecer leyes de manera discreta.

4.7. Limitaciones de los modelos simbólicos

Los sistemas simbólicos fueron concebidos por los matemáticos para captar de manera rigurosa y sistemática la demostración de teoremas mate-

máticos y lógicos. Estos sistemas simbólicos, con la aparición de la informática, se han empleado en la simulación por ordenador de fenómenos no matemáticos, como el lenguaje natural. Todo tratamiento informático de fenómenos no matemáticos por medio de sistemas simbólicos está limitado por dos grandes problemas (Ganascia, 1994):

1. La *combinatoria:* en algunos casos el espacio de búsqueda de las posibles combinaciones válidas se convierte en interminable, o al menos insatisfactorio para las expectativas de comunicación. El caso más claro es la explosión de análisis de una misma oración producida por estructuras ambiguas. En muchos casos los análisis son válidos y posibles, pero en la vida real los hablantes no son conscientes de tanta ambigüedad, pues como mucho dudan entre 2 o 3 interpretaciones. Por tanto, es ineficaz que el sistema proporcione un número mayor de análisis. Además, el hecho de que el programa busque todas las combinaciones válidas produce como consecuencia un retraso o, incluso, bloqueo en el sistema. Ya sabemos que los sistemas simbólicos comprueban la gramaticalidad de las oraciones mediante derivaciones. La solución clásica a los problemas de búsqueda y de ambigüedad es recurrir a la *heurística*, es decir, a algún método para "ayudar a descubrir" la derivación más apropiada. Utilizando un término muy habitual en IA, las heurísticas actúan como oráculos que predicen las consecuencias de las elecciones. Son una manera de introducir determinismo en un sistema por definición no determinista. Las heurísticas determinan básicamente el orden de aplicación de las derivaciones, dando preferencias a ciertas derivaciones sobre otras y evitando de esta manera las búsquedas exhaustivas. El ejemplo más típico son las heurísticas para adjunción de sintagmas preposicionales (véase el apartado 5.5.2.). Curiosamente las heurísticas se incorporan a los sistemas simbólicos para apartarlos un poco de su función original, que era la de explorar todas las posibilidades. Todo ello está justificado por la necesidad de tener implementaciones informáticas eficientes. Sin embargo, las heurísticas no son la solución definitiva al problema de la búsqueda en el espacio combinatorio, ya que en muchas ocasiones entran en conflicto unas con otras. No hay que olvidar que las heurísticas son simples estrategias desarrolladas por lo general de manera intuitiva y subjetiva, y sin que haya que suponer una garantía de acierto. Dicho de otra manera, es una forma de aproximarse a la incertidumbre, pero como se verá en el próximo capítulo, para las aproximaciones son mucho más fiables los métodos estadísticos.
2. *La adecuación* de modelo simbólico a la realidad simbolizada: esta segunda dificultad es una limitación intrínseca de los modelos simbólicos. Para

conocer la adecuación de un modelo simbólico hay que evaluar su *coherencia* y su *completitud*. El sistema es coherente si todos los resultados del sistema simbólico tienen una contrapartida en la realidad. Por ejemplo, si todas las oraciones que genera son realmente oraciones de una lengua natural. El sistema está completo si el conjunto de expresiones válidas (es decir, oraciones reales y posibles) son demostrables mediante el sistema simbólico. Dicho de otra manera, el sistema es adecuado si los datos reales y el mismo sistema concuerdan. Ahora bien, el problema está en demostrar esta concordancia. Demostrar la coherencia y la completitud de un sistema simbólico formal como las matemáticas es posible (aunque Gödel demostró que incluso en los sistemas formales hay siempre algún axioma indemostrable). Sin embargo, parece imposible o mucho más difícil decidir sobre la coherencia y completitud de un sistema simbólico acerca de una realidad no formal como el lenguaje natural. En cualquier caso, tenemos la evidencia de que las gramáticas formales no están completas y que, en ocasiones, son incoherentes con los datos lingüísticos.

Una de las características más destacadas de los modelos simbólicos en LC es su utilización de recursos lógicos. Por ejemplo, la idea del *parsing* como una deducción o derivación a partir de los axiomas establecidos en la gramática. Este concepto implica que todo fenómeno lingüístico se puede axiomatizar mediante reglas. Sin embargo, no todos los investigadores están de acuerdo con eso. Los conexionistas, por ejemplo, cuestionan la pertinencia de las reglas y de las representaciones simbólicas. Se desarrollará este punto en el capítulo 6.

Otro recurso de la aproximación lógico-simbólica es desarrollar una clasificación de tipos o clases. Esto proporciona rigor a la descripción, pero en numerosas ocasiones no es posible encontrar una clara definición que permita cortes discretos. Por ejemplo, ¿qué criterios podemos utilizar para realizar el inventario de primitivos semánticos? O incluso se puede llegar más lejos, ¿es posible establecer lógicamente todas las condiciones de buena formación de oraciones? La experiencia demuestra que los hablantes competentes no están de acuerdo acerca de la gramaticalidad de muchas oraciones (ni siquiera los gramáticos normativos coinciden en ocasiones).

En los dos capítulos siguientes se expondrán las críticas concretas a los modelos cualitativos por parte de los que adoptan una posición cuantitativa y de los que se inspiran en modelos biológicos.

4.8. Ideas principales del capítulo

Los modelos simbólicos tienen un variado y extenso repertorio de conceptos y técnicas para reflejar el conocimiento lingüístico en sus distintos

niveles. A pesar de que se afirma con frecuencia que los métodos son universales, lo cierto es que reflejan de una manera más o menos evidente las características de la lengua que se tuvo en mente cuando se diseñó la aplicación computacional. Una prueba de ello es la distinta eficacia con que se produce el procesamiento de lenguas diferentes empleando el mismo sistema computacional.

La precisión y la cobertura son los dos grandes parámetros con que se mide la adecuación de una gramática computacional a su tarea. En la última década se ha puesto de moda el diseño de técnicas de evaluación que permitan comparar distintos métodos, sistemas y gramáticas.

Cada nivel lingüístico tiene sus propias unidades y sus problemas particulares. Se enfrentan dos aproximaciones: una consiste en utilizar un marco general para todos los niveles (por ejemplo, las gramáticas de unificación y rasgos); la otra emplea métodos diferentes adaptados a las características de cada nivel (por ejemplo, la morfología en dos niveles o las redes semánticas).

Las consideraciones prácticas tienen un peso decisivo en la elaboración de una gramática computacional. La mejor recomendación es trabajar teniendo siempre en cuenta los datos. Los tres grandes problemas de los modelos simbólicos (la ambigüedad, la cobertura y las excepciones) proporcionan argumentos para investigar otros métodos que complementen la capacidad de dichos modelos.

4.9. Ejercicios

1. En español se está imponiendo la tendencia, considerada incorrecta desde un punto de vista normativo, de utilizar la preposición *a* delante de cualquier objeto directo, sea humano o no: *las preposiciones marcan a los SN, el búho se comió al ratón*, etc. ¿ A qué se debe esta tendencia ? ¿Cómo puede beneficiar la implantación de este fenómeno al procesamiento computacional del español?
2. Para ilustrar las limitaciones de la lógica formal y simbólica en IA y PLN, incluimos este fragmento de Devlin (1991: 6), *Logic and Information*, (la traducción es nuestra):

> Todos los computadores y lenguajes de programación dependen, en mayor o menor medida, de la lógica formal. A menudo esta dependencia está oculta, enterrada entre la circuitería interna de la máquina o el diseño del lenguaje de programación. Ocasionalmente, la [palabra] "lógica" es explícita, como en el caso de Prolog (acrónimo de "programación con lógica"). [...]
>
> Desafortunadamente, la lógica como la conocemos no está diseñada para manejar las demandas de un diseñador de ordenadores o un

ingeniero de software. Los ordenadores se utilizan para procesar información. La lógica actual fue diseñada para tratar los conceptos de verdad y prueba matemática, y no implica ningún intento de capturar información. No es extraño que, contando con la única herramienta disponible no bien equipada para la tarea, una investigación con objetivos tan ambiciosos como los de la Inteligencia Artificial tenga tantas dificultades.

Genuinamente, los sistemas inteligentes como el Hombre no operan con la lógica formal clásica; más bien se basan en mecanismos mucho más sofisticados de procesamiento de la información.

¿Cuáles son las limitaciones de la lógica clásica con respecto a la forma de procesar información? Los recientes desarrollos en lógica (semántica de situaciones, lógica no monótona, lógica dinámica) no se consideran a sí mismos una *alternativa* a la lógica de predicados de primer orden, sino una extensión suya que permita dar cuenta de hechos relacionados con la inferencia y el razonamiento que no pueden tratarse con los conceptos clásicos de "conectivas lógicas" y "cuantificadores".

3. El autómata mostrado en la figura 4.1 sobregenera. Por ejemplo, sería capaz de reconocer cadenas de caracteres como *conto, cuentamos, contan* o *cuentáis*. Refórmese la red para que sólo reconozca las formas gramaticales.
4. Utilizando alguno de los programas morfológicos de uso público (véase el capítulo 8), desarróllese un pequeño fragmento (por ejemplo, la flexión verbal) de la morfología de dos lenguas distintas, a poder ser de diferente tipo morfológico (por ejemplo, el catalán y el suahili). Compárense las gramáticas de ambas lenguas y la eficacia del programa para tratar problemas equivalentes.
5. Hágase una descripción de la casuística de los clíticos y las construcciones con *se* en español. Inténtese traducir a un formalismo computacional concreto y pruébese.
6. Constrúyase una red semántica sobre el dominio temático del automóvil y compárese con la proporcionada por un tesauro. ¿Coinciden las relaciones y los conceptos ?
7. Búsquese en alguna fuente teórica información sobre la construcción de expresiones de medida o fechas en español. Recurriendo a datos de textos, compruébese hasta qué punto se recogen todas las posibilidades. Una vez que se tenga una lista bastante completa, escríbanse reglas para dar cuenta de ellas.

5.
Modelos probabilísticos

5.1. Introducción histórica

La aplicación de la probabilidad y la estadística al estudio del lenguaje tiene una tradición al menos tan antigua –si no más– como la de los modelos formales. Se puede señalar a Zipf y sus célebres leyes (por ejemplo, la relación constante entre la posición de una palabra en una lista de frecuencias y la frecuencia con que aparece en el texto) así como la Teoría de la Información, de Shannon y Weaver, aplicada a las lenguas naturales. Recuérdese que los trabajos de Chomsky de finales de los años cincuenta criticaron duramente esas teorías y los modelos empiristas en general, proponiendo en su lugar la utilización de gramáticas formales.

En LC la utilización de cálculos estadísticos también se remonta a sus orígenes. Los primeros usos fueron la elaboración de índices y concordancias.

En los años sesenta comenzaron los primeros trabajos sobre corpus (por ejemplo, el Brown Corpus) que empleaban ordenadores. La elaboración de diccionarios de frecuencias también se dio en esa década (Juilland y Chang-Rodríguez prepararon uno para el español en 1964).

Aunque es poco conocido en nuestro país, y en general en la lingüística occidental, la lingüística estadística ha tenido considerable desarrollo en los antiguos países comunistas de la Europa oriental, así como en ciertos grupos aislados de Alemania (Tréveris, por ejemplo). Lo cierto es que la lingüística estadística ha sido muy minoritaria (si no tanto en número, sí en prestigio e influencia) debido a la fuerza de las corrientes formales y cualitativas.

Uno de los casos más injustamente olvidados ha sido el uso de métodos estadísticos en la sociolingüística cuantitativa. Labov y Sankoff son sus dos

principales investigadores. Desde mediados de los años setenta han desarrollado sucesivas versiones del programa VARBRUL, que permite trabajar con variables dependientes e independientes y extraer conclusiones estadísticas. Este programa se puede conseguir vía ftp en la siguiente dirección:

ftp://ftp.cis.upenn.edu/pub/ldc/misc_sw/varbrul.tar.Z

En resumen, durante varias décadas se estuvieron utilizando métodos estadísticos en programas de ordenador para estudiar sobre todo la variación lingüística, desde las variantes sociolingüísticas hasta las de estilo. Fue a finales de los años ochenta, sin embargo, cuando se produjo un gran cambio de tendencia: se empezó, dentro de la LC, a desarrollar modelos puramente estadísticos en la modelización de una lengua. Es decir, ya no se pretendía exclusivamente dar cuenta de la variabilidad de los fenómenos lingüísticos o recoger una serie de regularidades estadísticas, sino que se presentaba como la alternativa a la visión lógico-simbólica de los modelos formales. Los grupos pioneros se dieron en la Costa Este americana (Pennsylvania, AT&T Bell Labs e IBM Yorktown). Su principal influencia llegó de los avances en procesamiento del habla, donde estos modelos consiguen mucho mejores resultados que los basados en conocimiento lingüístico.

En los años noventa podemos decir que esta tendencia se ha "institucionalizado": se dedican dos números monográficos de la revista *Computational Linguistics* a este tema (volumen 19, números 1 y 2, marzo y junio de 1993). Por otra parte, las comunicaciones en congresos y artículos en revistas han aumentado considerablemente a lo largo de los años noventa.

Al igual que los modelos simbólicos, hay diferentes tipos de aproximaciones a la modelización estadística. Igualmente, cada modelo estadístico tiene sus puntos fuertes (donde supera a otros modelos) y sus puntos débiles. Los primeros proyectos eran muy entusiastas con las posibilidades todavía inexploradas por los modelos cuantitativos, pero a medida que se han ido desarrollando las aproximaciones se han conocido sus limitaciones: es muy difícil mejorar la precisión una vez llegado a un punto determinado.

La conclusión general que se ha extendido dentro de la disciplina es que se necesita la combinación de diferentes modelos dentro de un mismo sistema para conseguir superar las limitaciones inherentes a cada uno.

5.2. El atractivo de los modelos estadísticos

Eugene Charniak es uno de los investigadores más reconocidos en PLN, sobre todo desde la perspectiva de la IA. Trabajó durante dos décadas (la de los setenta y ochenta) en modelos de representación del conocimiento para com-

prensión de textos, utilizando la aproximación simbólica. A principios de los años noventa se produjo lo que él mismo denomina su "conversión" al procesamiento estadístico del lenguaje (Charniak, 1993). Algo parecido se puede decir de Fernando Pereira, uno de los investigadores más productivos e influyentes en la LC simbólica de los ochenta. Abney (1996) afirma que en 10 años los métodos estadísticos han pasado de ser virtualmente unos desconocidos a ser algo fundamental. ¿A qué se puede deber todo este cambio de actitud? Como suele ocurrir, por una parte debemos buscar las causas de la crisis del paradigma vigente y por otra las expectativas que despierta el nuevo paradigma.

5.2.1. Crisis de los modelos racionalistas y simbólicos

Church y Mercer (1993) destacan la importancia del redescubrimiento en los años noventa de los métodos empíricos y estadísticos de la lingüística de los cincuenta. Señalan a su vez tres factores determinantes en esta nueva época para los métodos cuantitativos, aunque sólo relevantes desde el punto de vista de la LC:

1. Los ordenadores son considerablemente más potentes y accesibles que cuando se empezaron a aplicar por primera vez las ideas estadísticas al estudio del lenguaje, en los años cincuenta.
2. La accesibilidad a los datos en forma de corpus electrónicos en la actualidad es muy superior a décadas anteriores. Y no sólo para el inglés, ya que los organismos nacionales e internacionales han incentivado en los años noventa la elaboración de corpus en otras lenguas.
3. Hay presión por parte de las instituciones que costean las investigaciones en LC para obtener resultados utilizables. Tanto en Estados Unidos como en la Unión Europea, las evaluaciones rigurosas de los resultados obtenidos se han convertido en el principal instrumento para conceder o rechazar proyectos. Si el dominio de aplicación se amplía, los métodos estadísticos consiguen resultados que no pueden conseguirse con métodos simbólicos. La "robustez" (la capacidad para no quedarse atascado cuando el sistema se enfrenta a un problema, es decir, una construcción o una palabra desconocida) de los métodos estadísticos es mayor.

Por tanto, factores tecnológicos (mejores ordenadores, mayor cantidad y calidad de datos en formato electrónico) y factores económicos (la presión de los organismos que subvencionan las investigaciones) pueden explicar el auge de la estadística en LC.

Otros autores como Susan Armstrong-Warwick (1993) o E. Charniak (1993) insisten en que la causa fundamental de este cambio paradigmático

se encuentra en la "frustración creciente al aplicar los métodos clásicos basados en reglas" a textos no restringidos. Armstrong-Warwick destaca la ausencia relativa de resultados prácticos y el éxito de los modelos estadísticos en los sistemas de reconocimiento de habla como acicate para probar nuevos caminos en LC.

Según Charniak, el estancamiento de los sistemas basados en el conocimiento se debe a que en ellos se asume que la comprensión de las lenguas naturales depende básicamente de una gran cantidad de "conocimiento del mundo", por lo tanto los sistemas de PLN tienen que contar con dicho conocimiento para tener éxito en su simulación de la facultad lingüística. Parece un hecho indiscutible el que, a pesar de los múltiples intentos realizados en IA, no se dispone de un modelo o marco formal para representar con éxito dicho conocimiento de "sentido común" que todo hablante parece emplear para entender los mensajes que recibe.

5.2.2. Expectativas del paradigma empirista y estadístico

Precisamente uno de los mayores atractivos de los modelos estadísticos en LC es que permiten analizar mucho mejor grandes cantidades de texto sin restricciones de dominio, algo que es imposible con los actuales sistemas basados en el conocimiento, que se limitan a tratar aceptablemente dominios acotados. A cualquier lingüista computacional, en principio, le interesa dar cuenta del procesamiento lingüístico que hacen los seres humanos. Por tanto, uno de los mayores desafíos de la LC es simular cómo pueden los hablantes manejar todo tipo de emisiones lingüísticas con tan poco esfuerzo, sobre todo si lo comparamos con las limitaciones de los sistemas computacionales clásicos. En este sentido, los métodos estadísticos funcionan mucho mejor que los basados en las gramáticas formales, ya que sin llegar a resultados nada comparables con la actuación humana, al menos producen resultados utilizables.

Otro atractivo importante de los métodos estadísticos es que permiten de una manera natural y simple implementar el "aprendizaje" de una lengua. Efectivamente, una de las aplicaciones más investigadas en los últimos años es la inferencia automática de reglas gramaticales a partir de datos.

Por último, los métodos cuantitativos proporcionan un remedio al gran problema de los modelos cualitativos: la ambigüedad. A la hora de tener que escoger entre varias estructuras posibles o varias interpretaciones, la estadística proporciona la pista de las probabilidades de cada opción, de tal manera que siempre se puede escoger la más probable. Esto supone una considerable ventaja sobre los modelos que sólo incorporan conocimiento basado en la competencia.

Resumiendo, los tres puntos fuertes del procesamiento estadístico residen en :

1. Tratar cualquier dominio lingüístico.
2. Tratamiento de la ambigüedad por medio de la ordenación de las opciones según su probabilidad en un contexto dado.
3. Capacidad para aprender automáticamente reglas.

5.3. Modelización estadística de las lenguas naturales

Se empezará esta sección presentando los conceptos generales que subyacen a cualquier modelo estadístico. Es importante hacer notar que es muy poco habitual que un lingüista haya recibido alguna formación en modelos cuantitativos. De hecho los manuales de lingüística matemática (por ejemplo, Partee *et al.*, 1988; los españoles como Serrano, 1977 o Moreno Cabrera, 1994) no incluyen ningún apartado sobre lingüística estadística. En este sentido, cabe destacar el recomendable capítulo 3 de Martínez Celdrán (1991).

Algo similar podría decirse de los informáticos, cuya formación básica se centra en modelos formales, distintas lógicas, autómatas, etc. El propio Charniak en su libro de 1993 avisa que la mayor parte de los conocimientos supuestos en procesamiento estadístico no se incluyen en los programas curriculares de Informática en las universidades de todo el mundo, siendo ésa una de las razones que le llevaron a escribir su manual. Sin embargo, esta observación no es cierta aplicada a los estudiantes de informática en España, donde consta que se dan asignaturas troncales de estadística en todos los planes de estudios.

Esta exposición, por tanto, va dirigida fundamentalmente a los lingüistas y su pretensión es introducir de manera intuitiva los conceptos necesarios para entender los modelos y aplicaciones que se explican a continuación.

5.3.1. Conceptos esenciales de Probabilidad y Estadística

Lo primero que hay que definir y distinguir es qué es la Teoría de la Probabilidad y qué es la Estadística. La Teoría de la Probabilidad, en una definición sencilla, consiste en un "conjunto de proposiciones y teoremas que permite calcular la probabilidad de una gama muy variada de sucesos partiendo de determinadas probabilidades que se suponen dadas de antemano" (Bernis *et al.*, 1981: 667). El ejemplo más típico es el del lanzamiento de la moneda o del dado. En el primer caso, la probabilidad a priori de que salga cara o cruz es de 1/2; en el segundo la probabilidad elemental de que salga cualquiera de las caras es de 1/6.

A partir de estos supuestos, la Teoría de la Probabilidad ha desarrollado conceptos fundamentales como el de *suceso aleatorio* y su probabilidad, así como la *probabilidad condicionada* (es decir, la probabilidad de que se produzca un suceso que dependa de otro anterior). Ahora bien, estos conceptos funcionan sobre probabilidades dadas: si queremos calcular la probabilidad real de que salga un 2 en un dado trucado, por ejemplo, no podremos partir del supuesto de que su probabilidad es 1/6. No nos quedará otro remedio que hacer una larga serie de lanzamientos del dado, anotar las apariciones de cada lado, calcular las frecuencias relativas y, entonces, intentar predecir las probabilidades reales de la cara 2 en nuestro dado trucado.

En este ejemplo habríamos estado haciendo Estadística: determinar las probabilidades reales de los sucesos, basándonos en la frecuencia relativa que hemos anotado. La *Encyclopaedia Britannica* define la Estadística como "the art and science of gathering, analyzing, and making inference from data". Por supuesto, la ciencia que recoge, analiza y predice a partir de datos o sucesos utiliza los instrumentos y conceptos que le proporciona la Teoría de la Probabilidad. En la actualidad, la Estadística constituye uno de los métodos de análisis y predicción más utilizados e importantes en las ciencias naturales y sociales, desde la Física y la Biología hasta la Psicología, la Sociología y la Economía.

En una ciencia social como la Lingüística también tiene aplicación la Estadística, aunque hasta la fecha no haya sido utilizada muy extensamente. La Estadística tiene por objeto el estudio de poblaciones, colectivos, conjuntos de individuos que tienen alguna característica o comportamiento en común. Del estudio del comportamiento individual se pueden establecer unas leyes generales sobre el comportamiento de tipo predominante o promedio. Por ejemplo, se puede estudiar e intentar predecir los resultados de unas elecciones analizando las respuestas de algunos individuos de la población votante. De igual forma, se puede estudiar estadísticamente el lenguaje: supongamos que tomamos como población un conjunto de datos lingüísticos o *corpus* (puede ser el vocabulario de un escritor, los sonidos emitidos en un programa radiofónico, las oraciones recogidas en un manual de gramática, etc.). Es decir, tendríamos un subconjunto de una determinada lengua en un determinado nivel lingüístico. Es de suponer que las unidades de este subconjunto mantendrán algún tipo de relación entre sí. Podemos, por tanto, estudiar algunas de estas unidades y establecer algunas reglas de comportamiento: por ejemplo, la probabilidad condicionada de que aparezca un sonido oclusivo detrás de uno nasal, la preferencia del autor X por determinado vocablo Y en lugar de su sinónimo Z, o la probabilidad de que un nombre aparezca en determinada posición en la oración.

Evidentemente, estas leyes pueden establecerse mediante un estudio completo de la población, estudiando una por una todas las unidades (sonidos, pala-

bras, etc. según el tipo de nivel-población que hayamos escogido). Sin embargo, en la mayoría de los casos nuestro subconjunto lingüístico tendrá un tamaño tan inmenso (por no decir infinito), que su estudio exhaustivo será del todo inviable. Es posible estudiar el número de palabras utilizadas por un autor prolijo, o el número de construcciones sintácticas que pueden aparecer en cualquier texto voluminoso, pero pensemos en intentar comparar varios autores o varios textos: llegará un momento en que, sencillamente, será imposible. En último término, nunca se podrá recoger todos los aspectos de una lengua, puesto que el número de oraciones posibles es infinito.

Descartada la exhaustividad en el acercamiento a un subconjunto lingüístico (al que a partir de este momento denominaremos *sublengua)* se impone inferir las leyes de comportamiento de dicha sublengua a partir de las leyes de comportamiento de determinados subconjuntos de ésta, llamados *muestras*. Consideraremos, por tanto, que una determinada sublengua puede ser recogida en un corpus y que las muestras serán ejemplos tomados de dicho corpus. Dicho de otra forma, ante la imposibilidad de realizar un estudio exhaustivo, por ejemplo de las construcciones sintácticas utilizadas en los periódicos actuales (la sublengua periodística), se elige una muestra de entre todas las oraciones aparecidas en los periódicos, se estudian y, utilizando instrumentos de probabilidad, se infieren reglas para toda la sublengua periodística. Las observaciones realizadas en la muestra se extienden a toda la población.

Pero las muestras no se pueden elegir de cualquier manera. Si elegimos oraciones de periódicos conservadores el resultado de nuestras estimaciones será probablemente diferente del calculado a partir de datos en periódicos progresistas; si tomamos más ejemplos de los artículos de opinión, las conclusiones serán muy distintas de las que se puedan extraer de la sintaxis de la información meteorológica, etc. Es decir, si primamos una tendencia, las conclusiones serán muy poco representativas de la sublengua en cuestión. Como norma general en estadística, las muestras no deben presentar predisposiciones sistemáticas, sino que deben ser representativas de la composición de la población. De ahí que, en general, se prefiera escoger los elementos de la muestra al azar.

Se denominan *métodos de muestreo* los diversos métodos que existen de elegir muestras al azar. El más sencillo es el *muestreo aleatorio simple*, que consiste en numerar a los individuos de la población y extraer la muestra por algún sistema de lotería o mediante algún sistema de obtención de números aleatorios (algunos lenguajes de programación tienen instrucciones para ello, por ejemplo. También existen tablas con sucesiones preestablecidas de números obtenidos al azar). Si aplicamos el método a nuestro corpus de oraciones de periódico, numerándolas y extrayéndolas aleatoriamente, obtendremos una muestra válida para nuestro estudio estadístico.

El siguiente paso del muestreo se conoce como la *inferencia estadística*. Consiste en trasladar las leyes observadas en la muestra a toda la población. Supongamos que hemos establecido una clasificación de construcciones sintácticas que aparecen en textos periodísticos y las hemos ordenado según su frecuencia relativa de aparición en nuestra muestra. Nuestra tarea ahora consiste en extrapolar esos resultados a la sublengua periodística. Esta inferencia presenta una problemática variada, ya que hay limitaciones evidentes. En pocas palabras, es necesario plantear la significación de los resultados obtenidos y la posibilidad de error en nuestras predicciones. El *contraste de la hipótesis*, término con que se conoce esta fase, es básico para la decisión o inferencia estadística.

El contraste de la hipótesis consiste básicamente en decidir si una población cumple o no cierta ley de comportamiento que hemos inferido del estudio de una muestra. La ciencia estadística, tomando como punto de apoyo la teoría de la probabilidad, ha desarrollado una serie de conceptos como *intervalo de confianza*, *probabilidad de error*, *certidumbre estadística* para medir la capacidad de predicción de la hipótesis. Una prueba básica es la del contraste con la *hipótesis nula*: una vez que hemos establecido una regularidad estadística se utiliza para comprobar si tiene relevancia o no. Estos conceptos, al igual que se aplican a los sondeos de opinión, se utilizan en los estudios lingüísticos (pueden consultarse en manuales como el de Woods, Fletcher y Hughes, 1985, capítulo 8).

En esta presentación no se ha hecho referencia a conceptos básicos de estadística (cuadro 5.1) descriptiva *(media, mediana, desviación típica, moda, etc.)* que el lector interesado puede encontrar en cualquier texto introductorio.

CUADRO 5.1. Algunos de los conceptos básicos de estadística.

Estadística descriptiva	*Inferencia estadística*
Media, mediana, desviación típica, moda, muestra, métodos de muestreo	Intervalo de confianza, probabilidad de error, certidumbre estadística, hipótesis nula

Regresemos ahora a la Teoría de la Probabilidad. El concepto de *regularidad estadística* se basa en lo siguiente: si repetimos un número elevado de veces una experiencia aleatoria (por ejemplo, la aparición de determinado fonema o determinada construcción) empíricamente se comprueba que su frecuencia relativa se estabiliza, es decir, se observa que la frecuencia relativa tiene una marcada tendencia a permanecer constante cuando aumenta el número de datos consultados. La idea de la regularidad sugiere que si repitiésemos la experiencia un número infinito de veces, las frecuencias alcanzarían un determinado valor límite (en el caso de una moneda, el valor límí-

te de "cara" sería 1/2). Esto nos permite asignar un número *P* entre 0 y 1, tal que la frecuencia relativa de *A*, tras una larga serie de repeticiones de la experiencia, se aproxime a *P*, que llamaremos la *probabilidad* de *A*. De este modo, la probabilidad de un suceso sería el límite de la frecuencia relativa cuando el número de repeticiones tiende a infinito. Dicha definición permite la asignación de un número (su probabilidad) a cada suceso (cuadro 5.2), de modo que quede cuantificada la idea intuitiva de la probabilidad.

CUADRO 5.2. Ejemplos de probabilidad en el lenguaje.

Probabilidad de una consonante en español	0,5288
Probabilidad de una vocal en español (datos de Rojo, 1991)	0,4712
Probabilidad de la sílaba *de* en español	0,041
Probabilidad de la sílaba *do* en español	0,018
Probabilidad de la sílaba *da* en español	0,010

Fuente: datos calculados a partir de Alameda y Cuetos, 1995.

Las regularidades estadísticas significativas en el lenguaje no dependen de los hablantes ni tampoco del tipo de dominio temático: es asombroso comprobar cómo en la práctica el comportamiento lingüístico se ajusta a los pronósticos estadísticos (Crystal, 1987). Una de las leyes estadísticas más antiguas es la propuesta por Kaeding en 1898: se da una relación inversa entre la longitud de la palabra en términos de sílabas y la frecuencia con que la palabra aparece. Dicho de otra manera, las palabras más utilizadas son las más cortas. En español, las 10 palabras más frecuentes son monosilábicas: *de, el, la, y, a, en, él, que* (pronombre), *ser, que* (conjunción). Lo significativo es que esta regularidad se cumple sistemáticamente en lenguas con distinto grado de complejidad morfológica, como el inglés (que es una lengua de tendencia aislante), el español (con bastante morfología flexiva, sobre todo verbal) o el alemán (que se caracteriza por una morfología compositiva muy rica y una estructura marcadamente polisilábica).

Esta ley sirve para ilustrar uno de los puntos esenciales de los métodos estadísticos: podemos encontrar numerosísimas regularidades, de las cuales la mayor parte son irrelevantes e incluso perturbadoras de la realidad (Charniak, 1993). Por tanto, la tarea central consiste en encontrar las regularidades estadísticas apropiadas.

Se utilizará un ejemplo no lingüístico para ilustrar este punto tomado del ingenioso libro del matemático J. A. Paulos (1995). Es muy habitual que la gente encuen-

tre coincidencias significativas en hechos fortuitos. Por ejemplo, alguien ha pensado esa mañana en una persona que no ha visto desde hace años y a la que ha perdido la pista y, de pronto, se la encuentra en el dentista. "El mundo es un pañuelo", "el destino está escrito" y otras tantas máximas populares reflejan una tendencia natural en el ser humano a explicar las coincidencias. Efectivamente, se trata de encontrar significado a una coincidencia cuya probabilidad de producirse es baja. Si bien esto es cierto, el fallo está en no darse cuenta de que aunque la probabilidad de que un determinado suceso ocurra (por ejemplo, el encuentro de los dos viejos amigos en el dentista) es muy baja, en cambio hay bastante probabilidad de que algunas coincidencias asombrosas se den. El cuadro 5.3. muestra algunas coincidencias entre los presidentes Lincoln y Kennedy, y entre los también presidentes americanos asesinados William McKinley y James Garfield. Como señala Paulus, el hecho de que Lincoln y Kennedy hayan sido dos de los personajes más importantes de la historia de los Estados Unidos ha favorecido que la gente busque coincidencias entre sus vidas, sin molestarse en buscarlas en las vidas de McKinley y Garfield. En consecuencia, no se debe inferir que existe una regularidad estadística, por ejemplo, en el hecho de ser elegido presidente de los EE UU en el año sesenta de cada siglo y ser asesinado. Las coincidencias, por sí solas, no implican una regularidad o una ley.

CUADRO 5.3. Coincidencias en las vidas de 4 presidentes americanos asesinados.

Entre Lincoln y Kennedy:

- Lincoln fue elegido en 1860; Kennedy en 1960
- Sus nombres tienen 7 letras
- Lincoln tenía una secretaria que se llamaba Kennedy; Kennedy tenía una secretaria que se llamaba Lincoln
- Ambos fueron asesinados por individuos que tenían tres nombres (John Wilkes Booth y Lee Harvey Oswald)
- El asesino de Lincoln le disparó en un teatro y se escondió en un almacén; el asesino de Kennedy disparó desde un almacén y se escondió en un teatro

Entre McKinley y Garfield:

- Ambos eran republicanos que nacieron y se criaron en Ohio
- Ambos eran veteranos de la Guerra civil
- Ambos fueron congresistas
- Ambos fueron ardientes defensores del proteccionismo y del estándar basado en el oro
- Ambos nombres contienen 8 letras
- Ambos fueron asesinados en el primer mes de septiembre de sus respectivas presidencias por individuos con nombres extranjeros

Fuente: Datos tomados de Paulus, 1995: 51.

La Teoría de la Probabilidad no es otra cosa que la disciplina matemática que estudia las leyes por las que se rigen los fenómenos fortuitos, de tal manera que podamos distinguirlos de los que no son fortuitos (que por lo tanto tienen una causa lógica). Para los fenómenos aleatorios, el cálculo de probabilidades nos informará provisionalmente y de manera aproximada. Precisamente se han desarrollado los métodos de contraste de hipótesis, como el de la hipótesis nula, para discriminar entre las regularidades estadísticas significativas y las que no lo son. Se deja para la sección 5.4 la exposición de los conceptos de probabilidad condicionada e independencia entre sucesos.

La aproximación estadística en muchos casos es más ajustada y predictiva de lo que se puede esperar de una ley lógica. El lector de formación lingüística puede que no esté familiarizado con el uso de los modelos probabilísticas en Física Cuántica. Se utilizará un ejemplo clásico: el comportamiento de las moléculas de un gas en un recipiente. "Si su velocidad respondiera a alguna ley física sencilla, sería casi imposible hacer alguna predicción sobre su comportamiento. Sin embargo, gracias a la hipótesis de que dichas velocidades varían al azar se obtienen las sencillas leyes de Gay-Lussac y de Mariotte" (Bernis *et al.*, 1981).

Podemos resumir el método probabilístico-estadístico en dos:

— La idea general: inferir conocimiento directamente de los datos, buscando regularidades significativas
— La estrategia general: contar con la mayor cantidad posible de datos para poder establecer una probabilidad lo más cercana a la frecuencia relativa estable

5.3.2. Relevancia de la estadística en el estudio del lenguaje

Abney (1996) proporciona una excelente presentación, actualizada, de la relevancia de los métodos estadísticos en Lingüística, ya sea Teórica o Computacional. Se resumen a continuación sus argumentos:

— *Adquisición del lenguaje*: una manera de explicar la variación de comportamiento en el aprendizaje de una lengua es suponer que los cambios en la gramática del niño (cuando éste altera una regla o parámetro como por ejemplo la conjugación de un verbo irregular) reflejan cambios en las frecuencias relativas de uso de las estructuras implicadas durante un período de tiempo. Esta visión propone que las dos versiones de la regla (la vieja y la nueva) coexistirían durante un período de prueba, durante el cual la probabilidad de una y otra irían cambiando hasta que la versión más antigua acabara en una probabilidad de 0.

— *Cambio lingüístico:* en general, los cambios producidos en las lenguas no son abruptos sino más bien graduales. Al igual que en el caso de la adquisición, pero esta vez en lugar de la gramática del niño en la gramática de la comunidad, podemos asumir que los cambios son modificaciones en la frecuencia relativa de las construcciones dentro del habla de la comunidad en un período extenso de tiempo. Es importante destacar que en ambos casos, en la adquisición y en la evolución de una lengua, se asume que hay básicamente una gramática simbólica suplementada por una gramática estocástica.
— *Variación lingüística:* la diversidad en el lenguaje se manifiesta tanto interlingüísticamente (donde el lenguaje humano se presenta en una gran colección de lenguas diferentes que son una muestra particular de él) como intralingüísticamente (donde cada lengua se compone de diferentes dialectos, sociolectos, etc.). Los modelos estadísticos son significativos en el estudio de la variación lingüística por los motivos mencionados al principio de esta sección: permiten dar cuenta de la gradación mucho mejor que los modelos simbólicos discretos.
— *Competencia frente* actuación: *la ficción de la homogeneidad.* La confrontación en este punto es frontal con la tradición lingüística mayoritaria en este siglo que, desde Saussure, entiende que los aspectos interesantes del lenguaje son aquellos que tienen que ver con el conocimiento homogéneo compartido por los hablantes. De hecho, el argumento principal de los lingüistas teóricos contra los métodos estadísticos es que éstos no proporcionan ninguna ayuda para estudiar lo que ellos consideran relevante: la competencia. Por su parte, los defensores de los métodos estadísticos argumentan que estudiar la competencia es limitar y simplificar el estudio del lenguaje, ya que hay muchos aspectos de la actuación que son propiamente lingüísticos y, por tanto, deben ser tratados por la teoría lingüística. Abney afirma que los métodos simbólicos son incapaces de explicar satisfactoriamente algunas propiedades importantes del lenguaje humano, entre otras:

a) *La gramaticalidad y la aceptabilidad:* éste es un tema clásico que Chomsky ha tratado en diferentes ocasiones. Un modelo simbólico discreto lo único que puede hacer es definir las condiciones de cuándo una oración es gramatical o agramatical y, a lo sumo, establecer que una oración es aceptable (es decir, en el caso en que se viola alguna restricción no fundamental). Cualquier lingüista computacional que haya escrito una gramática simbólica recordará su estupor ante el enorme número de análisis que ha producido una oración, muchos de ellos verdaderamente increíbles. Análogamente, cualquier lingüista que haya realizado encuestas

de gramaticalidad entre hablantes competentes habrá observado con asombro la divergencia de criterios con respecto a oraciones con una estructura algo problemática. Estas limitaciones son propias de los modelos basados en criterios cualitativos, pues cuando la complejidad aumenta, la dificultad de encontrar elementos de juicio objetivos para establecer cortes discretos se convierte en un problema real. ¿Qué pueden aportar los modelos estadísticos al respecto? Proporcionan una cuantificación que permite establecer una gradación fiable de la gramaticalidad. Es decir, se asigna un *peso* específico (una probabilidad) a cada estructura en un contexto determinado, basándonos en probabilidades extraídas de datos. Para elegir el mejor análisis para una oración de todos los ofrecidos por nuestra gramática, sólo tenemos que sumar los pesos combinados de todas las estructuras que aparecen en la oración y escoger el que obtenga mayor puntuación. De esta manera, aunque haya varios análisis gramaticales, habremos tomado el que se ha percibido como el más apropiado en anteriores contextos similares.

b) *La tolerancia al error:* una de las propiedades más asombrosas de la comprensión lingüística humana es su tolerancia al error: los hablantes perciben y comprenden estructuras que una gramática simbólica daría como agramaticales. Considérese la siguiente oración: *Yo soy de los que pienso...* Muchos hablantes del español utilizan y prefieren esta versión "agramatical" a su versión "correcta": *Yo soy de los que piensan*, donde el verbo de la cláusula subordinada relativa concuerda en persona y número con el sujeto de su oración (el pronombre "los que") en lugar de concordar con el sujeto de la oración principal (el pronombre "yo"). Probablemente la razón de esta preferencia se deba a que el significado interfiere en la estructura, de tal manera que el agente (el emisor en este caso) se "apropia" del sujeto sintáctico en ambas oraciones. Análogamente, no es difícil interpretar oraciones como *Tú eres de los que piensas...* o *Ella es de las que piensa...* Desde el punto de vista teórico, estos casos pueden desecharse alegando que son construcciones contaminadas por factores comunicativos: los hablantes competentes ideales (es decir, los lingüistas, según Chomsky) son capaces de reconocer la violación estructural. Sin embargo, muchos hablantes competentes no perciben dicha violación. Los lingüistas computacionales se enfrentan con mucha más asiduidad a oraciones producidas por hablantes en contextos comunicativos reales que a oraciones perfectamente gramaticales. Por tanto, si sus sistemas tienen que ser robustos y eficientes, deben detectar y entender los "errores" que producen los hablantes. Pre-

cisamente uno de los métodos estadísticos más conocidos, el modelo del canal de ruido *(noisy channel)* de Shannon, proporciona los fundamentos para tratar errores mediante distintas técnicas computacionales.

c) *La adquisición del léxico y la gramática:* como se vio al tratar las gramáticas y los lexicones formales, una meta esencial para cualquier sistema de PLN es conseguir recoger la mayor cantidad de conocimiento lingüístico. Si uno intenta dar cuenta de manera sistemática de la estructura de constituyentes del español, por ejemplo, encontrará gran variedad y cantidad de construcciones, consideradas "menores" o poco "interesantes" por los manuales de gramática, que se separan de las reglas generales. Por ejemplo, cómo construimos las fechas, las operaciones matemáticas, las unidades de medida, los nombres propios, los títulos, etc., todas ellas muy comunes en textos de uso general. Una de las aplicaciones más importantes de los métodos empiricistas es la adquisición de las estructuras sintácticas más frecuentes. Pero sin duda lo más difícil y costoso de integrar es el conocimiento léxico, y en esa parcela la aplicación de métodos de inferencia inductiva ha sido muy productiva: se han desarrollado técnicas para adquirir compuestos terminológicos, colocaciones, marcos de subcategorización y restricciones seleccionales, entre otros. Por supuesto, estas técnicas no son adecuadas por sí solas para ser consideradas modelos de adquisición humana, como advierte Abney, pero su efectividad en LC hacen de ellas un recurso esencial para cualquier sistema de gran cobertura.

5.3.3. La Teoría de la Información aplicada al lenguaje natural

Los trabajos de Shannon y Weaver influyeron de manera considerable en la lingüística pre-chomskyana. Efectivamente, para los estructuralistas, interesados básicamente por el papel del lenguaje en las relaciones humanas, la Lingüística forma parte de la teoría general de la comunicación. En concreto, los distribucionalistas americanos defendían que el interés del estudio del lenguaje no estaba en el conocimiento en sí mismo, sino en su ayuda para comprender y mejorar los procesos comunicativos. Significativamente, esta visión esencialmente comunicativa del lenguaje es la que siempre ha prevalecido en la LC, frente al planteamiento biológico y psicologista del paradigma chomskyano y, por extensión, de la LT de los últimos treinta años.

A) La comunicación

Se dará una descripción teórica general del proceso de comunicación desde una perspectiva lingüística:

Emisor —— Mensaje codificado/Canal ——— Receptor

1. El emisor quiere intercambiar información con un receptor y emite un mensaje codificado, transmitido a través de un canal y siguiendo un código determinado. En nuestro caso, el código sería una lengua natural y el canal podría ser un medio oral o escrito.
2. El mensaje codificado puede ser distorsionado o cambiado mientras se transmite. A los efectos distorsionantes se les denomina *ruido*.
3. El mensaje llega al receptor, que lo descodifica y recupera la información original enviada por el emisor. El hecho de que entienda el mensaje depende del número de posibilidades que tenga de entender el comportamiento del emisor. Es decir, *el grado de comprensión está en gran medida relacionado con el grado de predictibilidad de la información comunicada*. La aplicación del código inadecuado da lugar a una interpretación errónea.

Este esquema general se puede aplicar a todos los sistemas comunicativos, desde las lenguas naturales a la escritura musical, es decir, a cualquier sistema que esté compuesto de signos que tengan un significado asociado a ellos. Con estos signos y unas reglas de combinación cualquier procesador de información (humano o artificial) puede codificar y descodificar información, así como almacenarla en su memoria.

B) La Teoría de la Información

A partir del esquema general de la comunicación se ha desarrollado la Teoría de la Información, cuyo objetivo es descubrir las leyes matemáticas que gobiernan los sistemas diseñados para comunicar o manipular información. En concreto, establece medidas cuantitativas de la información y estudia la capacidad de distintos sistemas para transmitir, almacenar y procesar información. Por lo que concierne a este libro, interesa la Teoría de la Información aplicada al estudio de los sistemas de las lenguas naturales. A continuación se ven algunos conceptos fundamentales.

Rendimiento funcional

En cualquier nivel lingüístico nos podemos encontrar con dos o más elementos que pueden aparecer en la misma posición. Esto se conoce con el término de *contraste paradigmático*. Lo significativo, desde el punto de vista de la información, es que no todos los contrastes paradigmáticos tienen la misma importancia en el funcionamiento de una lengua. El rendimiento funcional es una medida para conocer la importancia distintiva de la oposición contrastiva dentro de un sistema. Por ejemplo, el contraste entre /r/ y /rr/ en español (por ejemplo, *caro* frente a *carro*) es superior a la oposición /y/ y /ll/ *(poyo* frente a *pollo)*, que no llega a 20 parejas.

La importancia del rendimiento funcional de los contrastes es clara, ya que es necesario que existan contrastes generales que faciliten la economía lingüística y la reducción del número de situaciones ambiguas en la interpretación de los mensajes. Cuanto más rendimiento funcional tiene una oposición, más importante es que los hablantes la incorporen a sus hábitos lingüísticos. En ese sentido, es de esperar que los niños aprendan antes los contrastes con mayor carga funcional y, por otra parte, estos mismos contrastes serán los que más resistan al cambio.

Sin embargo, a pesar de la evidente utilidad del concepto, la cuantificación precisa del rendimiento funcional es complicada, pues depende de dos factores:

a) La posición estructural o contexto donde se realiza el contraste, ya sea dentro de la palabra para el caso de los fonemas, ya sea la posición entre los constituyentes oracionales para el caso de las categorías sintácticas.
b) La frecuencia de aparición de los elementos en contraste, que no está determinada por el número de elementos que opone el contraste. Es decir, se puede dar el caso de que una oposición permita distinguir bastantes significados pero que éstos tengan una aparición relativamente baja en la lengua; y por otra parte, podríamos encontrar una oposición de pocos elementos pero muy frecuentes.

Lyons (1968) concluye que, dada por una parte la importancia del concepto de rendimiento funcional y por otra parte la dificultad para llegar a un medición precisa, "es posible aún decir que determinados contrastes tienen un rendimiento funcional más elevado que otros, aunque no podamos decir en qué medida exacta".

Contenido informativo

Es otro concepto fundamental en Teoría de la Información. La cantidad de información de una unidad lingüística en un contexto dado viene determinada por su probabilidad de aparición en tal contexto. Un principio básico de la Teoría de la Información dice que *el contenido informativo es inversamente proporcional a la probabilidad.*

Lyons señala que este principio puede parecer sorprendente a primera vista: cuanto más previsible es una unidad menos información aporta. Por ejemplo, podemos establecer una escala de 1 a 0, donde la probabilidad máxima corresponda a 1 y la mínima a 0. En oraciones como "A Juan le gusta hablar ... fútbol", si omitiéramos la palabra "de" en este contexto no habría ninguna pérdida de información. Esto es debido a que esta palabra no se halla en contraste paradigmático con ninguna otra unidad del mismo nivel que pueda aparecer en el mismo contexto. Por tanto, podemos asignar a dicha palabra una probabilidad de 1 y un contenido informativo de 0. Dicho de otra manera, "de" es totalmente *redundante* en este contexto.

Es importante entender que rendimiento funcional y contenido informativo no están en contradicción: si una unidad aparece muchas veces en contraste con otras unidades, su rendimiento funcional será elevado (y por tanto será más útil a los hablantes; sin embargo, desde el punto de vista informativo será menos útil dado que su predictibilidad es grande. Por ejemplo, "Ayer vi un perro a dos metros de mí" es menos informativa que "Ayer vi un oso panda a dos metros de mí", pues los osos panda se encuentran en muchos menos contextos que los perros, y el receptor puede interpretar que el emisor estuvo en el zoo. Sin embargo, no se puede decir que haya unidades más o menos informativas en abstracto, sino que es más apropiado hablar de contextos más o menos informativos en función del número de unidades que pueden aparecer. En una situación donde se está hablando de animales en peligro de extinción es mucho más predecible que aparezca el oso panda que el perro.

Es decir, el hecho de que un elemento sea redundante en un contexto determinado no significa que su rendimiento funcional sea nulo (más bien todo lo contrario) o que su capacidad informativa no exista en otros contextos.

CUADRO 5.4. Conceptos básicos de Teoría de la Información.

Rendimiento funcional *depende de* la frecuencia de aparición de un contraste
Contenido informativo *depende de* la probabilidad de aparición de una unidad en un contraste determinado

Importante: no identificar *frecuencia* con *probabilidad*, ya que una unidad frecuente puede ser poco probable en un determinado contexto, y viceversa.

Las probabilidades desiguales de diferentes unidades en un contexto dado es un fenómeno típico de las lenguas naturales. Supongamos que la probabilidad de dos unidades, X e Y, es X = 0,70 e Y = 0,30. Ninguna de las dos será totalmente redundante (probabilidad 1), pero la omisión de X (el elemento más probable en ese contexto) tendrá menos consecuencias que la omisión de Y. Se da, por tanto, la siguiente relación: cuanto más probable es una unidad, tanto mayor es su grado de redundancia y menor su contenido informativo. Tanto el grado de redundancia como el grado de información se compensan en un sistema comunicativo eficiente, ya que la redundancia permite cierto margen de confianza en la transmisión del mensaje a pesar de las distorsiones causadas por el ruido en el canal, de tal manera que la pérdida parcial de la información se compense con la redundancia. Por otra parte, la necesidad de mantener una comunicación eficaz (como defiende Grice, por ejemplo) exige que el mensaje no esté cargado de información redundante y por lo tanto poco interesante para el receptor, lo que pone en peligro el mismo acto de comunicación.

CUADRO 5.5. Principios generales de la Teoría de la Información. Lyons (1968).

1. Toda comunicación se basa en la probabilidad de elección o selección a partir de una serie de alternativas: una unidad lingüística de cualquier nivel carece de significado en un contexto dado si es completamente previsible.
2. El contenido informativo depende inversamente de la probabilidad: cuanto más previsible es una unidad, menos significado transmite.
3. La redundancia de una unidad lingüística es el margen que existe entre su contenido informativo y las distinciones que se requieren para identificarla. Es esencial cierto grado de redundancia para compensar el ruido.
4. La Ley de Zipf expresa la correlación entre extensión y frecuencia: una lengua será más eficaz si la extensión sintagmática de sus unidades está inversamente relacionada con la probabilidad de aparición de éstas. Esto se refleja en el hecho de que las palabras y las expresiones más utilizadas tienden a ser más cortas.

Se termina esta sección con un comentario histórico. La primera época de las investigaciones sobre la naturaleza estadística e informativa del lenguaje humano se desarrolló en los años cincuenta y principios de los sesenta, con autores como Herdan o Guiraud. Se centraron sobre todo en el campo léxico y fonológico. La implantación del paradigma racionalista lógico-simbólico de Chomsky desde los años sesenta impidió el desarrollo de esta aproximación estadístico-informativa. A modo de resumen, se mencionan algunas de las principales hipótesis desarrolladas por Herdan (1964):

1. La lingüística estadística es una cuantificación de la teoría de Saussure, donde la relación lengua-habla es equivalente a la de población estadística-muestra. Es decir, cualquier expresión lingüística individual actúa de muestra de la lengua (conjunto de todas las expresiones de una comunidad lingüística).
2. El uso lingüístico es algo más que el mero inventario de formas, pues hay que añadir la probabilidad de aparición de cada unidad en el habla.
3. En el lenguaje, como en el resto de los fenómenos del Universo, se dan leyes deterministas pero también azar. Por tanto, es pertinente la aplicación de métodos probabilísticos para dar cuenta de los fenómenos aleatorios.
4. El aspecto práctico de las regularidades estadísticas (por ejemplo, escoger entre varios sinónimos en un contexto dado) es muy relevante en áreas aplicadas, como la enseñanza de lenguas, la lingüística computacional o la sociolingüística, ya que nos permite hacer predicciones, saber lo que podemos esperar en circunstancias normales y, en consecuencia, tomar decisiones prácticas en función de las preferencias/opciones más probables.
5. Axiomas de Herdan:

   — Axioma 1: *la independencia del sonido y del significado.* Este axioma no es nada novedoso, pero tiene especial importancia en su aplicación estadística a la fonética. Se considera que el sistema fonético de una lengua es un universo estadístico en el que la distribución de los fonemas presenta una estabilidad constante.
   — Axioma 2: *la independencia entre la probabilidad de aparición de una unidad y su significado.* La distribución de las unidades léxicas en cualquier combinación depende sólo de las frecuencias de aparición y esta distribución es bastante imparcial con respecto al significado de las unidades.
   — Axioma 3: *la independencia del orden de palabras (sintaxis) con respecto al contenido de la expresión.* Es decir, las categorías sintácticas pueden aparecer en cualquier orden dentro de la oración. No existe ninguna posición determinada donde deba aparecer un significado concreto, por ejemplo, el agente de la acción. Esto contrasta con las reglas gramaticales tradicionales que insisten en un orden canónico de los constituyentes. El uso real demuestra que estas reglas están muy lejos de reflejar la realidad lingüística.

Estos axiomas reflejan el pensamiento muy extendido en el estructuralismo de que la semántica, el significado, debe estudiarse de una forma com-

pletamente separada del estudio del significante para hacer un análisis estrictamente formal de este último. El hecho de que se puedan aplicar parámetros estadísticos en el nivel léxico y sintáctico con resultados comprobables. (axiomas 2 y 3) implica que su estudio es relevante desde el punto de vista lingüístico, y que forman parte de las propiedades de la codificación lingüística.

5.4. Métodos estadísticos en LC

En esta sección se presentarán los modelos estadísticos más extendidos en PLN. Es ahora el momento de regresar a la exposición interrumpida en la sección 5.3.1 sobre conceptos de Teoría de la Probabilidad.

La probabilidad es una estimación sobre la posibilidad de que determinado suceso ocurra. Esta medida se cuantifica mediante un número entre 0 y 1, siendo 1 la certeza absoluta de que el suceso se producirá y 0 la certeza de que no se producirá el suceso. Cualquier número entre 0 y 1 indicará cierto margen de incertidumbre, que se utiliza para entender los fenómenos que no son deterministas (cuadro 5.6). Como se ha visto, esto es lo que supone el mayor atractivo de esta teoría.

CUADRO 5.6. Categorías de fenómenos.

— *Fenómenos deterministas:* aquellos de los que poseemos un grado de conocimiento tal que nos permite realizar predicciones exactas sobre los aspectos de su desarrollo que consideramos relevantes. Por ejemplo, los fenómenos astronómicos.
— *Fenómenos totalmente imprevisibles:* los conocimientos que tenemos son tan escasos que nada podemos predecir sobre ellos.
— *Fenómenos aleatorios:* ocupan un lugar intermedio entre los deterministas y los totalmente imprevisibles y se caracterizan porque, pese a ser totalmente imprevisibles de manera aislada, presentan regularidades estadísticas cuando se repiten un número elevado de veces. Estos son los fenómenos que trata la Teoría de la Probabilidad.

Nota: La concepción de un fenómeno como determinista o como aleatorio depende de nuestro conocimiento sobre el mismo. Por ejemplo, si conociéramos todos los condicionamientos que afectan a una persona a la hora de acometer un acto concreto, como dar su voto o emitir un determinado mensaje lingüístico, podríamos calcular su realización concreta. Por el contrario, al desconocer las condiciones iniciales, podemos recurrir a la Teoría de la Probabilidad para estudiar los resultados posibles, asignarles una probabilidad y realizar una predicción conociendo su margen de error.

La probabilidad se suele definir en términos de una *variable aleatoria* cuyos valores posibles se establecen de entre los de un conjunto predefinido. Por ejemplo, la variable FONEMA aplicada al español sólo tendrá como valores posibles los 24 fonemas del sistema fonológico del castellano. Así, si tomamos como referencia los datos de frecuencia de los fonemas, calculados por Alarcos en su *Fonología Española*, podemos decir que la probabilidad de que al tomar un fonema al azar en cualquier emisión del castellano salga /a/ es 0,137. Dicho más formalmente, P(FONEMA = /a/) = 0,137, o simplemente P(/a/) = 0,137.

5.4.1. Probabilidad condicionada e independencia de sucesos

La probabilidad se define como una función P que asigna un número a cada valor posible de la variable aleatoria. Es bastante habitual que, para un determinado fenómeno, queramos establecer diferentes variables y comprobar si hay relación entre ellas y, en caso positivo, en qué grado están relacionadas.

En el ejemplo anterior hemos estimado la probabilidad de que ocurra un suceso aislado (la de que aparezca el fonema /a/ en una elección al azar). Supongamos que tenemos acceso a información adicional. Por ejemplo, sabemos que el fonema anterior es /n/. En teoría, como hay cinco vocales en español, la probabilidad de cada una sería 1/5 o 0,20 (sin contar con las consonantes). Pero en ninguna lengua las frecuencias de los fonemas son las mismas para cada uno de ellos. ¿Qué ocurrirá si tomamos las frecuencias medias para estimar la probabilidad?

Intuitivamente nos imaginamos que la probabilidad de la combinación /na/ es diferente a la de /ne/ o /an/, pero ¿cómo lo cuantificamos? La definición de la *probabilidad condicionada* de que ocurra un suceso B dado un suceso A es la siguiente:

$$P(B \mid A) = P(A \& B) / P(A)$$

donde P (A & B) es la probabilidad de que los dos sucesos A y B ocurran simultáneamente. ¿Cómo se calcula esta probabilidad de aparición conjunta de A y B? El método más sencillo y fiable es contar en nuestra muestra las veces que aparecen las combinaciones /na/ y /an/, a los que llamaremos *bigramas* (veáse más adelante).

De momento, como no disponemos de dicha información, intentaremos otra aproximación (aunque se verá que es un camino equivocado): sustituiremos en la definición P (A & B) por P(A) * P(B). Es decir, se va a comprobar si la probabilidad conjunta de A y B es igual a la probabilidad independiente de A por la de B.

Para calcular la probabilidad de /n/ dado /a/ (es decir, la combinación /an/), sustituimos A por /a/ y B por el fonema /n/. La definición quedaría así:

$$P(/n/ \mid /a/) = P(/a/ \& /n/) / P(/a/) =$$
$$0{,}137 * 0{,}027 / 0{,}137 = 0{,}027$$

La probabilidad de que aparezca /n/ condicionada a que aparezca /a/ es igual a la probabilidad de que aparezca /n/. En este caso, se dice que la probabilidad de ambos sucesos es *independiente* de la aparición de cada uno. Efectivamente, podemos comprobar que la probabilidad de /a/ dado /n/ (la combinación /na/) es igual a la probabilidad de /a/:

$$P(/a/ \mid /n/) = P(/n/ \& /a/) / P(/n/) =$$
$$0{,}027 * 0{,}137 / 0{,}027 = 0{,}137$$

Se dice que dos sucesos son *independientes* si la aparición de uno no afecta a la probabilidad de aparición del otro. Dicho de manera más formal, dos sucesos A y B son independientes uno del otro si y sólo si

$$P(A \mid B) = P(A)$$

Puede que el lector se encuentre sorprendido por el resultado del ejemplo, pero en realidad la razón es muy simple: no se ha utilizado el recuento de la aparición consecutiva de esos dos fonemas. La probabilidad que se ha estado manejando es la media *a priori* sobre todas las posiciones, y es bien sabido que los fonemas tienen distintas probabilidades según su posición en la palabra o en la sílaba: la /b/ es bastante más probable en posición inicial que en posición final de sílaba, por ejemplo. Dado que la probabilidad condicionada nos informa de la relación de dos unidades (es decir, su independencia o su condicionamiento), si utilizamos la probabilidad media estamos suponiendo la independencia estadística entre ambas unidades, ya que no estamos considerando ningún contexto concreto sino cualquier posición en la que puedan aparecer. Corolario: no podemos utilizar las frecuencias medias de los fonemas para calcular probabilidades condicionadas ya que estas últimas dependen de un del recuento de bigramas.

¿Cómo podemos saber entonces la probabilidad de una unidad si conocemos la unidad que ha aparecido inmediatamente antes? Lo que hay que hacer es un recuento estadístico de las veces que aparece la unidad en cuestión detrás de la otra en una muestra representativa. Afortunadamente, para el español tenemos los datos sobre bigramas recogidos por Alameda y Cuetos (1995). La muestra analizada contiene 2.000.000 de palabras, 4.616.502 bigramas y 9.233.004 caracteres.

Consultamos en las tablas de bigramas los datos para /an/ y /na/. No utilizaremos el conjunto total de apariciones, sino el recuento en un contexto determinado, por ejemplo, principio de palabra. Los datos serían:

/an/: 8.880 apariciones en principio de palabra
/na/: 6.481

Para calcular su probabilidad, dividimos por el número total de bigramas en posición inicial de palabra, 2.000.000:

P(/an/): 8.880 / 2.000.000 = 0,0044
P(/na/): 6.481 / 2.000.000 = 0,0032

Ahora procedemos a hacer lo mismo con los fonemas /a/ y /n/ en posición inicial de palabra. Desgraciadamente, Alameda y Cuetos (1995) no proporcionan el recuento de las unidades fonológicas en distintos contextos, sino que dan únicamente el recuento total de cada unidad, es decir, su frecuencia media. Como sabemos, la frecuencia media no nos sirve para calcular la probabilidad condicionada en el contexto inicial de palabra. De hecho, la probabilidad de encontrar /a/ o /n/ en posición inicial de palabra puede variar significativamente con respecto a la frecuencia media de ambos fonemas. Para terminar el ejemplo, recurriremos a una *apreciación subjetiva* de la probabilidad de ambas unidades en principio de palabra. Téngase en cuenta, por tanto, que la estimación resultante no tiene ninguna validez empírica, aunque sí pedagógica. Dado que la frecuencia relativa de /a/ y /n/ en la mencionada muestra es de 12,85 y 6,97 respectivamente, pero que tal diferencia no es tan acusada en los bigramas /an/ y /na/ en posición inicial, supondremos que:

P (/a/) en posición inicial de palabra = 0,09
P (/n/) en posición inicial de palabra = 0,07

Con todos los datos, aplicamos ahora la probabilidad condicionada:

$P(B \mid A) = P(A \& B) / P(A)$
$P(/n/ \mid /a/) = P(/an/) / P(/a/) = 0{,}0044 / 0{,}09 = 0{,}048$
$P(/a/ \mid /n/) = P(/na/) / P(/n/) = 0{,}0032 / 0{,}07 = 0{,}045$

Por tanto, según estas estimaciones apenas hay diferencia en este contexto, aunque puede variar considerablemente en otros: Alameda y Cuetos (1995) contabilizaron 26.805 apariciones de /an/ en posición final y 32.523 apariciones de /na/.

Recapitulemos sobre las características de la probabilidad condicionada y la independencia de sucesos:

1. Si utilizamos frecuencias medias, es decir, recuentos de aparición de las unidades y su proporción sobre el total de la muestra, lo que obten-

dremos será una estimación de la probabilidad de aparición para cada unidad de manera aislada. Asumir la independencia entre las unidades es incorrecto en la mayoría de los casos.

2. Es mucho más interesante calcular la probabilidad de una unidad con respecto a otra en un contexto concreto. La clave de la probabilidad condicionada está en conocer el valor de P(A & B) y para ello la única manera es tener el recuento de las veces que aparecen en el contexto que nos interesa. Por otra parte, el condicionamiento contextual puede ser positivo o negativo:

   — A está condicionada positivamente por la aparición de B en un contexto determinado si la probabilidad de A, dado B, es mayor en ese contexto que la probabilidad de A. Es decir,

$$P(A|B) > P(A)$$

   — A está condicionada negativamente por la aparición de B en un contexto concreto si su probabilidad es menor que la probabilidad no condicionada de A. Es decir,

$$P(A|B) < P(A)$$

Hay dos casos extremos: el de máximo condicionamiento positivo, que se produce cuando hay una redundancia total (B presupone A), y el de máximo condicionamiento negativo, cuando se da la imposibilidad de que aparezcan ambas en ese contexto (B excluye A).

Por otra parte, hay que tener en cuenta que P(A|B) no es necesariamente igual a P(B|A); es más, lo habitual es que sean diferentes. No hay la misma probabilidad de que aparezca un adjetivo si ya ha aparecido un sustantivo la de que aparezca un sustantivo condicionada por la aparición de un adjetivo. En lenguas como el español, el adjetivo pospuesto al nombre es mucho más frecuente. (Por supuesto, en otras lenguas es a la inversa, e incluso en lenguas de orden estricto, una de las dos posiciones, antepuesta o pospuesta, representa una violación gramatical.) Como destaca Lyons (1968: 94):

> Por supuesto, es posible en principio calcular la probabilidad condicionada de cualquier unidad en relación con cualquier contexto. Lo importante es elegir el contexto y la dirección de condicionamiento [es decir, P(A|B) o P(B|A)] a la luz de lo que ya se conoce de la estructura sintagmática general de la lengua[...] Los resultados tendrán un interés estadístico si, y sólo si, P(A|B) o P(B|A) son notablemente diferentes respecto de P(A) y P(B).

5.4.2. Técnicas básicas: estimación y evaluación de probabilidades

Se tratará en este apartado cómo se calculan probabilidades. Naturalmente, en el caso ideal, si se tienen todos los datos relevantes para un problema, se pueden computar probabilidades exactas para tales datos. Supongamos que tenemos recogidos todos los textos de un autor. Podemos computar las frecuencias de las palabras que aparecen y deducir todas las probabilidades, medias y condicionadas. De esta manera, sabremos con seguridad la probabilidad de cada unidad en los textos del autor. Otro ejemplo similar sería preguntar a todos los votantes por su voto: la probabilidad de acertar el resultado es completa (siempre que los votantes no hayan mentido...).

Pero este uso de la probabilidad tiene escasa utilidad en la vida real; lo *que queremos es predecir acontecimientos a partir de cierta información* incompleta, ya sea el resultado de unas elecciones a partir de encuestas de opinión, ya sea el análisis más probable de una oración en un texto a partir de análisis anteriores.

El método de estimación más sencillo es el que emplea frecuencias relativas extraídas de un conjunto de datos. Aplicaremos ahora esta técnica a un corpus lingüístico, por etapas:

1. *Recogida de datos:* lo primero es contar con una muestra significativa. Un corpus es una colección de datos lingüísticos, normalmente formado por varios textos. (Sobre cómo elaborar un corpus se pueden consultar diversas fuentes, por ejemplo, Marcos Marín, 1994.)
2. *Anotación de las unidades del corpus:* para que se puedan inferir estadísticas no superficiales (es decir, no solamente cuántas veces aparece la palabra *sobre* en el corpus, etc.) el texto tiene que estar marcado con información. Las marcas más habituales son morfosintácticas: para cada unidad se especifica su categoría y, opcionalmente, otros atributos como la concordancia. Por ejemplo, *sobre* recibiría tres marcas en el caso de un etiquetado puramente categorial: una como preposición, otra como nombre y otra como verbo. Pero con un etiquetado morfosintáctico podría recibir hasta cuatro marcas diferentes, si consideramos que tiene dos formas con la categoría verbo, primera y tercera persona del singular de subjuntivo de *sobrar* (como en *yo sobre... él/ella/ello sobre*). Por supuesto, se puede anotar el texto con cualquier tipo de información pertinente, como por ejemplo sintagmática (describiendo la estructura interna de la oración en constituyentes no terminales) o semántico-léxica (mediante el marcado de categorías conceptuales). La anotación puede hacerse manual o automáticamente. Lo habitual es que los primeros datos se marquen por

especialistas, ya que pueden decidir en los casos de ambigüedad. Se suele asumir que el corpus anotado por los especialistas está libre de errores y se suele tomar como modelo para el entrenamiento de anotadores automáticos (de los que se hablará más adelante). Otra manera de anotar un corpus es utilizar una gramática computacional. Lógicamente, el anotado automático es menos fiable, pero tiene la ventaja de que es mucho más rápido. También se utilizan técnicas mixtas de marcación: un primer tratamiento se hace automáticamente y después los codificadores revisan y corrigen. Éste es el método más utilizado actualmente (por ejemplo, el Penn Tree Bank) porque se beneficia de las ventajas de los métodos manuales y automáticos.

3. *Cálculo de frecuencias de las unidades:* se hace un recuento de cuántas veces aparece cada unidad (ya sea palabra, sintagma, concepto, morfema, fonema, etc.) en función de los tipos posibles que hayamos definido de antemano. De esta manera, por ejemplo, se calcula la probabilidad asociada de cada palabra con una determinada categoría sintáctica.

Supongamos que hemos recogido un corpus pequeño que contiene 10.255 palabras. Hemos etiquetado cada una con su categoría sintáctica y hemos obtenido el siguiente recuento (el ejemplo no tiene base empírica real, pero a todos los efectos es ilustrativo):

sobre, PREP	115 veces
sobre, N	13 veces
sobre, V	11 veces
TOTAL	139 veces

Las probabilidades se calcularán mediante la división entre el número de veces que aparece una palabra con una categoría determinada y el número total de palabras. P *(sobre)* representa la probabilidad media, es decir, la probabilidad de que una palabra elegida aleatoriamente del corpus sea *sobre*. Los otros casos son las probabilidades conjuntas – P(A & B)– de que sea la palabra *sobre* y una de las tres categorías:

P *(sobre)*	139/10.255 = 0,0135
P(*sobre* & PREP)	115/10.255 = 0,0112
P *(sobre* & N)	13/10.255 = 0,0012
P *(sobre* & V)	11/10.255 = 0,0010

Con estos cálculos hemos obtenido la probabilidad de ocurrencia de una palabra con relación a toda la muestra. Un dato más interesante es pronosti-

car cuál de las tres categorías (PREP, N o V) es más probable que aparezca si la palabra escogida es *sobre*. Ésta es la tarea a la que se enfrenta un programa que analice categorialmente el corpus. Para ello, se utiliza la definición de la probabilidad condicionada:

$$P(A \mid B) = P(A \& B) / P(B)$$

$$\begin{aligned} P(\text{PREP} \mid \textit{sobre}) &= P(\textit{sobre} \& \text{PREP}) / P(\textit{sobre}) = \\ &= 0{,}0112 / 0{,}0135 = 0{,}829 \\ P(\text{N} \mid \textit{sobre}) &= P(\textit{sobre} \& \text{N}) / P(\textit{sobre}) = \\ &= 0{,}0012 / 0{,}0135 = 0{,}088 \\ P(\text{V} \mid \textit{sobre}) &= P(\textit{sobre} \& \text{V}) / P(\textit{sobre}) = \\ &= 0{,}0010 / 0{,}0135 = 0{,}074 \end{aligned}$$

Nótese que es diferente la probabilidad conjunta, P (A & B), de la probabilidad condicionada, P (B | A). La primera nos proporciona el contexto, y sin ese dato no podemos estimar la segunda.

CUADRO 5.7. Estimación de probabilidades.

Dos fases:

— Recuento de ocurrencias

— Aplicación de alguna *técnica estadística* (probabilidad condicionada, Ley de Bayes, n-gramas, árboles de decisión, etc.).

Con la definición de la probabilidad condicionada aplicada al corpus del ejemplo se puede predecir que la próxima ocurrencia de *sobre* tiene más del 80% de posibilidades de ser una preposición. Esto se conoce por *estimación de la máxima verosimilitud* o *probabilidad*. Obviamente, la exactitud de nuestra predicción dependerá de la cantidad de datos que hayamos utilizado para calcularla: cuantos más datos más fiable (aunque habrá ocasión de comprobar que esta "ley de los números grandes" no funciona con otros métodos de estimación más sofisticados).

Por otra parte, una estimación es poco o nada fiable cuando se ha utilizado una muestra pequeña. La cuestión es decidir cuándo el tamaño de la muestra es aceptable o no, es decir, determinar un *margen de error aceptable*.

Kubácek (1994) propone una fórmula para estimar el tamaño de muestra necesario para que pueda ser considerado representativo. La fórmula tiene en cuenta tres variables: N que es el tamaño de muestra necesario (es

decir, el número de unidades que tiene que tener como mínimo el corpus); *k* es el número de unidades consideradas (por ejemplo, fonemas o categorías sintácticas) y *r* es la desviación estándar media de las frecuencias relativas. Lo interesante de esta fórmula es que permite calcular N en función de las otras dos variables, que son fáciles de determinar. La fiabilidad de la muestra depende, en esta definición, de la desviación estándar media escogida. Las características de este libro no nos permiten entrar en las complejidades de la definición de Kubácek, pero daremos los resultados aplicados a un corpus con 13 tipos de unidades diferentes:

$k = 13$				
r	0,01	0,005	0,001	0,0005
N	378	1.513	37.817	151.268

Otro problema de estimación son los datos con muy baja frecuencia o incluso que no hayan aparecido nunca en el corpus (por lo tanto su probabilidad en el modelo es 0). Representan un serio problema para el grado de fiabilidad de la estimación. Veremos una propuesta de solución en el siguiente apartado.

Es importante tener en cuenta que la estadística es una aproximación, por tanto cabe siempre la posibilidad de que una estimación sea incorrecta. Una vez que tenemos un conjunto de probabilidades y alguna técnica estadística para aplicarlos sobre datos similares, ¿cómo podemos evaluar la calidad de nuestras estimaciones de manera que podamos compararlas con otras conseguidas por técnicas diferentes?

El *método general de evaluación* es el siguiente:

1. Antes de proceder al cálculo de probabilidades se divide el corpus en dos partes: el *conjunto de entrenamiento* y el *conjunto de prueba*. El conjunto de entrenamiento suele ser una parte pequeña del conjunto total de datos, aunque, por supuesto, depende del tamaño del corpus. En los casos en que se disponga de un marcado (total o parcial) del texto, el conjunto de entrenamiento es necesariamente una parte anotada manualmente. El conjunto de entrenamiento se utiliza para calcular las probabilidades, como hemos visto en el ejemplo de *sobre*.
2. El conjunto de prueba representa una muestra de lo que el sistema puede encontrarse en situaciones reales. Normalmente, suele tener un tamaño equivalente al de entrenamiento, pero depende de distintos factores. Los datos del corpus de prueba son proporcionados como entrada al sistema, y el resultado generado se evalúa. La manera de realizar la evaluación depende de las diferentes aproximaciones y de la disponibilidad de recursos. Una bastante habitual, por fiable, es

comparar el resultado con el obtenido por seres humanos. En este caso se pide a especialistas que analicen el mismo conjunto de prueba y den sus respuestas. Una variante de este método consiste en utilizar un corpus etiquetado o anotado: al sistema se le ofrece un conjunto de prueba no anotado y su resultado se compara con el mismo conjunto de prueba pero en la versión anotada. La investigación sobre corpus está especialmente interesada por la disponibilidad de estos corpus anotados, tanto para entrenamiento como para evaluación. En la actualidad hay un buen número de proyectos encaminados a desarrollar estos recursos en diferentes lenguas. En español podemos destacar el proyecto CREA (Corpus de Referencia del Español Actual), realizado por la RAE.

3. En función de los resultados y de los medios disponibles, se puede repetir el procedimiento escogiendo otros conjuntos de entrenamiento y prueba diferentes. Un método de evaluación bastante más fiable y refinado es la validación cruzada *(cross-validation)*, que consiste en tomar diferentes partes del corpus como conjunto de prueba y entrenar sobre el resto. Al tomar distintos fragmentos se reduce la posibilidad de que el conjunto de prueba sea de alguna manera más sencillo, por parecido, al conjunto de entrenamiento. No olvidemos que lo que interesa es medir la respuesta del modelo en casos reales, por tanto la introducción de algún tipo de aleatoriedad redundará en fiabilidad.

Se ha visto en el ejemplo de *sobre* el método más sencillo de calcular probabilidades condicionadas. Para conseguir resultados más fiables tendremos que recurrir a un método más sofisticado que dé cuenta de más contexto. Estamos ahora en situación de introducirnos en la siguiente técnica estadística, los n-gramas.

5.4.3. Modelo de N-gramas

Es de lejos el modelo más empleado y el que mejores resultados ha obtenido. Permite mejorar la fiabilidad de la estimación teniendo en cuenta parte del contexto local. El modelo surge de la asunción de que sólo unas pocas unidades anteriores condicionan la probabilidad de aparición de la siguiente unidad. Se denomina *n-grama*, donde la n representa el número de unidades que se tienen en cuenta, contando la que se reconoce. Para casi todos los sistemas $2 \leq n \geq 7$, siendo $n = 2$ el bigrama, $n = 3$ el trigrama, etc. Los bigramas pueden ser modelos demasiado pequeños para algunas aplicaciones, ya que sólo tienen en cuenta la unidad anterior, mientras que los sexagramas o septagramas son modelos que se han experimentado muy recien-

temente. El modelo más común es el trigrama (Charniak, 1993). Así, el trigrama calcula la probabilidad condicionada de una unidad (típicamente una palabra) dadas dos unidades precedentes; esto es:

$$P(U_i \mid U_{i-2}\, U_{i-1})$$

Podemos sustituir U por cualquier unidad que nos interese (fonemas, morfemas, palabras, categorías sintácticas, etc.). Para crear un trigrama lo que tenemos que hacer es utilizar un corpus de entrenamiento y registrar cada una de las parejas y tríos de palabras (o cualquier otra unidad) que aparecen en el texto, así como el recuento de las veces que aparecen. La manera de hacer esto, naturalmente, es utilizando algún tipo de programa sencillo, pero como ilustración lo haremos "a mano" con el fragmento de Martin Kay que se citó en el Prólogo:

> Digámoslo claramente, no es fácil hablar con los ordenadores y a veces es más fácil utilizar un intérprete. Es económico y efectivo, tanto para el hombre como para el ordenador, hablar lenguas diferentes e interactuar a través de un intermediario.

Los signos de puntuación se consideran también "palabras", en el sentido gráfico de "cualquier unidad aislada entre blancos que aparezca dentro de los límites de la oración". No se hará distinción entre mayúsculas y minúsculas, para simplificar. Empezando por la primera oración del texto, haríamos una lista de parejas y tríos, con recuento de la aparición de cada uno. (Los espacios en blanco entre comillas representan inicio de texto. Su recuento es pertinente para saber la probabilidad de una unidad al principio del texto.)

Parejas	*Recuento*	*Tríos*	*Recuento*
(" ", " ")	1	(" ", " ", digámoslo)	1
(" ", digámoslo)	1	(" ", digámoslo, claramente)	1
(digámoslo, claramente)	1	(digámoslo, claramente, ",")	1
(claramente, ",")	1	(claramente, "," , no)	1
(",", no)	1	(",", no, es)	1
(no, es)	1	(no, es, fácil)	1
...		...	
(para, el)	2	(para, el, hombre)	1
...		(para, el, ordenador)	1
...			

En el texto de entrenamiento, al ser tan pequeño, sólo hay un pareja que aparece en más de una ocasión: (para, el). Obviamente, al aumentar el tamaño del corpus, surgen más repeticiones de parejas y tríos. Podemos ahora estimar la probabilidad de la palabra, pal_i, condicionada por sus dos palabras anteriores, pal_{i-2} y pal_{i-1}, mediante la siguiente fórmula:

$$P(pal_i \mid pal_{i-2}, pal_{i-1}) = \frac{\text{Recuento}(pal_{i-2}, pal_{i-1}, pal_i)}{\text{Recuento}(pal_{i-2}, pal_{i-1})}$$

Por tanto, para calcular la probabilidad estimada de que aparezca "hombre" detrás de "para el" tenemos que contar las veces que aparece "para el hombre" (en la fórmula: "Recuento(pal_{i-2}, pal_{i-1}, pal_i)") y dividirlo por el número de ocurrencias de "para el". El resultado en este caso sería 1/2 = 0,5. Si el corpus fuera representativo podríamos predecir que cada vez que se da la combinación "para el" hay un 50% de posibilidades de que la siguiente palabra sea "hombre" (el otro 50% sería "ordenador"). Como ya sabemos, nuestra estimación será más fiable a medida que aumentemos el tamaño de los datos de entrenamiento.

Si tomamos como base del recuento las palabras sólo conseguiremos un corpus representativo para un número limitado de aplicaciones. Por ejemplo, supongamos que estamos desarrollando un sistema de reconocimiento de voz que sólo trata un léxico reducido de palabras, bien porque el dominio está muy acotado o porque sólo reconoce las palabras para las que ha sido entrenado. Como conocemos de antemano las palabras que pueden aparecer, podemos entrenar el modelo de manera que sea capaz de predecir, dada una palabra, su continuación más probable. Los n-gramas basados en palabras también tienen sentido si nuestra aplicación busca selectivamente significados a partir de palabras. Por ejemplo, podemos desarrollar un sistema de recuperación de información que busque determinado tipo de contenidos, fijados previamente. Conocidos los significados que nos interesan e identificadas las palabras que los representan, podemos entrenar nuestro modelo de manera que busque contexto anterior y posterior con respecto a las palabras pertinentes. Esto nos permitirá identificar los contextos temáticos relevantes en documentos nuevos.

Naturalmente, los n-gramas se pueden aplicar a muchos tipos de unidades, típicamente categorías sintácticas y fonemas. Éste es el caso de los etiquetadores (morfo)sintácticos y de los reconocedores de habla. En estas aplicaciones el objetivo es reconocer unidades más abstractas que las palabras. Por tanto, estos modelos estadísticos se construyen sobre conjuntos de unidades mucho más reducidos. Las palabras posibles del español no se pue-

den conocer con certeza, pero seguramente pasan de varios millones. En contraste, podemos elaborar etiquetarios *(tagsets)* de pocas decenas de categorías gramaticales.

En cualquier caso, el método es el mismo: nuestro recuento de bigramas y trigramas sobre palabras en el ejemplo anterior se puede convertir en un recuento sobre sus correspondientes categorías:

Parejas	*Recuento*	*Trios*	*Recuento*
(P-de-O, V)	1	(P-de-O, V, ADV)	1
(V, ADV)	1	(V, ADV, COMA)	1
(ADV, COMA)	1	(ADV, COMA, NEG)	1
(COMA, NEG)	1	(COMA, NEG, V)	1
...		...	
(PREP, ART)	2	(PREP, ART, N)	2
...		...	

Según esta nueva estimación, la probabilidad de que aparezca un N detrás de PREP ART es 2 / 2 = 1, es decir, el 100%. Al utilizar categorías en lugar de palabras concretas encontramos muchos más casos de combinaciones repetidas. El hecho de partir con un conjunto de unidades pequeño presenta la ventaja de que conseguir un corpus representativo es una tarea más asequible.

Si las probabilidades estimadas por este método se añaden a un autómata de estados finitos, entonces obtenemos un autómata probabilístico, también conocido como *cadenas de Markov* (Charniak, 1993). Los modelos markovianos asumen que las gramáticas de las lenguas naturales son de estados finitos, lo cual no es apropiado desde un punto de vista teórico, como se ha visto en los capítulos anteriores. Recuérdese, por ejemplo, que un n-grama no podría predecir la aparición de construcciones incrustadas correlativas como "si ... entonces...".

Sin embargo, a pesar de esas limitaciones teóricas reales (como ya se expuso en el capítulo anterior con el modelo en dos niveles para la morfología), los n-gramas tienen gran aplicación práctica en sistemas de reconocimiento de habla, básicamente por su eficiencia.

La razón fundamental por la que los modelos de n-gramas funcionan eficazmente es debido a que las restricciones locales en algunas lenguas naturales son muy fuertes. Como demuestran las investigaciones y resultados en la última década (Church y Mercer, 1993, ofrecen un amplio panorama), las

unidades lingüísticas parecen bastante condicionadas por el contexto, especialmente las restricciones de tipo semántico-léxico. Por ejemplo, las *colocaciones*, un término usado en Lingüística y especialmente en Lexicografía para aludir a emparejamientos habituales de palabras. Se ha constatado en numerosas lenguas que palabras con casi la misma sintaxis y semántica tienen contextos en los que una es más apropiada (o frecuente) que la otra. "Estupendo" y "admirable" son sinónimos ("algo digno de asombro y admiración"), pero es mucho más frecuente encontrar las parejas "cena estupenda" y "valor admirable" que "cena admirable" y "valor estupendo".

Significativamente, las gramáticas simbólicas son incapaces de capturar qué opción entre varias de la misma categoría es la más probable, precisamente porque todas las opciones cumplen los requisitos de buena formación sintáctica: tan gramatical es "valor estupendo" como "valor admirable". Y a la inversa, las gramáticas independientes del contexto pueden tratar cláusulas incrustadas y dependencias a larga distancia, lo que resulta impracticable para los modelos de n-gramas.

Como se decía más arriba, una *cadena de Markov* es una red que utiliza probabilidades. Desde una perspectiva más general, es un tipo de proceso estocástico, es decir, un conjunto finito de variables aleatorias encadenadas que tienen una probabilidad conjunta. Dicho de otra manera, el modelo markoviano combina la idea de la probabilidad condicionada (los n-gramas) con la noción de sucesos encadenados. La figura 5.1 representa gráficamente una cadena de Markov para un modelo imaginario de un fragmento reducidísimo de las oraciones del español (es imaginario porque sus probabilidades han sido inventadas).

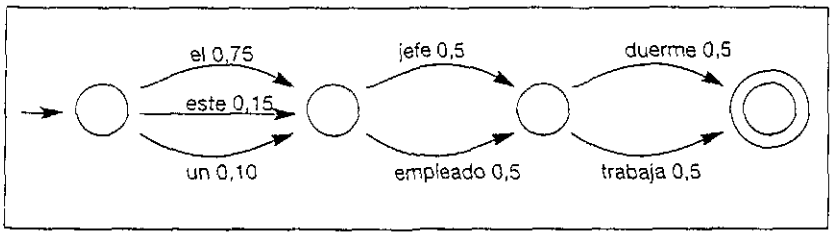


Figura 5.1. Una cadena de Markov muy sencilla aplicada al español.

Es un autómata con sólo cuatro estados. Los símbolos de cada arco son elementos terminales (es decir, palabras) que tienen asociado un valor de acuerdo con su probabilidad (estimada con algún método de recuento de los ya vistos en el ejemplo es inventada, pero ilustra igualmente la técnica).

Naturalmente la suma de las probabilidades de los arcos que salen de un estado concreto tiene que ser 1. Así por ejemplo, desde el estado inicial se pueden escoger tres arcos, cada uno con una probabilidad diferente. En los dos estados restantes sólo se puede escoger entre dos opciones, ambas con la misma probabilidad. Estos autómatas pueden funcionar como generadores o como reconocedores. En cualquier caso, la probabilidad de generar o aceptar una determinada cadena de elementos terminales con este autómata es simplemente el producto de las probabilidades de los arcos atravesados para generar o reconocer dicha cadena.

Las oraciones modeladas por el autómata, con su probabilidad asociada se muestran en el cuadro 5.8.

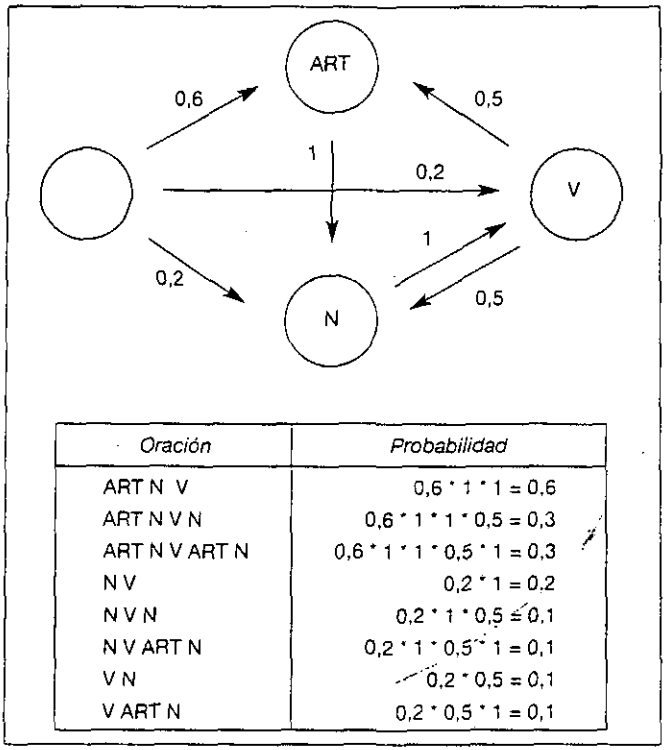
CUADRO 5.3. Oraciones procesadas con el autómata y su probabilidad.

Oración	*Probabilidad*
El jefe duerme	0,75 * 0,5 * 0,5 = 0,1875
El jefe trabaja	0,75 * 0,5 * 0,5 = 0,1875
El empleado duerme	0,75 * 0,5 * 0,5 = 0,1875
El empleado trabaja	0,75 * 0,5 * 0,5 = 0,1875
Este jefe duerme	0,15 * 0,5 * 0,5 = 0,0375
Este jefe trabaja	0,15 * 0,5 * 0,5 = 0,0375
Este empleado duerme	0,15 * 0,5 * 0,5 = 0,0375
Este empleado trabaja	0,15 * 0,5 * 0,5 = 0,0375
Un jefe duerme	0,10 * 0,5 * 0,5 = 0,0250
Un jefe trabaja	0,10 * 0,5 * 0,5 = 0,0250
Un empleado duerme	0,10 * 0,5 * 0,5 = 0,0250
Un empleado trabaja	0,10 * 0,5 * 0,5 = 0,0250

Según el autómata que hemos diseñado, la probabilidad de cada una de las cuatro primeras oraciones (las que empiezan con el artículo definido "el") es cercana al 20%, y mucho menor en las ocho oraciones restantes. Obviamente, este modelo es extremadamente simplificado, pero nos proporciona el ejemplo más sencillo de cómo funcionan los modelos estocásticos. En él se han empleado bigramas para reflejar las probabilidades de transición de un estado a otro. Dicho de otra manera, la probabilidad condicionada de "el" con respecto al estado inicial es 0,75 –o en fórmula, P (*el* | estado inicial) = 0,75; la probabilidad de "jefe" dado "el" es 0,5, P(jefe | el) = 0,5; etc.–.

Análogamente, podemos construir una cadena de Markov para categorías sintácticas, de manera que podamos determinar la probabilidad de una

determinada secuencia de categorías simplemente multiplicando las probabilidades de transición. La figura 5.2 y su correspondiente tabla muestran un ejemplo, con probabilidades inventadas, con sólo tres categorías sintácticas, ART, N y V.



Oración	*Probabilidad*
ART N V	0,6 * 1 * 1 = 0,6
ART N V N	0,6 * 1 * 1 * 0,5 = 0,3
ART N V ART N	0,6 * 1 * 1 * 0,5 * 1 = 0,3
N V	0,2 * 1 = 0,2
N V N	0,2 * 1 * 0,5 = 0,1
N V ART N	0,2 * 1 * 0,5 * 1 = 0,1
V N	0,2 * 0,5 = 0,1
V ART N	0,2 * 0,5 * 1 = 0,1

Figura 5.2. Cadena de Markov con categorías sintácticas.

La cadena de Markov de la figura 5.2. representa bigramas, es decir, la probabilidad de una categoría con respecto a la precedente. Si queremos incorporar la probabilidad de una determinada transición teniendo en cuenta los dos últimos estados recorridos, entonces necesitamos trigramas encadenados, y un modelo más sofisticado conocido como *modelo de Markov oculto*, que se tratará brevemente en el apartado de etiquetadores probabilísticos.

La técnica de los n-gramas tiene problemas con el tratamiento de los datos no encontrados en el corpus de entrenamiento *(sparse data* es el término inglés). No importa lo grande que sea el corpus de entrenamiento, siempre habrá algunas combinaciones nuevas en el conjunto de prueba para las que no se tienen computadas las probabilidades y, por tanto, su probabilidad es cero. Naturalmente, ésta es una conclusión falsa: el hecho de que una combinación gramatical no se haya registrado en un corpus no significa que no se pueda dar. Su probabilidad será baja, pero nunca cero. Desafortunadamente, es una situación frecuente e inevitable: en un experimento realizado sobre un corpus de entrenamiento de 1.500.000 palabras del inglés, y aplicado sobre un conjunto de prueba de solo 300.000 palabras, el 25 % de los trigramas del segundo texto, que era cinco veces más pequeño, no había aparecido en el de entrenamiento (Charniak, 1993). Una técnica habitual que se aplica en estos casos consiste en "suavizar" *(smooth)* las probabilidades mediante el cómputo adicional de bigramas y unigramas (Allen, 1995). La fórmula sería:

$$P(U_i \mid U_{i-2}\, U_{i-1}) \cong \lambda_1 P(U_i) + \lambda_2 P(U_i \mid U_{i-1}) + \lambda_3 P(U_i \mid U_{i-2}\, U_{i-1})$$

donde λ_1, λ_2, λ_3 son constantes cuya suma es igual a 1. Como asumimos que los trigramas dan mejor estimación que los bigramas y unigramas, λ_3 debería ser mayor que las otras dos constantes para que el trigrama dominara en el cálculo. Utilizando esta fórmula, cada vez que aparezca un trigrama con probabilidad cero, al menos obtendremos alguna estimación gracias a la probabilidad del bigrama y del unigrama.

Otros dos problemas de los n-gramas son:

a) El contexto es bastante limitado, sobre todo en el caso de bigramas y trigramas.
b) A veces se fragmenta el texto innecesariamente.

Por ejemplo, cuando hay palabras compuestas por más de un elemento: "San Sebastián" , "ojo de buey", "Impuesto sobre la Renta de la Personas Físicas", etc. Está claro que no se deben segmentar, pero un trigrama tendrá una distribución de la probabilidad para (ojo, de, buey) y otra para (ojo, de, insecto), cuando en el primer caso sólo hay un concepto léxico representado y en el segundo dos ("ojo" e "insecto") y una relación ("posesión", marcada por la preposición *de).*

En cuanto a la limitación al contexto más cercano, ésta es una de la características más sobresalientes del modelo, tanto en su vertiente positiva como en la negativa. Una solución es aumentar el n-grama, pero esto es muy costoso: para un tetragrama se calcula que se necesitaría, partiendo de un con-

junto de sólo 40 posibles categorías, el tamaño de una matriz de 2.560.000 estadísticas para obtener un modelo fiable (Allen, 1995).

Se acaba aquí esta presentación de técnicas probabilísticas. Por supuesto, al igual que los modelos simbólicos no se agotan con las gramáticas generativas, existen más modelos estadísticos que los aquí expuestos, especialmente el modelo del Canal de Ruido y el de cadenas de Markov ocultas, que se tratarán brevemente en el apartado siguiente de aplicaciones.

5.5. Aplicaciones

5.5.1. Reconocimiento de habla

Gran parte de la popularidad de los métodos estadísticos en LC se debe al consenso, dentro de la comunidad dedicada a los sistemas de habla, en favor de las aproximaciones empíricas sobre las basadas en la competencia y el conocimiento. Para un sistema de reconocimiento de habla el principal problema es la proyección entre fonemas y alófonos: de una cadena de sonidos, el sistema tiene que identificar cada realización alofónica con su correspondiente fonema. La razón por la que la tarea es bastante complicada es, como señalan Church y Mercer (1993), que la relación entre fonemas y sus alófonos es muy variable y ambigua. Es frecuente encontrar en las lenguas que dos fonemas comparten el mismo alófono en algunos contextos. Por ejemplo, en el dialecto meridional del castellano peninsular /s/ y /θ/ se pronuncian de la misma manera (/kása/ *casa* y *caza*). Sin embargo, los oyentes no suelen tener problema en reconocer el fonema apropiado en cada caso porque saben o se imaginan lo que el emisor va a decir. Así, una oración como "Me voy de casa / caza", se interpreta de una manera y otra en función del contexto en que se emita.

La primera aproximación a este problema consistió en desambiguar la información alofónica de la señal acústica mediante la comprensión del texto. Esto implicaba realizar un procesamiento completo del mensaje para poder desambiguar el fonema. Además, había que hacerlo en un tiempo razonable: una espera de más de cinco segundos no suele ser admisible para establecer una comunicación oral fluida entre un hombre y un ordenador, y los procesamientos de oraciones sencillas podían llevar minutos. A mediados de los años setenta, un grupo de investigación de IBM empezó a aplicar técnicas estadísticas sobre un banco de datos de señales acústicas. Dejando de lado las teorías fonéticas y computacionales del momento, replantearon el problema del reconocimiento acústico en términos de transmisión a través de un canal con ruido, teoría desarrollada por Shannon en 1948.

El modelo del *canal con ruido* pretende desarrollar un procedimiento automático que recupere la señal de entrada a partir de la señal corrupta después de haber pasado por un canal con ruido:

Señal de entrada → Canal con ruido → Señal corrupta

Para un sistema de reconocimiento de habla, la *señal corrupta* sería la emisión real con las distintas ambigüedades alofónicas que tiene que reconocer para recuperar la secuencia de fonemas correcta (es decir, la *señal de entrada* en el esquema). La formulación del problema sería: hay que determinar la señal de entrada, I, más probable de entre todas las señales posibles, O:

$$\arg\max P(I \mid O)$$

donde la función *arg max* selecciona el argumento con mayor puntuación. Esta fórmula se suele reescribir utilizando la ley de Bayes y da como resultado:

$$\arg\max P(I \mid O) = \arg\max P(O \mid I)\, P(I)$$

donde el problema de selección de la entrada I más probable se divide en dos partes:

— La *probabilidad del canal*, P (O | I), es la probabilidad condicionada de que aparezca determinada señal corrupta O cuando la señal de entrada es I. Por ejemplo, en determinados contextos la realización de la /p/ y la /b/ pueden sonar muy similares, especialmente en posición intervocálica: cepo / cebo. Se le llama *probabilidad del canal* porque depende de la aplicación. Así, la ambigüedad entre *cepo/cebo* lo sería para un reconocedor de habla, pero no para un reconocedor óptico de caracteres (en cambio éste tendría problemas de ambigüedad para reconocer O/0, el carácter para la "o" en mayúscula y el carácter para el número cero).
— La *probabilidad del modelo lingüístico*, P (I), es la probabilidad de que I sea la señal de entrada. Esta probabilidad no puede ser conocida directamente y hay que aplicar alguna técnica estadística, generalmente un trigrama. Para ello, como ya sabemos, hay que computar las distintas estadísticas de parejas y tríos sobre una muestra grande.

Aunque la mayoría de los sistemas de reconocimiento de habla utilizan modelos de Markov para computar las probabilidades condicionadas necesarias para el modelo de canal con ruido, hay otras técnicas disponibles, como el modelo de Árbol de Decisión.

Un *árbol de decisión* se organiza en torno a un conjunto de preguntas: ¿cuál es la palabra n-1? ¿Cuál es la palabra n-2? Etc. En cada pregunta se

almacena una serie de probabilidades, cuando se llega al final del árbol se retrocede hasta la raíz (es decir, el punto de partida) y se computa la combinación de probabilidades más probable.

El problema es cómo elegir la mejor pregunta en cada nodo del árbol para obtener la máxima eficacia: si la respuesta produce bastantes probabilidades, la incertidumbre en la siguiente pregunta se reduce porque se espera haber eliminado más candidatos, pero a cambio de ello el coste computacional es muy elevado. Por eso se suelen preferir "preguntas binarias", es decir, aquellas que sólo acepten "sí" o "no" como respuesta, ya que la computación se reduce.

El atractivo de estos modelos está en su posibilidad de ser entrenados automáticamente a partir de un corpus, y en su posibilidad de obtener información suministrada por una jerarquía de preguntas. Tienen problemas con la fragmentación de datos, pero hay ciertas técnicas que permiten tratar las unidades compuestas, lo que no pueden hacer los n-gramas.

¿Cuál es el estado de la cuestión a finales de los 90? Borthwick (1997), en un interesante informe técnico, señala que el sentimiento actual dentro de la comunidad de reconocimiento del habla es el de encontrarse en un punto muerto. En la parte de reconocimiento acústico (lo que hemos denominado probabilidad del canal), cada vez que aparece una buena idea se consigue una mejora significativa en la actuación del sistema. Sin embargo, los planteamientos nuevos en la parte de modelización lingüística apenas si mejoran algo la precisión. Borthwick apunta que una de las razones es que el modelo de n-gramas es tan rígido que es casi imposible mejorarlo de manera significativa. Y la mayoría de los sistemas siguen confiando en los n-gramas.

Otro de los problemas señalados por Borthwick es que buena parte de los errores cometidos por los actuales sistemas de reconocimiento de habla son de tipo sintáctico. Por ejemplo, la confusión entre el pronombre enclítico "la" y el artículo definido "la" en oraciones como "*damela* llave", en lugar de "dame la llave". En el estado actual, las técnicas estadísticas tratan bien las restricciones léxico-semánticas pero no las sintácticas.

Por último, Borthwick señala que, a pesar de las inmensas cantidades de texto disponibles para computar estadísticas (el *Wall Street Journal* en formato electrónico contiene más de 242 millones de palabras), los n-gramas son un método demasiado rudimentario (de 'fuerza bruta", es la expresión que utiliza). Las esperanzas están puestas en que surjan nuevas técnicas estadísticas para explotar esta enorme masa de datos.

5.5.2. Desambiguación léxica y sintáctica

La idea de incluir información estadística en las gramáticas simbólicas para tratar la ambigüedad es la aplicación más aceptada por todos. Las gra-

máticas simbólicas no pueden incluir toda la información necesaria para tratar dicho problema, como se vio en el anterior capítulo. La estimación de las regularidades estadísticas ayuda a elegir, de una manera fundamentada, la opción más probable. El resultado más claro de esta técnica es una mejora considerable en la eficacia del sistema.

Cualquier gramática computacional del español de cierta cobertura basada en la competencia produce un número muy grande de análisis sintácticos alternativos para la mayoría de las oraciones. Esto es debido, sobre todo, a las diferentes posibilidades de adjunción de sintagmas preposicionales, cláusulas de relativo y demás modificadores de núcleos nominales y verbales. Recuérdese el ejemplo de la sección 4.6.2:

Vi a un hombre en el parque con un telescopio.

Hay varias posibilidades de análisis, dependiendo de quién esté en el parque y con el telescopio (el emisor o el hombre). Una gramática simbólica debe dar cuenta de todas las posibles estructuras para oraciones de este tipo (y otras mucho más complicadas). Sin embargo, es significativo que los hablantes de español no noten tanta ambigüedad cuando procesan oraciones: de manera espontánea y natural sólo vienen a la mente dos o tres posibilidades. La explicación más frecuente de este hecho es que la interpretación ayuda a desambiguar sintácticamente, ya que los hablantes prefieren las interpretaciones plausibles a las poco probables. La plausibilidad de una interpretación concreta viene dada por el contexto semántico-pragmático. Esta explicación de las ambigüedades sintácticas es la que ha prevalecido en el modelo simbólico: la solución para reducir el número de análisis es incorporar restricciones semánticas y pragmáticas muy finas. Nótese que este método es inherentemente poco eficiente, ya que deja la resolución de las ambigüedades para los últimos niveles de análisis. Es decir, arrastra muchos análisis sintácticos que al final se tienen que descartar.

La solución que se propone desde los modelos basados en los datos y la actuación es utilizar un tipo de información diferente: la frecuencia de aparición de elementos léxicos y de estructuras sintácticas. Bod y Scha (1996) dan tres argumentos de tipo psicolingüístico:

1. Los hablantes registran frecuencias de uso y diferencias entre frecuencias.
2. Los hablantes prefieren los análisis que ya han experimentado a análisis que tienen que construir por primera vez.
3. La preferencia está influida por la frecuencia de aparición de los análisis, de manera que se prefieren los análisis más frecuentes a los menos frecuentes.

Esta postura implica una aproximación radicalmente diferente a la de los modelos simbólicos, ya que el procesamiento se basa en *preferencias* y no en conocimiento. Church y Mercer (1993) sugieren que algunas regularidades estadísticas, por ejemplo la de que las palabras de mucha frecuencia de uso, tienden a ser más breves y más predecibles (la Ley de Zipf), pueden ser el resultado de la evolución natural del habla humana en su búsqueda de comunicación fiable en presencia de ruido.

Se analizarán brevemente un ejemplo de aplicación de la idea de las preferencias léxicas a la asignación de sintagmas preposicionales. Hindle y Rooth (1993) proponen utilizar estadísticas de coaparición de verbo-preposición y nombre-preposición. Es una aproximación de clara inspiración lexicográfica, en concreto de la idea de las colocaciones.

Como se vio en el ejemplo *Vi a un hombre en el parque con un telescopio*, los sintagmas preposicionales se pueden adjuntar o bien al verbo ("vi") o bien al nombre ("hombre", "parque"). Las dos maneras básicas de resolver este dilema de una manera estructural, sin recurrir a la interpretación semántico-pragmática, son:

1. Adjunción a la derecha: el constituyente tiende a unirse con el constituyente más próximo a su derecha.
2. Adjunción mínima: el constituyente tiende a unirse al núcleo que implique menos nodos sintácticos intermedios.

Como señalan Hindle y Rooth, en el caso de la adjunción de un SP en un contexto verbo + objeto, estos dos principios hacen predicciones opuestas: la adjunción a la derecha elige al núcleo nominal y la adjunción mínima elige al verbo. Varios experimentos psicolingüísticos sugieren que ninguno de estos dos principios por sí solos dan un tratamiento satisfactorio del problema, pues unas veces se prefiere el verbo y otras el nombre.

De acuerdo con estos hechos, los autores buscaron evidencias de *asociaciones léxicas entre parejas verbo-preposición y nombre-preposición*. Para ello utilizaron un corpus analizado automáticamente y calcularon estadísticas sobre las distribuciones.

Posteriormente, utilizaron dichas estimaciones para resolver las ambigüedades en favor de la adjunción que tenía más probabilidad. Los resultados fueron comparados con los producidos por dos "analizadores humanos": los humanos obtuvieron tasas de error entre 12% y 15%, mientras que el programa consiguió un 20%, algo peor que la de los humanos pero significativamente mejor que con estrategias estructurales, que apenas pasaban del 50% de aciertos.

5.5.3. Anotadores estocásticos

El etiquetado automático de textos es una de las aplicaciones más exitosas de las técnicas estadísticas. Consiste en colocar la marca de categoría más apropiada a cada una de las palabras que aparecen en un corpus. Dicha marca dependerá del tipo de información que se quiera destacar: morfosintáctica, puramente sintáctica, semántica, etc. Como la anotación, ya sea manual o automática, es un paso esencial para la Lingüística de corpus y para todo procesamiento estadístico, la aparición de programas de etiquetado ha puesto a disposición de muchos investigadores una herramienta utilísima: el número de datos con información anotada puede aumentar rápidamente en comparación con el tamaño de la muestra anotada por un humano.

La principal razón de su popularidad, sin embargo, se debe a su altísimo índice de precisión: la mayoría de los etiquetadores estocásticos actuales sobrepasan el 95% de acierto, es decir, nos encontramos con una palabra mal analizada cada 20. Bien es cierto que esta cifra se consigue en parte gracias a que más del 50% de las palabras de cualquier corpus no son ambiguas (Allen, 1995), pero en cualquier caso es un acierto cercano al de un ser humano, y obtenido con mucho menor esfuerzo y tiempo. Los etiquetadores de este tipo suelen combinar conocimiento (en forma de lexicón de palabras y afijos morfológicos) y probabilidades reflejadas en un modelo de Markov. Por tanto, la tarea de estos etiquetadores se divide en dos:

1. Reconocimiento de las palabras no ambiguas morfosintácticamente.
2. Selección de la categoría o etiqueta más probable cuando hay más de una opción para una palabra determinada.

Sólo en el segundo caso se aplica la información estadística. Es otro ejemplo de cómo utilizar la probabilidad para resolver la ambigüedad.

Como ilustración de la técnica, se presentará la versión del Etiquetador de Xerox para el español, desarrollada por F. Sánchez León y A. Nieto Serrano para el proyecto CRATER, en el Laboratorio de Lingüística Informática de la UAM (Universidad Autónoma de Madrid). Aquí se sigue lo expuesto en Sánchez y Nieto (1995), así como en un informe técnico del proyecto.

El Etiquetador de Xerox es un programa de dominio público que, en principio, puede aplicarse a cualquier lengua. Sánchez y Nieto mostraron que, con ciertas modificaciones, el modelo puede utilizarse para etiquetar morfosintácticamente textos del español, sin restricciones de dominio temático. La adaptación consistió en "ajustar" el modelo original para el inglés a la peculiaridades morfosintácticas del castellano. Para ello los autores tuvieron que adoptar varias decisiones en los componentes esenciales:

1. El *conjunto de etiquetas* (o *tagset*): son las etiquetas concretas que se utilizarán para marcar el texto. La elección de este conjunto está determinada por las características de la lengua y por los planteamientos teóricos y prácticos asumidos por el proyecto. Así por ejemplo, para la lengua inglesa se han propuesto 87 etiquetas (Brown Corpus), 135 (Lancaster-Oslo/Bergen), 166 (Lancaster UCREL) y 197 (London-Lund). La idea de refinar el etiquetario lo más posible para dar cuenta de distinciones gramaticales más sutiles es una estrategia que tiene que ser contrapesada por la precisión en el etiquetado: cuanta más información se incluya en el etiquetario (es decir, cuanto mayor sea el conjunto de etiquetas), menos preciso y más complejo será el etiquetado. Todo ello forma parte de lo que se ha denominado "planteamiento teórico y práctico del proyecto". Si además añadimos las peculiaridades de la lengua, el conjunto de etiquetas puede aumentar considerablemente. Hay que tener en cuenta que esta tarea se basa en las características morfosintácticas superficiales de las palabras flexionadas, de modo que cada lengua en función de su riqueza de fenómenos morfológicos (número, género, caso, tiempo, aspecto, incorporación, etc.) puede necesitar un conjunto diferente. F. Sánchez León propone un total de 475 etiquetas para el español, en las que se recogen la inmensa mayoría de los rasgos morfosintácticos recomendados por EAGLES y TEI (dos comités internacionales para establecimiento de estándares en el área de la Ingeniería Lingüística). F. Sánchez realizó el experimento de establecer seis etiquetarios diferentes, en cada uno se reducía parte de la información codificada. Por ejemplo, eliminar distinciones semánticas, funcionales, o incluso morfológicas. Los tamaños de los seis conjuntos eran, por orden decreciente, 475, 387, 223, 152, 76 y 40. Curiosamente, se comprobó que reducir el número de etiquetas no suponía una disminución significativa de la ambigüedad. Es decir, para el español no parece que las consideraciones de tamaño del etiquetario contribuyan a mejorar la eficacia de la tarea, al menos para el tipo de sublengua técnica del corpus utilizado. De ahí que el mencionado investigador proponga utilizar el etiquetario más amplio y fino.
2. *El lexicón*: todos los etiquetadores estocásticos necesitan un lexicón para reconocimiento de formas flexionadas, suplementado con listas de sufijos. Estos dos componentes son los responsables de la asignación de al menos una etiqueta a cada palabra. La lista de sufijos (o terminaciones) proporciona la información necesaria para inferir la etiqueta morfosintáctica cuando no se encuentre la palabra en el diccionario. El lexicón debe incluir la mayor cantidad posible de formas que puedan tener más de una etiqueta, aunque eso no garantiza por sí solo la mejora de la precisión en el etiquetado: la ambigüedad úni-

camente se puede reducir efectivamente con el modelo probabilístico, el último componente del etiquetador.

3. *Modelo de Markov oculto:* se trata de una generalización de las cadenas de Markov para tratar la probabilidad condicionada de que se dé una etiqueta para una determinada palabra cuando dicha palabra puede tener más de una etiqueta. Dicho más formalmente, es un autómata en el cual un estado puede tener varias transiciones que salen de él, cada una de ellas con una probabilidad. La figura 5.3 representa una versión simplificada de una cadena de Markov oculta, donde un misma palabra puede tener varias etiquetas ("sobre/N" debe interpretarse como la palabra *sobre* con la categoría N, y así sucesivamente). Desde cada estado salen varias transiciones, que reflejan las posibles etiquetas. Naturalmente, desde cada estado las probabilidades de las transiciones son diferentes. La tarea del reconocedor es encontrar la secuencia más probable. Se suele utilizar un algoritmo especial, el algoritmo de Viterbi, para estimar la probabilidad de la mejor secuencia que va a cada etiqueta desde cada posición. Como todo modelo probabilístico, necesita ser entrenado para disponer de las probabilidades. Una de las particularidades interesantes de los modelos de Markov ocultos es que su entrenamiento se realiza sobre corpus no etiquetado. Sánchez León comprobó en su etiquetador estocástico para el español que aumentar el tamaño del corpus de entrenamiento no garantiza necesariamente una mayor precisión, como se suele afirmar de cualquier modelo estadístico. Esta observación ya había sido hecha por Merialdo (1994). En los experimentos de Sánchez León con diferentes tipos de etiquetarios y tamaños de corpus de entrenamiento, observó que para el español los mejores resultados se obtenían con 60.000 palabras de prueba.

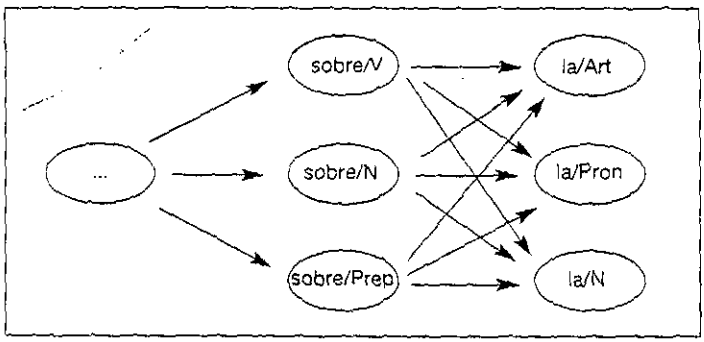


Figura 5.3. Un fragmento de una cadena de Markov oculta.

Análogamente a la comparación que se estableció entre gramáticas formales para distintas lenguas, esta aplicación de los modelos de Markov ocultos al etiquetado automático proporciona conclusiones interesantes:

1. Tanto las gramáticas formales como las cadenas de Markov son supuestamente instrumentos descriptivos universales. Sin embargo, parece que ambos modelos aplicados a lenguas distintas producen resultados diferentes.
2. Esta afirmación se sustenta en el hecho de que las adaptaciones a lenguas genética y, hasta cierto punto, tipológicamente relacionadas como el francés, alemán o español, han obtenido resultados comparables pero con modificaciones del modelo original. En el caso del etiquetador morfosintáctico para el castellano mencionado, hubo que utilizar un etiquetario dos veces mayor que los etiquetarios más finos para el inglés. Su lexicón estaba compuesto por unos 40.000 lemas que expandidos generaban más de 440.000 formas. Un diccionario equivalente para el inglés tendría probablemente una cuarta parte de palabras plenamente flexionadas. Como se vio en el ejemplo comparativo de la gramática de PROTEUS, para conseguir los mismos resultados que con el inglés hacen falta más recursos de máquina (memoria RAM) y de tiempo de procesamiento.
3. La supuesta universalidad de estos modelos puede quedar en entredicho si se experimentan con lenguas estructuralmente muy diferentes al inglés. En el caso del etiquetador de Xerox, F. Sánchez señala que para lenguas como el vasco, húngaro o finés, con muchas más formas derivadas de cada lema, el tamaño del fichero del lexicón aumentaría hasta hacer imposible el etiquetado. No solamente es un problema el tamaño del lexicón, el diseño mismo del sistema (basado en la identidad entre lema y forma superficial, típico de las lenguas aislantes como el inglés) no permitiría reconocer fenómenos como la incorporación de argumentos internos al lexema verbal, fenómeno muy habitual en numerosas familias no indoeuropeas, como la paleosiberiana, la esquimal, la indopacífica o la amerindia.

Nuevamente hay que señalar, como se hizo para los modelos formales, que la LC es una disciplina que busca ante todo resultados prácticos. La universalidad es deseable pero no prioritaria. Los modelos y las técnicas se suelen experimentar primero con el inglés y luego se van refinando con su extensión a otras lenguas. Cuando el modelo existente no vale para una lengua, suelen proponerse modelos alternativos: la morfología en dos niveles de Koskenniemi nació por la necesidad de tratar computacionalmente la morfología del finés. Análogamente, cabe imaginarse que surgirán nuevos modelos

estadísticos para dar cuenta de fenómenos que no tratan los actuales, a medida que se desarrollen sistemas PLN para lenguas distintas.

5.5.4. Gramáticas sintagmáticas probabilísticas

A lo largo del capítulo se ha visto cómo la probabilidad se podía implementar fácilmente en autómatas de estados finitos. En este apartado veremos que su aplicación también se puede generalizar a las gramáticas independientes del contexto. Cada regla de la gramática tendrá una probabilidad, $P(\alpha \rightarrow \beta)$. Al igual que en el caso de todas las transiciones a partir de un estado en un autómata, las probabilidades de todas las reglas que expandan el mismo elemento no terminal α_n tienen que sumar uno. De esta manera, $P(\alpha_n \rightarrow \beta_1)$ tiene una probabilidad de expandir α_n opuesta a $P(\alpha_n \rightarrow \beta_2)$, $P(\alpha_n \rightarrow \beta_3)$, etc. En la tarea de reconocimiento o generación de oraciones, cada análisis posible tendrá una probabilidad, resultado del producto de las probabilidades de todas las reglas empleadas en la construcción del árbol. Finalmente, se ordenarán los análisis resultantes por orden de mayor a menor probabilidad y, en su caso, se escogerá el más probable.

¿Cómo podemos elaborar una gramática probabilística a partir de una no probabilística? La idea intuitiva es recoger estadísticas sobre el empleo de cada regla gramatical. Para ello, lo más sencillo es tomar un corpus anotado con análisis sintácticos y contar el número de veces que se aplica cada regla. A la hora del recuento, hay que tener en cuenta tanto el símbolo no terminal que aparece a la izquierda de la regla como la regla en sí misma. El corpus de entrenamiento está compuesto por 10 oraciones muy sencillas, que se muestran anotadas en el cuadro 5.9. El cuadro 5.10 muestra una gramática probabilística para el español, de alcance muy limitado.

CUADRO 5.9. Corpus de entrenamiento de la gramática.

```
[O [SV [V Fuimos] [SP [P al] [SN [N cine]]]]]
[O [SV [SN [ PRO Me]] [V gusta]] [SN [DET el] [N pescado]]]
[O [SN [N Juan]] [SV [V colecciona] [SN [N sellos]]]]
[O [SN [N Brian]] [SV [V vive] [SP [P en [SN [N Yorktown]]]]]
[O [SN [DET Este] [N hombre] [SV [V es] [AD] egoísta]]]
[O [SN [DET El] [N médico]] [SV [V está] [SP [P en] [SN [N Barcelona]]]]]
[O [SV [ADV Ayer] [V terminó]] [SN [DET el] [N curso] [ AD] académico]]]
[O [SN [DET Mi] [N hermana] [SV [V terminó] [ADV ayer) [SN [DET el] [N curso]]]]
[O [SV [V Vi] [SP [P a] [SN [DET un] [N hombre]]]] [SP [P con] [SN [DET un] [N telescopio]]]]]]
[O [SV [V Vi] [SP [P a] [SN [DET un] [N hombre] [SP [P con] [SN [DET un] [N bastón]]]]]]
```

CUADRO 5.10. Una gramática probabilística para el español.

Regla	*Recuento de α*	*Recuento de la regla*	*Probabilidad*
O → SN SV	10	5	0,5
O → SV	10	3	0,3
O → SV SN	10	2	0,2
SV → V SN	10	1	0,1
SV → V SP	10	4	0,4
SV → SN V	10	1	0,1
SV → ADV V	10	1	0,1
SV → V ADJ	10	1	0,1
SV → V ADV SN	10	1	0,1
SV → V SP SP	10	1	0,1
SN → N	17	6	0,35
SN → PRO	17	1	0,06
SN → DET N	17	8	0,47
SN → DET N ADJ	17	1	0,06
SN → DET N SP	17	1	0,06
SP → P SN	6	6	1

Con esta gramática podemos calcular la probabilidad de los dos análisis posibles de *Saludé a una mujer con un sombrero:*

(1)

[O [SV [V Saludé [SP [P a] [SN [DET una] [N mujer]]] [SP [P con] [SN [DET un] [N sombrero]]]]]] =

0,3 * 0,1 * 1 * 0,47 * 1 * 0,47 = 0,006

(2)

[O [SV [V Saludé [SP [P a] [SN [DET una] [N mujer] [SP [P con] [SN [DET un] [N sombrero]]]]]]]=

0,3 * 0,4 * 1 * 0,06 * 1 * 0,47 = 0,003

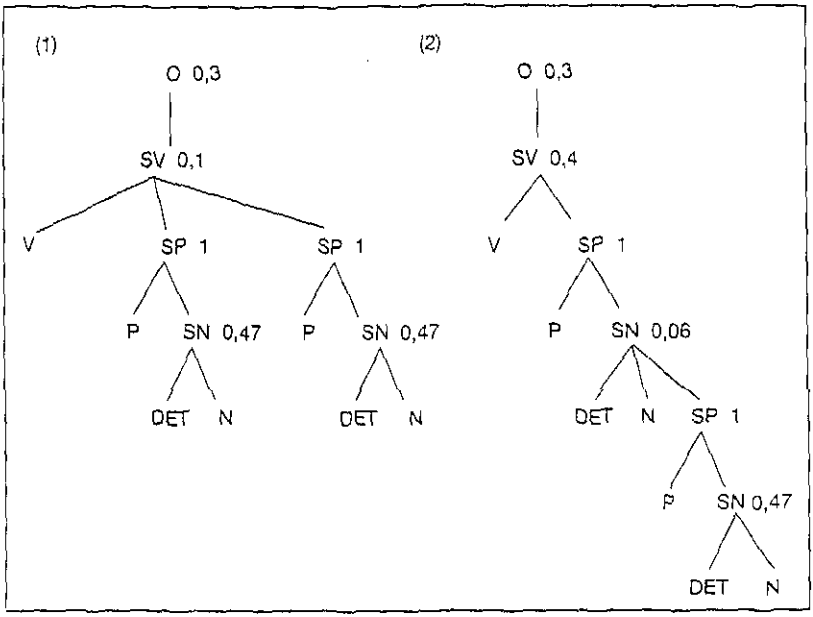


Figura 5.4. Árboles con probabilidad asociada.

Es decir, el primer análisis es más probable según la pequeña gramática probabilística. Obsérvese que aunque SV → V SP es más probable que SV → V SP SP, la escasa frecuencia en la muestra de SN → DET N SP hace que la probabilidad total de análisis quede por debajo. Naturalmente, al aumentar el corpus de entrenamiento los valores cambiarán.

Una gramática de este tipo no es más que una gramática independiente del contexto aumentada con información sobre su probabilidad de uso. Lo que ofrece al final del *parsing* es el análisis más probable para una oración, pero estructuralmente es el mismo análisis que proporcionaría la misma gramática independiente del contexto sin probabilidades. ¿Por qué tomarse el trabajo adicional de computar estadísticas para obtener el mismo resultado? Charniak (1993) expone cuatro argumentos para preferir las gramáticas probabilísticas:

1. *Resolución de la ambigüedad sintáctica:* el ejemplo anterior nos muestra la utilidad de las gramáticas probabilísticas cuando una oración puede tener más de un análisis. Sin embargo, esta afirmación hay que

tomarla con cautela: una gramática probabilística no es tan útil en desambiguación sintáctica debido a que los factores semánticos (las colocaciones) parecen ser realmente los decisivos, como serio en el apartado 5.6.2.

2. *Inducción de reglas gramaticales:* el aprendizaje de reglas gramaticales directamente de un corpus es uno de los aspectos más apreciados de las gramáticas probabilísticas. Sin embargo, la inducción no es tan sencilla como puede parecer a primera vista. Como señala Charniak, para obtener una gramática fiable hay que contar con dos conjuntos diferentes de entrenamiento, uno con oraciones gramaticales y otro con oraciones agramaticales. ¿Por qué son necesarias las oraciones agramaticales? Supongamos que estamos probando nuestra gramática y nos aparece una construcción no recogida en ella. Si asumimos que todas las oraciones de nuestro corpus de prueba son gramaticales, entonces el programa de inducción de reglas añadirá la construcción desconocida a nuestra gramática. Pero asumir que un corpus sólo contiene oraciones gramaticales es muy arriesgado: puede haber errores de todo tipo. Una solución a este problema es que un lingüista supervise cada nueva regla que se incorpora a la gramática, pero entonces ya pierde el atractivo de la adquisición automática de reglas... A pesar de su dificultad, hay varios proyectos que están trabajando en esta línea. Por ejemplo, Sekine y Grishman (1995) dan cuenta de un experimento con una gramática inducida con la que se obtuvo mejores resultados que con una gramática construida por lingüistas, aplicada a textos irrestrictos.
3. *Dar cuenta de la aceptabilidad:* una crítica habitual a las gramáticas basadas en el conocimiento es que no recogen muchas construcciones que aparecen en textos reales. Las gramáticas probabilísticas desarrolladas a partir de corpus permitirían asignar alguna probabilidad a oraciones aceptables. Charniak propone incluso tener dos gramáticas, una con reglas ortodoxamente gramaticales y otra con reglas "aceptables", que tendrían asignada una probabilidad mucho menor. En este punto se puede aplicar la crítica del apartado anterior: no habría forma automática de distinguir las oraciones gramaticales y aceptables de las totalmente agramaticales, que en ningún caso queremos en nuestra gramática.
4. *Un buen modelo lingüístico:* las gramáticas probabilísticas tienen el atractivo de incorporar tanto el conocimiento como el uso, mientras que las gramáticas estándares sólo manejan la competencia. Además, las gramáticas probabilísticas utilizan cierta información sobre el contexto (proponen la estructura más probable) mientras que las gramáticas independientes del contexto se caracterizan precisamente

por no tratarlo (a no ser que tengan alguna extensión del formalismo, como el uso de rasgos). En cuanto al tratamiento del contexto, los n-gramas suelen funcionar mejor que las gramáticas independientes del contexto si la relación es muy local (las n palabras anteriores). En cambio los n-gramas suelen fallar si la relación está más allá del número n.

El proyecto conjunto IBM-Lancaster es probablemente el ejemplo más conocido de cómo construir una gramática computacional a partir del uso de un corpus y de estadísticas. Su experiencia está recogida en Black, Garside y Leech (eds.) (1993). Su punto de partida es la incapacidad de los sistemas basados en el conocimiento (a principios de los años noventa) de analizar oraciones escogidas de textos de la vida real, no meramente oraciones de interés teórico. En varios experimentos de evaluación de sistemas tradicionales llegaron a la conclusión de que apenas producían análisis correctos para un tercio de las oraciones proporcionadas.

Su aproximación está organizada en dos grandes líneas:

— Desarrollo de una gramática basada en la frecuencia.
— El *parsing* (o análisis) probabilístico.

El objetivo de una gramática basada en la frecuencia es organizar el proceso de desarrollo de una gramática computacional para conseguir analizar la mayor cantidad posible de oraciones de un corpus. Para ello se fijaron varias metas:

— *Desarrollo de un banco de árboles y de un método de evaluación:* ambos son esenciales para saber si el análisis producido por la gramática es correcto o no. El banco de árboles *(treebank)* es un corpus que recoge los análisis apropiados para un número de oraciones. Este banco de datos tiene que ser anotado a mano, por lo menos al principio. Para el inglés existen varios corpus anotados sintácticamente, pero en español desgraciadamente no contamos todavía con ninguno de acceso público. El interés de estos bancos de árboles es enorme, ya que se emplean tanto en el entrenamiento como en la evaluación de los sistemas.
— *Tratar los errores más frecuentes primero:* esta estrategia está encaminada a cumplir el objetivo de analizar la mayor cantidad posible de oraciones. El método de evaluación permite elaborar una tipología de errores y ordenarlos por orden de frecuencia. Esta es la fase de mejora de la gramática
— *Repetición de ciclos Mejora/Evaluación:* el desarrollo de la gramática se organiza en torno a estas dos fases, que se repiten en ciclos sucesivos, lo que permite seguir la historia del progreso de la gramática.

Como se explicó anteriormente, los corpus de mejora y evaluación tienen que ser diferentes, para garantizar un avance real.

La metodología de IBM-Lancaster destaca por su interés en establecer criterios para conocer la cobertura real de las gramáticas y mejorar en los fenómenos más frecuentes, frente a orientaciones mucho más teóricas típicas de los modelos simbólicos. El libro es especialmente recomendable por su grado de detalle en la exposición de la metodología y en el carácter práctico de sus consejos. Tiene capítulos dedicados a la elaboración de un banco de árboles sintácticos y a la creación de gramáticas estadísticas, así como su entrenamiento y *parsing* probabilístico con ellas.

5.5.5. Traducción automática probabilística basada en alineamiento

Tradicionalmente la traducción automática ha consistido en desarrollar dos gramáticas basadas en el conocimiento, una que analice las oraciones de la lengua fuente y otra que genere las oraciones correspondientes en la lengua meta. La clave del proceso está en el punto de contacto, en establecer las correspondencias entre las dos gramáticas. El método más extendido ha sido utilizar un nivel de transferencia, lo suficientemente abstracto para permitir una traducción de estructuras y contenidos no demasiado literal o ligada a la estructura superficial de la lengua fuente. En este nivel de representación interna del contenido de la oración, los lingüistas computacionales especifican las reglas de traducción de una lengua a la otra.

En la última década el problema de la traducción automática se ha enfocado también desde la perspectiva de la adquisición de conocimiento automáticamente a partir de corpus. Para ello se suelen emplear corpus bilingües traducidos por traductores humanos. La traducción automática basada en corpus pretende sacar partido de los métodos estadísticos para aprender las correspondencias entre lenguas.

La aproximación estadística más radical a la traducción automática ha sido la del proyecto Candide de IBM Yorktown (Brown *et al.*, 1990 y Brown *et al.*, 1993). Utilizaron las actas del Parlamento canadiense escritas en inglés y francés. El objetivo era crear un modelo probabilístico de traducción. Esto es, si F es una oración de la lengua fuente y M una oración de la lengua meta, entonces

$$P(M \mid F)$$

es la probabilidad de generar una traducción M dada la oración F. La tarea consiste en asignar a cada pareja de oraciones (F, M) una probabilidad P(M|F). Esto debe interpretarse como la probabilidad de que un traductor

traduciría la oración M si se encontrara con la oración F. Por tanto, la traducción implica encontrar la oración M que tenga más probabilidad de aparecer dada la oración F. Aplicando el teorema de Bayes, la ecuación se queda en P(F | M) * P(M), donde P(M) es el modelo lingüístico (la probabilidad de la oración M) y se computa mediante n-gramas. Por otro lado, P(F | M) es el modelo de traducción, para el cual se experimentaron distintas técnicas que tenían que ver con el alineamiento de fragmentos de oraciones en cada lengua. El resultado final acabó en fracaso, debido principalmente a que trataban las oraciones como "secuencias de palabras sin estructura". De esta manera era imposible que el sistema capturara generalizaciones, por ejemplo acerca del orden de palabras en las oraciones de la lengua F y la lengua M.

El alineamiento de dos textos, original y su traducción, consiste en identificar los fragmentos relacionados en cada texto. El resultado es una lista con parejas de elementos, ya sean palabras, sintagmas, oraciones. Una vez recogidas las suficientes parejas, el sistema de traducción reconocerá las estructuras en la lengua fuente y las sustituirá por las correspondientes en la tabla de alineación. Se describirá a continuación brevemente un experimento bastante menos radical que el de IBM, pues combina el conocimiento de los lingüistas para encontrar las diferencias sistemáticas entre las lenguas, al tiempo que utiliza métodos estadísticos para aprender correspondencias bilingües. Este experimento pretende encontrar automáticamente reglas de transferencia entre el español y el inglés a partir del alineamiento de árboles de estructura, no de oraciones, como hacía el modelo de IBM. El experimento se está llevando a cabo en el proyecto PROTEUS de la NYU, con la colaboración del Laboratorio de Lingüística Informática de la UAM.

Grishman (1994) propone el alineamiento de estructuras funcionales, es decir, estructuras sintácticas abstractas donde se representa la relación predicado-argumento. Artículos posteriores (Meyers *et al.*, 1996 y Meyers *et al.*, 1998) describen los resultados de la puesta en práctica de las ideas de Grishman, que se resumen a continuación:

1. Dos gramáticas computacionales basadas en reglas analizan dos textos, uno traducción del otro. En el experimento, utilizamos un fichero de ayuda de Microsoft, del cual se escogieron 100 oraciones, aunque el objetivo naturalmente es analizar la mayor cantidad posible de oraciones. El resultado del análisis, los *parses*, son "estructuras lingüísticas regularizadas", que son bastante similares a las estructuras funcionales de LFG (Gramática Léxico Funcional). Una estructura regularizada está formada por un tipo (por ejemplo, cláusula o sintagma nominal), un núcleo y cero o más argumentos y modificadores, cada uno de los cuales tiene a su vez un papel y un valor. El papel puede ser un símbolo funcional gramatical (sujeto, objeto, tiempo, etc.) o puede ser una

palabra con valor funcional (una preposición o una conjunción). Los valores en la estructura regularizada pueden ser o elementos léxicos ("base de datos", "celda", etc.) o valores de algunos de los papeles funcionales, por ejemplo para género, número, tiempo, aparecen valores como femenino, plural o pasado, respectivamente.

2. Alineamiento de las estructuras regularizadas de las oraciones en las dos lenguas. Esta representación sintáctica más abstracta parece un nivel apropiado para realizar la transferencia, o al menos mucho mejor que el alineamiento de palabras, ya que se reducen la mayor parte de las diferencias sintácticas específicas de cada lengua. La clave del método consiste en encontrar un procedimiento para computar el alineamiento óptimo de las estructuras regularizadas. Para ello definimos lo que es un alineamiento completo de estructuras y asociamos una puntuación a cada posible alineación de tal manera que se escoge el alineamiento con mayor puntuación. La definición de alineamiento que se utiliza es la relación biunívoca (uno a uno) entre un subconjunto de nodos en el árbol fuente y un subconjunto de nodos en el árbol meta. Como condición esencial, exigimos que el alineamiento no viole la relación de dominio, es decir, si el nodo A domina al nodo B en el árbol fuente, entonces el correspondiente A' domina a B' en el árbol meta, para que se admita el alineamiento.
3. Extracción de reglas de transferencia: una vez que se ha producido un alineamiento óptimo de nodos en un árbol, entonces se procede a "cortar" el árbol para producir correspondencias bilingües. Estas correspondencias pueden ser de dos tipos: si se trata de elementos terminales entonces se produce una correspondencia léxica (palabra con palabra); si el alineamiento es de dos nodos no terminales se obtiene una correspondencia estructural. Una de las ventajas que tiene este modelo es que se adquieren reglas de transferencia tanto léxicas como estructurales. Aquí acaba el proceso de entrenamiento.
4. *Traducción: el proceso de traducción consiste en analizar un texto y* obtener sus estructuras regularizadas. Entonces las reglas de transferencia proyectan las estructuras regularizadas originales en las correspondientes estructuras regularizadas meta, siguiendo el criterio del candidato más probable cuando hay varias posibilidades. Finalmente, un generador convierte estas estructuras regularizadas en oraciones en la lengua meta.

El experimento se ha probado con 100 oraciones, y el corpus de entrenamiento es de unas 1.000 oraciones, todas ellas con una estructura moderadamente simple y bastante paralela en ambas lenguas. El sistema se encuentra en una fase muy temprana de desarrollo y no ha sido sometido a ninguna eva-

luación formal. Es probable que al aumentar la complejidad del texto, el sistema tenga problemas serios. En cualquier caso, parece que esta aproximación proporciona una interesante combinación de métodos basados en el conocimiento (para la producción de estructuras regularizadas) con otros basados en la adquisición automática de conocimiento por medio de técnicas estadísticas (para la producción de alineamientos y reglas de transferencia).

5.6. Limitaciones de los modelos estadísticos

En este apartado es obligado comenzar con las críticas de Chomsky a los modelos estadísticos. En *Estructuras sintácticas* y otros trabajos de su primera época, Chomsky proporcionó una serie de argumentos que durante muchos años han sido considerados como definitivos, extendiendo la idea de que los modelos probabilísticos "no proyectan ninguna luz especial sobre algunos de los problemas básicos de la estructura sintáctica" (Chomsky 1957: 32). Al igual que Gazdar a principios de los años ochenta revisó los argumentos de Chomsky contra las gramáticas independientes del contexto, Abney (1996) pasa revista a los argumentos de Chomsky contra los modelos estadísticos. Se recoge aquí sucintamente la polémica.

El principal argumento de Chomsky es que *no se puede definir gramaticalidad en términos de probabilidad*. Chomsky arguye que utilizar un corpus para reconocer las oraciones gramaticales es inherentemente insuficiente, ya que siempre habrá combinaciones gramaticales y agramaticales que no aparecerán en el corpus, por grande que sea. Toda oración que no aparece en el conjunto de entrenamiento tiene asignada la *probabilidad cero*, y por tanto será considerada agramatical. Chomsky (1957: 31) dice:

> Evidentemente, la habilidad que el hablante tiene de producir y reconocer locuciones gramaticales no está basada en nociones de aproximación estadística ni cosa por el estilo. La costumbre de llamar oraciones gramaticales a aquellas que "puedan existir", o a aquellas que son "posibles", ha sido responsable de cierta confusión en este punto. Es natural entender "posible" en el sentido de "altamente probable" y asumir que la nítida distinción del lingüista entre gramatical y agramatical se debe a la convicción de que, como la "realidad" de la lengua es demasiado compleja para ser descrita completamente, el lingüista debe contentarse con una versión esquematizada, en la que se reemplaza "probabilidad cero, y todas las probabilidades extremadamente bajas, por imposible, y todas las probabilidades más altas por posible" [El texto entre comillas es de Hockett].

Abney responde que este argumento es correcto en teoría, pero que en la práctica hay diversas técnicas para calcular las probabilidades de los suce-

sos que no aparecen en la muestra, y en particular cómo distinguir la probabilidad cero real (es decir, oraciones agramaticales) de la probabilidad cero por insuficiencia de datos. De ello ya se habló al tratar los n-gramas y el problema de los datos sin representación en la muestra.

La crítica de Chomsky se dirigía claramente al modelo de cadenas de Markov propuesto por Shannon. Como señala Abney, el propio Shannon se *cuidó mucho de avisar sobre las limitaciones de su modelo en este punto*, reconociendo que algunas cuestiones de buena formación no se pueden capturar con el modelo de n-gramas. Pero el mismo Shannon observó que a medida que aumenta n, el conjunto de oraciones agramaticales que se reconocen erróneamente como gramaticales por su modelo disminuye. De esta manera, sí se puede definir gramaticalidad en términos de probabilidad, según Abney: una oración será gramatical cuando su probabilidad sea mayor que cero, cuando n tiende a infinito.

El último argumento de Chomsky es mucho menos conocido, seguramente porque lo desarrolló en un capítulo del *Handbook of Mathematical Psychology*, de 1963. En él, el padre de los modelos formales generativos argumenta que las cadenas de Markov no son un modelo psicológicamente válido para la adquisición del lenguaje porque su estimación de parámetros es muy costosa. Abney responde de nuevo que la observación de Chomsky se puede matizar, ya que ni siquiera los estadísticos más recalcitrantes defenderían que *las cadenas de Markov son una teoría adecuada para explicar el procesamiento y adquisición del lenguaje*.

Por tanto, las críticas de Chomsky siguen manteniendo su validez en el plano teórico, pero en la práctica los modelos probabilísticos proporcionan soluciones, en consonancia con su carácter aproximativo y eminentemente aplicado. Además, Abney insiste en que la "inadecuación de los modelos de Markov no está en que son estadísticos, sino en que son las versiones estadísticas de los autómatas de estados finitos" (Abney, 1996: 23). Efectivamente, parece que los argumentos de Chomsky se deben a que las cadenas markovianas son de estados finitos y no a que son estocásticas.

En un artículo publicado a finales de 1997, Abney, con su clara perspicacia, muestra la aproximación que puede superar las limitaciones señaladas por Chomsky: la mayoría de los modelos estadísticos se han aplicado a gramáticas regulares e independientes del contexto, y precisamente se ha demostrado ampliamente que estas dos clases de gramáticas son inadecuadas para las lenguas naturales. Su inadecuación explicaría que los resultados obtenidos no sean fácilmente mejorables a partir de un punto, como consecuencia de su menor poder expresivo. Las gramáticas computacionales más utilizadas son las gramáticas atributo-valor (también conocidas por gramáticas de unificación, o basadas en rasgos), pero hasta la fecha no se ha propuesto ninguna versión probabilística satisfactoria, precisamente porque

los algoritmos de estimación de parámetros aplicados a las gramáticas estocásticas no son adecuados para las gramáticas de rasgos.

Como ya se vio, la utilización de rasgos permite tratar ciertos fenómenos dependientes del contexto, que no pueden ser tratados con las gramáticas de nivel inferior. Los algoritmos usuales de estimación tienen problemas con estos fenómenos. Abney (1997) propone utilizar el modelo de los *campos aleatorios (random fields)*, muy utilizado en el campo del procesamiento de imagen. A diferencia de los otros modelos estocásticos donde cada transición es independiente de las otras, en los campos aleatorios los procesos "evolucionan en sincronía". Este tema de investigación promete ser muy productivo en los próximos años.

La representatividad del corpus de entrenamiento es probablemente el problema más importante de todo modelo estadístico en general: son totalmente dependientes del corpus, de tal manera que si intenta extrapolar el modelo a otro dominio los resultados son pobres. Las dos "soluciones" habituales son la aplicación de alguna técnica de suavizado para tratar los datos no representados en el corpus, y el establecimiento de un período de aprendizaje para el caso de adaptación a un nuevo dominio.

Otra de las limitaciones de los modelos probabilísticos tiene que ver con la localidad: la estadística es muy eficiente con las relaciones locales, aunque incapaz con las relaciones a larga distancia, que es algo parecido a lo que le pasa a las gramáticas sintagmáticas clásicas. Pero mientras que las gramáticas sintagmáticas tienen medios para tratar los elementos discontinuos, la estadística parece que no ha encontrado soluciones.

En este sentido, las palabras de Lyons (1968: 90) parece que todavía se cumplen: "Muchos criterios al uso sobre la estructura estadística del lenguaje producen la impresión de que las probabilidades condicionadas que operan en todos los niveles de la estructura lingüística son necesariamente secuenciales, transicionales y progresivas. Evidentemente no es así". Efectivamente, se pueden citar los casos de constituyentes discontinuos (por ejemplo, la morfología no concatenativa o las lenguas con sintaxis no configuracional).

Por último, cabe resaltar que pese a las optimistas expectativas que suscitó a comienzos de los años noventa la posibilidad de contar con enormes corpus de datos, lo cierto es que ahora son una realidad y los resultados conseguidos no son tan espectaculares como se esperaban. Los métodos de "fuerza bruta", es decir, de recuento masivo de estadísticas, han demostrado que son poco fiables a partir de un punto (incluso llegan a perder precisión, como se vio con los etiquetadores estocásticos). En este momento se están explorando técnicas más sofisticadas que permitan un tratamiento sintáctico mucho más elaborado.

5.7. Ideas principales del capítulo

La estadística descriptiva sin contexto no sirve para hacer predicciones en un modelo probabilístico de una lengua.

Los modelos probabilísticos no explican algunos hechos interesantes sobre el lenguaje (cómo lo aprendemos, cuáles son las características universales de las lenguas, etc.). Si embargo, son eficaces para la descripción exhaustiva y la predicción aproximada.

Los partidarios de los modelos probabilísticos argumentan que son tan válidos teóricamente como los simbólicos, en el sentido de que proporcionan una teoría más o menos coherente y completa sobre el lenguaje. Pero en la realidad, pocos lingüistas creen en esta afirmación. Simplemente porque los modelos probabilísticos tienen (de momento) muchos menos recursos teóricos que los simbólicos. Dicho de otra manera, se pueden describir y explicar muchos más problemas con el aparato metodológico y conceptual simbólico que con el probabilístico.

Otra característica teórica propia de estos modelos es que defienden que la modelización de una lengua se debe realizar sobre la actuación (el uso), no sobre la competencia (que es típico de los modelos simbólicos). Su principal interés para la LC proviene precisamente del hecho de que son muy útiles para modelizar el uso lingüístico. Los modelos probabilísticos tienen básicamente tres grandes tipos de aplicaciones:

— Desambiguación sintáctica y semántica.
— Reconocedores de habla y anotadores eficientes.
— Elaboración de gramáticas probabilísticas.

5.8. Ejercicios

1. ¿Cómo se puede compaginar el hecho de que los hablantes tenemos la facultad de crear y entender mensajes nuevos (la creatividad lingüística que destaca Chomsky) con el hecho de que nuestra actuación presenta regularidades estadísticas?
2. Búsquese un ejemplo de oración ambigua estructuralmente que pueda dar varios análisis. A continuación, establecer una ordenación por orden de plausibilidad.
3. Proponer una pequeña encuesta de gramaticalidad, con el objetivo de que varios hablantes competentes se pronuncien, y comparar los resultados.
4. Ejemplos de oraciones agramaticales que los hablantes interpretan sin problema.
5. Contrástese el siguiente texto escrito por J. Lyons (1968: 99) con la vigencia actual de los métodos estadísticos:

> Hemos establecido [...] dos principios aparentemente contradictorios: el primero, que las consideraciones estadísticas son esenciales para comprender el funcionamiento y el desarrollo de las lenguas; el segundo, que en la práctica (y quizá también en principio) es imposible calcular con precisión la información que contienen las unidades lingüísticas en las expresiones reales. Esta aparente contradicción se resuelve admitiendo que la teoría lingüística, por lo menos en la actualidad [1968], no se interesa, ni puede, por la producción y comprensión de las expresiones en sus situaciones de uso reales [...] sino por la estructura de las oraciones consideradas en abstracto en cuanto a las situaciones en que aparecen las expresiones reales.

6. ¿Cómo se establece la relación Gramática Universal / Gramática Particular dentro de la aproximación probabilística? Evidentemente, para cada lengua habrá que crear el modelo basado en sus datos, pero ¿las técnicas estadísticas se aplican de manera equivalente a todos los fenómenos de las lenguas naturales y a todas las lenguas? ¿Podemos obtener resultados diferentes si aplicamos la misma técnica de modelización estadística a lenguas distintas? ¿De qué dependerá esa diferencia: de los datos utilizados o de la estructura de la propia lengua?
7. Búsquense ejemplos de aplicación de código equivocado en la descodificación del mensaje. Sirva esta anécdota como caso real: un hablante nativo de español de visita en Nueva York necesita urgentemente utilizar los lavabos de un restaurante. Sabe que la palabra apropiada en el dialecto neoyorquino es "restrooms". Cuando llega a los lavabos descubre un letrero que dice "Restrooms only for patrons". Se vuelve muy contrariado a su mesa y pregunta por qué no hay servicios para los clientes. Alguien le explica que "patrons" en el dialecto americano es "cliente".
8. Analícese críticamente la afirmación de "que las restricciones locales en algunas lenguas naturales son muy fuertes". ¿Hasta qué punto la localidad –es decir, la influencia estructural de los elementos más cercanos entre sí– es un fenómeno universal? ¿Es posible que algunas lenguas sean más localistas que otras (por ejemplo, el inglés frente al español, latín o vasco)? Relaciónese este punto con el problema de los constituyentes discontinuos en las gramáticas formales.
9. Tomando un pequeño texto de un periódico o de una revista (20 oraciones puede ser una cifra de referencia), anotarlos sintácticamente, construir la gramática basada en los datos y calcular las probabilidades, como en el ejemplo de la gramática independiente del contexto probabilística.

6.
Modelos inspirados en la Biología

A lo largo del libro se han visto muchos aspectos del lenguaje humano que se pueden modelizar matemáticamente y, por tanto, implementar en un programa de ordenador. En este capítulo se examinarán modelos que se inspiran no en las técnicas matemáticas existentes, sino en el funcionamiento de los organismos vivos. La metáfora de "lengua como organismo" es antigua en Lingüística: fue el punto de apoyo principal en la lingüística del siglo XIX, *del mismo modo que el concepto de "estructura" lo ha sido en el siglo XX*. Esta concepción biológica del lenguaje insiste especialmente en los aspectos evolutivos, dinámicos, heterogéneos, inestables, de variación y diversidad.

Obviamente, en la actualidad la difusión de estos modelos es sólo marginal, por su propia inmadurez y por la supremacía casi absoluta de los modelos matemáticos clásicos. Sin embargo, varios factores han motivado la aparición de estos modelos "biológicos":

1. Desarrollo de la Biología en la segunda mitad del siglo XX, especialmente la Genética.
2. Insuficiencia de los modelos matemáticos clásicos para dar cuenta del funcionamiento general y completo de los sistemas complejos, como el cerebro, la sociedad o el lenguaje. Estos sistemas se caracterizan por ser dinámicos y adaptarse continuamente a las nuevas situaciones del entorno.
3. Mayor conocimiento sobre el funcionamiento material del cerebro humano.

4. Interés por la diversidad lingüística, como lo demuestran las corrientes tipologistas recientes, y su relación con la diversidad biológica y humana (Cavalli-Sforza y Cavalli-Sforza, 1993).
5. Interés por el origen del lenguaje y las lenguas humanas, así como por la evolución y cambio lingüístico.

El punto de conexión de todas estas tendecias con el tema de este libro es el uso de ordenadores y su capacidad para simular realidades. Efectivamente, los nuevos modelos "biológicos" se caracterizan por estar definidos formalmente de tal manera que, igual que los modelos simbólicos y estadísticos, pueden ser codificados en un programa y comprobada su validez mediante la experimentación. (Es más, en realidad la mayoría de estos modelos o bien emplean algunas técnicas estadísticas para el aprendizaje –las redes neuronales– o bien combinan representaciones simbólicas con cuantificaficaciones selectivas –los algoritmos genéticos–). Dicho de otra forma, son modelos que también simulan la capacidad lingüística y por tanto deben ser incluidos dentro de la LC. Sin embargo, hay que avisar que la aplicación de estos modelos a las lenguas naturales está todavía muy poco experimentada, no así en Inteligencia Artificial, donde toda una especialidad está dedicada al estudio de sistemas que aprenden y se desarrollan por sí solos, como si fueran organismos vivos. Por tanto, se puede decir que en este punto más que en ningún otro, los sistemas de PLN están basados en técnicas desarrolladas en IA.

Estos sistemas que aprenden y evolucionan se suelen dividir en dos grandes grupos, en función de si imitan el comportamiento del órgano cerebral a nivel individual o de si imitan el comportamiento de las especies en su lucha por la supervivencia:

1. Modelos inspirados en el cerebro: conexionismo (redes neuronales)
2. Modelos inspirados en la vida y en la evolución: Computación Evolutiva (algoritmos genéticos).

6.1. Redes neuronales (conexionismo)

En esta sección se agruparán varios métodos conocidos por distintos nombres: *conexionismo* y *redes neuronales*. Todos ellos investigan la posibilidad de simular computacionalmente procesos intelectuales complejos mediante grandes redes formadas por unidades parecidas a las neuronas. Las unidades *(nodos)* son todas idénticas y de muy escasa complejidad. El comportamiento inteligente se obtiene mediante patrones que miden la "fuerza" de conexión entre unidades (de ahí su término más extendido, conexio-

nismo). Estos modelos parecen especialmente indicados para el reconocimiento de patrones y el aprendizaje.

El objetivo es modelizar fenómenos cognitivos utilizando algunas propiedades básicas que se dan en el funcionamiento de las neuronas en el cerebro. Sin embargo, hay que ser consciente de que estos sistemas son simplificaciones muy grandes de sus modelos reales, las neuronas. Por ejemplo, se calcula que el cerebro contiene más de 100.000.000.000 de neuronas, y cada neurona tiene una media de 1.000 conexiones. Frente a estas magnitudes, nos encontramos con que los sistemas más elaborados llegan a 1.000 nodos. En cuanto a la excitación de las conexiones, estos sistemas sólo permiten dos estados (excitado o inhibido) mientras que las reacciones químicas que se producen en las sinapsis (el punto de encuentro de los axones y las dendritas) son de tipos muy variados, en función de los neurotransmisores que intervengan, de tal manera que se producen distintos niveles de activación y propagación de señales. Evidentemente, ni siquiera se puede especular con la idea de que las redes neuronales imitan directamente el funcionamiento real del cerebro. Por eso, desde hace unos años se prefiere el término *conexionismo* (o incluso procesamiento distribuido en paralelo), ya que "redes neuronales" puede sugerir la idea equivocada de que se realiza una réplica artificial del cerebro. Dicho de otra manera, las simulaciones computacionales apenas pueden aportar conocimiento a la ciencia neurológica sobre la actividad neuronal y los comportamientos que surgen de dicha actividad (memoria, lenguaje, pensamiento). Rumelhart y McClelland (1986) insisten en que no tratan de reproducir el funcionamiento neuronal, sino que se inspiran en las neuronas para modelar los procesos cognitivos.

Muchos investigadores piensan que el verdadero atractivo de estos sistemas reside en el desarrollo de una nueva matemática para sistemas complejos (Smith, 1991). Básicamente, las redes conexionistas son sistemas dinámicos que se describen mediante ecuaciones matemáticas. Es decir, cuando la información de los nodos de la red empieza a combinarse se produce un comportamiento dinámico complejo que es el resultado de una formalización matemática basada en la asignación de valores de peso a cada conexión. Por tanto, la investigación en conexionismo está produciendo conocimiento acerca de las propiedades matemáticas de las redes y sistemas dinámicos en general, los cuales han demostrado que son capaces de resolver determinados problemas que parecían intratables, en concreto problemas de optimización de recursos. Como señala Smith (1991), lo que no está claro todavía sobre las propiedades matemáticas de los modelos conexionistas son las clases de problemas que son capaces de resolver en un tiempo razonable (es decir, su poder computacional).

¿Qué diferencias hay entre estos modelos y los simbólicos y probabilísticos que se han visto en anteriores capítulos?

A) Conexionismo frente a modelos simbólicos

Ya se ha mencionado que el conexionismo supone un planteamiento radicalmente diferente a los modelos simbólicos. Ambos son modelos computacionales, pero tienen concepciones muy distintas de lo que es "computación". Para el paradigma simbólico, computación es manipulación de símbolos de acuerdo con unas reglas. Para el paradigma conexionista, computación son los procesos por los cuales las unidades se excitan y se inhiben; por tanto, no hay ni símbolos ni reglas, sino unidades y conexiones que se van creando o reforzando a medida que la red aprende a resolver problemas concretos. Dicho de otra manera, en un modelo conexionista no hay ni una gramática ni un lexicón explícitos, sino que el reconocimiento de estructuras (fonemas, morfemas, oraciones) se realiza sobre la base de semejanzas en patrones de activación de nodos: dos estructuras son similares si excitan los mismos nodos.

Una de las características más destacadas de esta aproximación es la asunción del *procesamiento distribuido en paralelo* (de hecho, este el nombre adoptado por un conocido grupo de investigación): los diferentes procesos de reconocimiento de palabras, análisis morfosintáctico y semántico actúan simultáneamente, activándose en el momento en que les llega alguna información, y proporcionando el resultado de los cómputos a los otros nodos. Este concepto resalta el carácter profundamente interactivo del procesamiento del lenguaje natural (y de cualquier otro módulo cognitivo) al tiempo que puede proporcionar una explicación al hecho de que los hablantes obtienen alguna interpretación a pesar de las deficiencias del "estímulo" o mensaje. Esta aproximación también asume una posición muy diferente a los modelos simbólicos clásicos en cuanto a la naturaleza del procesamiento: éste se realiza de una manera aproximada y cuenta con bastante redundancia. Desde la perspectiva conexionista el procesamiento consiste en la acción conjunta de múltiples unidades distribuidas actuando en paralelo. Debido a la redundancia y al carácter aproximativo del procesamiento, estos sistemas intentan explicar por qué los hablantes son capaces de superar las deficiencias y los errores en la comunicación lingüística: aunque alguna unidad implicada en el procesamiento "falle" el mensaje suele interpretarse gracias a que el conjunto de unidades es muy numeroso. Por tanto, una idea clave es que cualquier tipo de procesamiento implica la interacción de un gran número de unidades.

El modelo conexionista, desde mediados de los años ochenta, supone el paradigma alternativo al modelo simbólico inspirado en la metáfora del ordenador, dentro de las ciencias cognitivas. Efectivamente, la Inteligencia Artificial, la Psicología y la Lingüística, entre otras disciplinas, han estado dominadas desde la década de los sesenta por modelos basados en símbolos y

reglas. El paradigma conexionista cuestiona ambos y ha provocado una fuerte polémica con los cognitivistas simbólicos. Bechtel y Abrahamsen (1991) exponen las principales críticas y contrarréplicas de esta polémica.

Un pequeño comentario histórico: la idea de utilizar las neuronas como modelo de computación es muy antigua. Ganascia (1994) sitúa los orígenes del conexionismo en la llamada cibernética, a finales de los años cuarenta, y marca una clara diferencia con los modelos simbólicos en IA:

— La cibernética se planteaba como objetivo modelizar el comportamiento fisiológico del sistema nervioso.
— La inteligencia artificial simbólica sólo se interesaba por simular el comportamiento cognitivo, por ejemplo, el lenguaje o el razonamiento lógico.

En la primera etapa (años cincuenta y sesenta) se impusieron claramente los modelos simbólicos, pero el resurgir del conexionismo en los ochenta y noventa, así como la crisis de los modelos basados en reglas, están abriendo expectativas a las redes neuronales artificiales.

B) Conexionismo frente a modelos estadísticos

La oposición entre ambos paradigmas no es tan radical como en el caso anterior, pues las redes conexionistas utilizan habitualmente cálculos estadísticos para la asignación de pesos. El conexionismo se basa en computación numérica, no en manipulación de símbolos. En ese punto, los modelos conexionistas y estadísticos coinciden frente a los simbólicos. Sin embargo, las redes neuronales no se consideran un subtipo de los modelos estadísticos, porque en ellas el cerebro funciona como la referencia esencial, mucho más que la inducción de conocimiento estadístico a partir de datos codificado en un autómata.

Se pasará ahora a la descripción del modelo conexionista básico y en el siguiente apartado se comentarán algunas aplicaciones al tratamiento del lenguaje.

6.1.1. El modelo conexionista básico

Se trata de una red de unidades elementales (nodos), cada una de las cuales tiene un valor de activación que se computa de acuerdo con unas sencillas fórmulas numéricas. Es un sistema dinámico que, una vez suministra-

da una información de entrada, comienza a expandir excitaciones e inhibiciones entre sus unidades, y no se para hasta que se consigue un estado estable. La activación inicial que se proporciona es el problema (por ejemplo, reconocer un patrón) y la configuración estable obtenida al final del proceso es la solución del sistema al problema.

Los componentes esenciales de un sistema conexionista son cuatro (Bechtel y Abrahamsen, 1991):

1. *Unidades.*
2. *Ecuaciones* que determinan un valor de activación para cada unidad en cada momento
3. *Conexiones* entre unidades con valores de peso, de modo que la actividad de una unidad puede influir en la actividad de otras unidades.
4. *Reglas de aprendizaje* que cambian el comportamiento de la red mediante la modificación de los pesos de las conexiones. Estos pesos cambian de manera gradual y lenta, de forma que durante cualquier computación permanecen esencialmente constantes.

Se utilizará una aplicación al español como ejemplo de funcionamiento de una red neuronal. Espinosa *et al.* (1996) desarrollaron un segmentador de oraciones dentro de textos en castellano. Los signos de puntuación provocan ambigüedades en la segmentación de oraciones; por ejemplo, un punto puede indicar final de oración, final de abreviatura ("etc.", "*et al.*") o aparecer en medio y al final de un acrónimo (O.N.U.). En el caso más complejo, el punto puede ser parte de una abreviatura o acrónimo y además ser final de oración. Estas distintas posibilidades funcionales suponen una ambigüedad de carácter textual, que tiene que resolverse antes de comenzar el análisis oracional. La segmentación en oraciones es un paso ineludible en cualquier pre-procesador de textos. Habitualmente se utilizan métodos basados en conocimiento: reglas heurísticas que determinen las condiciones de cada caso, o listas de abreviaturas y palabras que suelen aparecer en el contexto de los signos de puntuación. Como señalan Espinosa *et al.* (1996), estas técnicas adolecen de la especificidad con respecto a determinado tipo de textos y a las abreviaturas de una determinada lengua. Esto supone un gran coste de adaptación a otros textos y a otras lenguas. El sistema neuronal propuesto por estos autores supera estos condicionamientos. Veamos cómo funciona.

El problema que tiene que resolver la red neuronal es decidir la función de determinado signo de puntuación. Aquí se tratará el punto, que es el que tiene más funciones posibles: final de oración, parte de abreviatura, parte de acrónimo y parte de número. La red neuronal reproducirá el contexto alrededor de un posible final de oración. Cada una de las palabras (mejor dicho,

sus categorías sintácticas) que componen el contexto son las unidades de la red. Los pesos iniciales de cada conexión se asignan mediante las probabilidades de cada categoría, calculadas a partir de textos. Entonces empieza el "entrenamiento" de la red para reconocer distintas configuraciones. Al final del entrenamiento, la red neuronal es capaz de clasificar los signos de puntuación por sí sola, con una efectividad del 97,5%. En otras palabras, ha "aprendido" a segmentar oraciones en un texto (o más genéricamente, a reconocer patrones de estructura textual). De ello se hablará a continuación.

A) Aprendizaje

Para cualquier sistema conexionista, el aprendizaje no consiste en añadir nuevo conocimiento o modificar el existente (como hacen los sistemas simbólicos). El aprendizaje es cambio en el peso de las conexiones entre las unidades. Para entender esto hay tener en cuenta que los pesos de las conexiones determinan en parte la red y el resultado obtenido, por tanto, si cambiamos los pesos se obtiene un cambio en las características generales del sistema. Las reglas (también llamadas algoritmos) de aprendizaje están pensadas para reducir lo más posible los errores globales producidos a partir de los ejemplos de aprendizaje. Por tanto, el objetivo es proporcionar una manera de modificar los pesos que aumente la capacidad de la red para conseguir el resultado deseado en el futuro. Los algoritmos o procedimientos de aprendizaje realizan una inducción estadística a partir de un conjunto de entrenamiento. Se pueden distinguir, entonces, dos etapas:

1. *Entrenamiento:* Cuando el sistema está en esta fase, tanto las activaciones de los nodos como los pesos de las conexiones cambian cada vez que se prueba la red.
2. *Procedimientos de aprendizaje:* después de una serie de pruebas de entrenamiento, se evalúan los resultados. Los procedimientos de aprendizaje se suelen clasificar en:

   — *Aprendizaje supervisado:* cuando se le proporciona al sistema el resultado deseado para un *input* particular, de manera que tiene que compararlo con su resultado real,
   — *Aprendizaje por reforzamiento:* sólo se proporciona al sistema información global, positiva o negativa, acerca de su actuación, pero sin dar el resultado deseado
   — *Aprendizaje no supervisado:* donde no se proporciona información adicional.

En este punto hay que hacer una importante observación: aunque tanto los valores de activación de las unidades como los pesos de las conexiones pueden modificarse en respuesta a determinados *inputs*, lo cierto es que en la estrategia global de aprendizaje tienen papeles distintos:

— Los valores de activación de los nodos nos informan de los cambios de estado temporales en la red durante el procesamiento de una entrada.
— Los pesos de las conexiones representan los cambios más duraderos en la red, pues son el resultado del procesamiento con éxito de todos los diferentes *inputs* con que se ha estado entrenando al sistema.

De ahí que se considere aprendizaje en la red cuando cambian los pesos de las conexiones después de una etapa de entrenamiento y evaluación.

Esta concepción del aprendizaje ha provocado una interesante discusión en todas las disciplinas cognitivas. En concreto, ¿cuál es el *status* de las reglas?, ¿realmente existen o es más adecuado hablar de otros tipos de estrategias como la analogía? Por otra parte, este conocimiento ¿es explícito o implícito? En palabras de dos de los conexionistas más destacados, Rumelhart y McClelland, el comportamiento de un sistema cognitivo no está gobernado por reglas, sino más bien descrito por reglas de manera aproximada. Según los conexionistas, el aprendizaje y el conocimiento operan en un nivel por debajo del de las reglas simbólicas clásicas, es decir, a un nivel subconceptual o subsimbólico. Las referencias comentadas en el último capítulo del libro aportan más información sobre el tema.

En la sección de ejercicios se plantean algunas cuestiones relativas al aprendizaje automático desde las perspectivas simbólica y conexionista. En los años noventa, se han propuesto en esta área de investigación diferentes combinaciones híbridas, que se recogen en libros como Wermter (1995) o Wermter *et al.* (eds.)(1996).

Se han visto los componentes básicos de una red conexionista. Como resumen, se puede decir que estas redes proyectan una clase de patrones (los *inputs)* sobre otra clase de patrones (los *outputs)*, y lo hacen mediante la codificación de regularidades estadísticas en el peso de las conexiones, las cuales pueden ser modificadas de acuerdo con la experiencia *(aprendizaje)*. Precisamente esta capacidad de proyectar unos patrones sobre otros constituye el atractivo mayor de los modelos conexionistas, pues el procesamiento de patrones es una de las capacidades cognitivas más notables de los seres humanos.

B) Reconocimiento de patrones

Reconocimiento de patrones es la proyección de un patrón específico sobre otro más general. Es decir, identificar a un individuo como ejemplo de una clase. Esta capacidad también se conoce por el nombre de *categorización*, y se ha empleado en Psicología, Filosofía y Lingüística para referirse a la adquisición de categorías de objetos: ¿qué es lo que caracteriza a una "silla" frente a otros tipos de "muebles"?, ¿cómo aprendemos que "silla", "mesa" y "armario" son ejemplos de la clase superior "mueble"?

Además de la aplicación al reconocimiento de categorías de naturaleza semántica, las redes conexionistas se utilizan en la percepción sensorial, de ahí que se hayan empleado en el reconocimiento de caracteres, de unidades textuales o en procesamiento de habla. Se verá brevemente en el siguiente apartado.

6.1.2. Aplicaciones

Como hemos señalado, los modelos conexionistas se emplean con éxito en tareas que impliquen reconocimiento de patrones. Así algunas de sus aplicaciones en PLN son:

— Reconocimiento de habla: las redes neuronales permiten recoger grandes cantidades de datos acústicos y procesarlos en paralelo. Sacan partido de la posibilidad de organizarse en múltiples niveles relacionados entre sí: los niveles inferiores contienen rasgos acústicos, luego alófonos, fonemas, sílabas y hasta palabras. En la última década se ha experimentado con métodos mixtos de redes neuronales y modelos de Markov ocultos, que mejoran sobre todo el entrenamiento sobre datos insuficientes.
— Desambiguación léxica: uno de los aspectos destacados de las redes neuronales es su capacidad para establecer relaciones entre palabras, simulando un lexicón mental. Por ejemplo, son útiles para establecer grupos de palabras relacionadas. De esta manera, se pueden emplear en tareas como la desambiguación de palabras polisémicas.
— Etiquetación morfosintactica: como vimos en el ejemplo de reconocimiento de signos de puntuación con varias interpretaciones, las redes neuronales se pueden utilizar para reconocer distintas categorías sintácticas en palabras homógrafas.

La gran ventaja de los modelos conexionistas frente a los convencionales (simbólicos o estadísticos) es su capacidad de aprender. En muchas apli-

caciones donde hay que efectuar análisis no muy detallados de la estructura pero sí reconocer una enorme variedad de posibilidades (en concreto los etiquetadores y segmentadores, o los sistemas de habla) los resultados obtenidos por los modelos estadísticos y los modelos conexionistas son comparables. La diferencia radica en que una vez construida la red neuronal su adaptación a un nuevo dominio es completamente automática (la red se encarga de aprender a partir de ejemplos del nuevo dominio). En cambio los modelos simbólicos y estadísticos necesitan un notable esfuerzo de adaptación y, en cualquier caso, una mayor supervisión por parte de los lingüistas computacionales.

Otro aspecto muy interesante de los modelos conexionistas es que parecen muy bien equipados para tratar problemas que impliquen razonamiento aproximado, ambigüedad, variación. En ese sentido, comparten muchos puntos de contacto con los modelos probabilísticos, ya que ambos cuentan con el entrenamiento para su "aprendizaje".

Entre las limitaciones de los modelos conexionistas podemos señalar que, hasta la fecha, han tenido poco éxito en el procesamiento sintáctico y semántico de oraciones debido a su incapacidad para tratar la recursión (Uszkoreit, 1996), que como sabemos es uno de los rasgos más destacados de las lenguas naturales.

6.2. La Computación Evolutiva: los algoritmos genéticos

Este término es relativamente reciente y con él se intenta agrupar una serie de técnicas, cuyo paradigma más conocido es el de los algoritmos genéticos. En Computación Evolutiva se entiende que la naturaleza es una "inmensa máquina de resolver problemas" (De la Herrán, 1998; Sandoval, 1996) y buscan en su funcionamiento inspiración para modelos computacionales que resuelvan problemas prácticos. El padre de los algoritmos genéticos es John Holland, que publicó en 1975 uno de los clásicos de la IA, *Adaptación en sistemas naturales y artificiales*, donde se expuso por primera vez un paradigma computacional que imita a los sistemas complejos adaptativos. Los algoritmos genéticos han alcanzado gran relevancia a partir del desarrollo, desde mediados de los años ochenta, de un área de investigación multidisciplinar conocida por la Teoría de la Complejidad (Waldrop, 1992; Lewin, 1992; Casti, 1994; Gell-Mann, 1994). Los objetos de estudio son sistemas que presentan muchos agentes independientes que interactúan entre sí de diferentes maneras. Se caracterizan por la auto-organización espontánea (los elementos son capaces de buscar el equilibrio dentro del sistema) y por ser adaptables (aprenden de la experiencia y se adaptan al entorno). Los sistemas complejos se dan tanto en las ciencias de la naturaleza (evolución de las espe-

cies, fenómenos atmosféricos, movimientos sísmicos) como en las ciencias del hombre (culturas, economía, arte, lenguaje). En Moreno Sandoval (1996) se defiende que las lenguas naturales deber ser consideradas sistemas complejos adaptativos y se presenta el esbozo de la aplicación de los algoritmos genéticos de Holland a la simulación de la evolución dentro de un sistema fonológico de una lengua natural. Este trabajo servirá para ejemplificar la utilización de algoritmos genéticos en LC.

Los algoritmos genéticos son una de las técnicas computacionales más utilizadas en las simulaciones de sistemas complejos adaptativos. Se basan en estrategias de aprendizaje activo a partir de grandes cantidades de datos. Los modelos simbólicos están en desuso para este tipo de experimentos. Se prefieren los modelos inductivos que imitan el comportamiento de un sistema dinámico adaptable: estos programas aprenden a reconocer estructuras y patrones de comportamiento y cuando se enfrentan a situaciones nuevas donde se pueden tomar distintos caminos se ponen en funcionamiento estrategias de evaluación de candidatos. Cada hipótesis tiene asignado un valor que mide su plausibilidad y de todas las hipótesis en competencia gana la que tiene mayor puntuación ("fuerza" o "peso"; *fitness* es el término inglés). Este concepto de Holland está inspirado directamente en el comportamiento adaptativo de los sistemas evolutivos naturales. La idea central es reproducir el entorno en el que se produce una evolución, ya sea un ecosistema, un sistema económico o una lengua. Estos sistemas están compuestos por individuos que interactúan estableciendo relaciones muy complejas de competencia y colaboración. Por ejemplo, las unidades lingüísticas en cualquier nivel cambian (es decir, se transforman, surgen o desaparecen) con el paso del tiempo en función de factores internos y externos al sistema. En el experimento de simulación (Moreno Sandoval, 1996), el objetivo es especificar las unidades y su entorno, es decir, las fuerzas internas y externas que otorgan ventajas a unas unidades sobre otras.

Un algoritmo genético se basa en tres procesos consecutivos, que se repiten continuamente:

1. *Evaluación:* donde cada uno de los individuos (por ejemplo, los fonemas) recibe un valor en función de su capacidad para resolver problemas (en el ejemplo, la capacidad de combinarse con otros fonemas para formar unidades significativas de nivel superior).
2. *Selección:* de acuerdo con los valores asignados en la etapa anterior, los agentes adaptativos (es decir, los fonemas) se clasifican en supervivientes o no (estos últimos serán excluidos en la siguiente etapa evolutiva).
3. *Reproducción:* a partir de los supervivientes se generan nuevos individuos, en función de los valores obtenidos. Es necesario que se pro-

duzcan algunas mutaciones para garantizar la aparición de innovaciones que permitan la evolución de las unidades del sistema.

En general, los algoritmos genéticos tienen una fase inicial en la que especifican las unidades originales y las primeras reglas que regirán su interactuación dentro del entorno. A continuación se ejecutan repetidamente los ciclos de los tres procesos mencionados, y en cada uno se producen innovaciones que hacen cambiar las unidades y las reglas. Es importante señalar que aunque todo algoritmo genético se compone de alguna forma de evaluación, selección y reproducción, existen diferentes estrategias y técnicas para implementarlas en un programa. Precisamente la clave para desarrollar un buen modelo de simulación evolutiva reside en encontrar la estrategia que mejor se adapte al problema que tratamos. Por ejemplo, en el experimento fonológico se quiere determinar cuáles son los factores que influyen en un cambio fonético concreto. Básicamente, las estrategias son de dos tipos:

— Conseguir una convergencia más rápida hacia una solución (en este caso, se primaría buscar el factor decisivo).
— Hacer una exploración profunda del espacio de búsqueda (en este caso, se considerarían múltiples factores).

Ambas opciones son deseables pero contradictorias: si primamos la rapidez, simplificamos el problema y, si primamos la profundidad, puede que no encontremos una solución. Por tanto, lo habitual es optar por una solución de compromiso entre ambas, de tal manera que se consideren únicamente unas pocas reglas, lo que permite una estrategia de evaluación y selección eficiente.

En general, las reglas de selección se agrupan en dos grandes fuerzas, una conservadora y otra innovadora. La *fuerza conservadora* beneficia a los agentes o unidades que mejor resuelven un problema (en el experimento, por ejemplo, los fonemas que tienen mayor rendimiento funcional). La fuerza innovadora favorece la aparición de variantes en el sistema, de tal forma que se evite el estancamiento del sistema en torno a una configuración concreta. Dado que los sistemas complejos se caracterizan por su dinamismo y evolución, es necesaria cierta variedad para permitir la adaptación a problemas nuevos y cambiantes. Esto es lo que ocurre por ejemplo en cualquier lengua, que se mantiene "al borde del caos" (Moreno Sandoval, 1996). Es decir, son sistemas lo suficientemente estables para ser aprendidos y permitir la comunicación entre los miembros de una comunidad lingüística, pero lo suficientemente inestables para contar con variantes (dialectales, sociolectales, registros) de las cuales unas desaparecen y otras sobreviven. Esta

inestabilidad inherente es la que permite el cambio del sistema lingüístico, ya que no toda variante produce un cambio, pero sí todo cambio proviene de una variante.

En consecuencia, nuestro programa deberá contar con:

— Reglas que favorezcan la estabilidad, es decir, cualquier regla que prime la comunicatividad (por ejemplo, el rendimiento funcional, la simetría dentro del sistema, la clara distinción entre unidades).
— Reglas que favorezcan la innovación. En este punto se pueden incluir tanto reglas que impliquen la reducción del esfuerzo (por tanto una simplificación del sistema) como reglas que aumenten los recursos del sistema.

En esta exposición no se han incluido los factores externos o sociales que influyen en los cambios lingüísticos. En general, se reconoce que las causas sociolingüísticas son meros desencadenantes de cambios que se han producido en puntos débiles del sistema. Sociolingüistas tan reconocidos como Labov han llegado también a esa conclusión (Labov, 1994). En cualquier caso, todo factor externo puede clasificarse como fuerza conservadora (por ejemplo, las normativas académicas o las convenciones ortográficas) o como fuerza innovadora (el afán por distinguirse como grupo diferente o el prestigio social). Por tanto, la clasificación tradicional de factores internos y externos del cambio lingüístico se debe interpretar en nuestro experimento en términos de factores conservadores y factores innovadores.

Cualquier nivel lingüístico puede someterse a experimentación en una simulación evolutiva, pero evidentemente hay dos niveles que se prestan mejor, como lo demuestran los estudios diacrónicos: el fonológico y el léxico. Se ha escogido el nivel fonológico por una serie de motivos:

— El cambio fonético es un cambio producido en un sistema de reglas de la gramática (es decir, el cambio es básicamente regular).
— El cambio fonético puede reducirse a la siguiente tipología: adición, pérdida, reordenamiento y simplificación.
— El cambio fonético es relativamente simple, concreto y más cuantificable que otros tipos de cambio lingüístico.
— El cambio fonético está bien atestiguado en un número aceptable de lenguas, proporcionando una base fiable de comprobación empírica.

Se expondrá ahora la manera de codificar las unidades y las mutaciones. La forma habitual de especificar las unidades o agentes adaptativos es mediante cadenas de dígitos binarios. Si traducimos los fonemas a rasgos distintivos y éstos a dígitos podemos obtener la información en forma binaria, donde

cada zona corresponderá a un fragmento de información. Daremos un ejemplo muy simplificado (cuadro 6.1).

CUADRO 6.1. Algunos fonemas del español en formato binario.

	/p/	/b/	/m/	/t/	/d/	/n/	/a/	/i/	/u/
Conson.	1	1	1	1	1	1	0	0	0
Labial	1	1	1	0	0	0	#	#	#
Dental	0	0	0	1	1	1	#	#	#
Nasal	0	0	1	0	0	1	#	#	#
Sonora	0	1	1	0	1	1	1	1	1
Anterior	#	#	#	#	#	#	0	1	0
Central	#	#	#	#	#	#	1	0	0
Posterior	#	#	#	#	#	#	0	0	1

Nota: 1 significa valor positivo, 0 significa valor negativo, # significa valor no pertinente

Utilizando los rasgos distintivos apropiados, cada uno de los fonemas obtiene una codificación binaria. Por ejemplo, /b/ es 11001###, /a/ es 0###1010, etc. Por supuesto, esta codificación es meramente ilustrativa del procedimiento, pues la complejidad combinatoria de los rasgos fonológicos necesita cadenas mucho más largas: nada impide ampliar la cadena o utilizar varias subcadenas donde se especifique información, por ejemplo sobre la estructura silábica y prosódica (información relevante en el cambio fonológico).

La ventaja de traducir la información lingüística a cadenas binarias es que los algoritmos genéticos cuentan con distintas técnicas, inspiradas en las combinaciones genéticas reales, para "reproducir" las unidades mejor adaptadas y para provocar "mutaciones", y dichas técnicas trabajan sobre codificaciones binarias. Por ejemplo, si hemos asignado una zona de la cadena binaria a cada tipo de información, podemos establecer reglas de mutación que cambien un valor concreto en una posición definida. Así, podemos mutar una consonante sonora en sorda modificando el valor del dígito correspondiente de la cadena, o convertir una oclusiva en nasal cambiando de 0 a 1 el cuarto dígito de la cadena. Por supuesto, las cadenas que tienen más rendimiento funcional se repiten más, aunque también se permite alguna mutación esporádica para introducir variedad. En general, las técnicas de reproducción y mutación son muy flexibles: se pueden fijar tasas de mutación (cuánto y cuándo cambian los "hijos" con respecto a sus "padres"),diferentes para cada individuo (esto per-

mitiría, por ejemplo, establecer una cronología guiada de cambios, imitando a los cambios reales, de tal manera que al principio sólo mutaría un sonido en unas pocas palabras, las más frecuentes, y luego se extendería al resto de las palabras que contienen dicho fonema). También hay técnicas que hacen más probable que se produzca una mutación si se ha producido en un gen vecino, o hacen que una mutación sustituya una unidad sin utilidad.

Veamos ahora cómo serían las etapas del programa de simulación fonológica que se proponen:

1. Se traducen combinaciones de sonidos a combinaciones de cadenas binarias y se establecen unas reglas primitivas de combinación de unidades, así como un conjunto de factores conservadores e innovadores dentro del sistema. Estos factores actúan como presión selectiva sobre las distintas combinaciones posibles generadas por las reglas, premiando las unidades y reglas que se ajustan más a los datos reales que conocemos.
2. Las combinaciones más frecuentes y más ajustadas a los datos reales obtienen mayor puntuación, y así sucesivamente se otorga una valoración a todas las unidades.
3. Se lleva a cabo la reproducción y mutación, según las posibilidades de la tipología de cambios fonéticos antes mencionada. Este proceso introduce nuevas unidades en la población. Igualmente, se pueden cambiar, eliminar o incorporar nuevas reglas combinatorias y factores selectivos.
4. Se repite el ciclo varias veces, dejando que seleccione una serie de tendencias.
5. Se proporcionan nuevos datos reales de un momento histórico y se compara con los resultados de la simulación. Las coincidencias se valoran muy positivamente, se eliminan los resultados menos parecidos a la realidad y se premian las unidades, las reglas y los factores que han producido mejores resultados.
6. El proceso se repite intentando que el sistema aprenda de los datos reales y se vaya ajustando cada vez más a los factores que han condicionado el cambio del sistema fonológico.

¿Qué obtenemos con este tipo de simulación? Sobre todo, nos permite explorar distintas hipótesis evolutivas y sus resultados. Por ejemplo, podemos tomar como punto de partida algún conjunto de fonemas primitivo como los propuestos por Trombetti (consúltese Moreno Cabrera, 1997) y probar con diferentes estrategias de evolución. Todo consiste en dar más puntuación a determinados fenómenos y factores. También podemos partir de un sistema fonológico conocido, como el del latín vulgar, y tratar de reproducir

la evolución al sistema fonológico del castellano actual. En realidad, la simulación nos permite tanto ir hacia atrás (es decir, *retrodicción)* hasta llegar al hipotético origen del lenguaje, como ir hacia adelante *(predicción)* y postular posibles evoluciones futuras. El funcionamiento de estos programas, en su núcleo básico, es similar a los simuladores de vuelo o a los juegos de guerra. Como señala Gell-Mann (1994) lo verdaderamente importante de las simulaciones es conocer su relevancia con respecto a las situaciones reales. En concreto:

1. ¿Proporcionan intuiciones valiosas sobre situaciones reales?
2. ¿Revelan posibles comportamientos antes insospechados?
3. ¿Indican nuevas explicaciones de fenómenos ya conocidos?

Creemos que su atractivo es evidente para los lingüistas teóricos y que este tipo de experimentación se extenderá en el futuro. Como muestra de la potencialidad de estos sistemas hay que señalar que en la reunión anual COLING 98 está anunciado un taller sobre computación evolutiva.

CUADRO 6.2. Aplicaciones posibles de los algoritmos genéticos en Lingüística General

- *Retrodicción:*
  - Verificación de hipótesis de cambio lingüístico.
  - Reconstrucción de proto-lenguas.
  - Reconstrucción de la lengua original y de las distintas etapas evolutivas.
- *Predicción:*
  - Planificación lingüística.
  - Modelos de evolución de lenguas: valoración cuantitativa de los factores internos y externos.
- *Creación de juegos lingüísticos y lenguas artificiales.*

6.3. Una consideración marginal

Se pretende terminar este repaso a los modelos biológicos sin comentar que no sólo la Biología ha influido en la Lingüística, sino que esta última ha inspirado con sus métodos a algunos investigadores de Biología Computacional en el diseño de sistemas de reconocimiento y predicción de estructura genética. En concreto el sistema GenLang (Dong y Searls, 1994) desarrollado en la Universidad de Pennsylvania, el cual emplea herramientas y técnicas de Lingüística Computacional para encontrar genes en secuencias de datos. Los patrones genéticos son tratados como oraciones, que se espe-

cifican por medio de reglas gramaticales. En el apartado 8.3 se da su página de consulta en Internet.

6.4. Ideas principales del capítulo

Los métodos inspirados en la Biología permiten un tratamiento natural de dos propiedades esenciales del lenguaje humano: su aprendizaje y su evolución. El hecho de que su aplicación al procesamiento de lenguas haya sido poco experimentada, en comparación con los métodos simbólicos y estadísticos, se debe al poco interés que ambas propiedades tienen para las aplicaciones actuales.

Comparativamente, estos modelos adolecen de un desarrollo conceptual y técnico menor que los otros. Es difícil precisar si esto se debe a la falta de experimentación o a la inversa (no se emplean porque no están muy elaborados). Sin embargo, es probable que el agotamiento de los modelos puramente matemáticos contribuya a aumentar el interés por los modelos "biológicos".

6.5. Ejercicios

1. Compárese el planteamiento de problema y su solución desde las perspectivas simbólicas y conexionistas. En concreto, ¿se suministran el mismo tipo de información de entrada en ambos modelos? ¿Cuál es el tipo de información que se proporciona como solución en cada caso? ¿Tienen parser los modelos conexionistas?
2. Siguiendo con la comparación entre las aproximaciones simbólica y conexionista, ¿qué diferencias se dan entre las unidades de cada modelo? ¿Qué diferencias hay entre las reglas de una gramática formal y las reglas de aprendizaje de una red neuronal?
3. Sobre aprendizaje en sistemas computacionales: en Inteligencia Artificial existe un área de investigación muy activa conocida con el nombre de Aprendizaje Automático o Mecánico *(Machine Learning)*. Su objetivo no es otro que conseguir que los ordenadores aprendan de la experiencia. Dentro de los modelos simbólicos, las estrategias de aprendizaje consisten en modificar o añadir nuevas reglas al sistema. Compárense estas estrategias con las de los sistemas conexionistas que se han visto en el capítulo. ¿Cuál de ellas, en principio, estará más preparada para capturar la gradación y las sutilezas del aprendizaje?
4. Más sobre aprendizaje automático: analícese, utilizando algún caso concreto, el siguiente comentario de Bechtel y Abrahamsen (1991: 64-65) (la traducción es nuestra):