Procesamiento automático del español con enfoque en recursos léxicos grandes

Segunda edición ampliada y revisada

Alexander Gelbukh
Grigori Sidorov

Procesamiento automático del español con enfoque en recursos léxicos grandes

Segunda edición ampliada y revisada

Instituto Politécnico Nacional
— México —

Procesamiento automático del español
con enfoque en recursos léxicos grandes
Alexander Gelbukh y Grigori Sidorov

Primera edición, 2006
Segunda edición, 2010

Instituto Politécnico Nacional
Luis Enrique Erro s/n
Unidad profesional "Adolfo López Mateos"
Zacatenco, 07738, México, DF

Dirección de Publicaciones
Tresguerras 27, Centro Histórico
06040, México, DF

ISBN: 978-607-414-171-9

Impreso en México / Printed in Mexico
http://www.publicaciones.ipn.mx

Prólogo

El Laboratorio de Lenguaje Natural, fundado por los autores de este libro, Alexander Gelbukh y Grigori Sidorov, fue el primer grupo dedicado exclusivamente a la lingüística computacional en Hispanoamérica. Surgió en México, en el Centro de Investigación en Computación del Instituto Politécnico Nacional, hace casi catorce años.

El objetivo —y reto— del Laboratorio ha consistido en unir la lingüística y los avances de la tecnología computacional. Su misión consiste en "enseñar" a las computadoras el lenguaje humano, lograr que la máquina entienda no sólo las palabras y el discurso de manera literal, sino incluso intentar que descifre los matices y gradaciones de la lengua. No es una tarea fácil, pero los resultados han sido positivos y representan grandes avances para la investigación. Al mismo tiempo se han abierto caminos mucho más sencillos y prácticos en el análisis del propio lenguaje.

El ser humano aprende la lengua de forma natural y —como le sirve para relacionarse con el resto de la gente y para explicar su realidad— se familiariza con ella de manera casi intuitiva. Generalmente no necesitamos de un gran esfuerzo mental para expresarnos a través del lenguaje: lo hacemos espontáneamente porque estamos habituados a las convenciones que hacen falta para comunicarnos y entendernos.

La computadora no es como el ser humano, sino —como dicen Gelbukh y Sidorov— un «siervo tonto», al que se debe enseñar de manera distinta. La máquina no puede intuir ni actuar de forma espontánea, porque no conoce más realidad que la que nosotros podemos introducir en su «cerebro». Depende completamente del ser humano y de lo que uno ponga en ella. Por eso resulta muy complicado enseñarle el lenguaje, ya que requiere de esfuerzos enormes. La complicación aumenta más todavía cuando se trata de que la computadora interprete el discurso más allá de su estricto sentido literal.

No obstante, durante este complicado proceso, el investigador se enriquece. Al ir llenando los vacíos en el conocimiento de la computadora, tiene la oportunidad de profundizar aún más en su propio conocimiento de la lengua y la lingüística. Al mismo tiempo —de manera inevitable— renueva, amplía e incluso precisa sus dudas ante el conflicto de tener que resolver problemas que pudieron no parecerlo con anterioridad.

Afortunadamente, a pesar de lo difícil que puede ser «enseñarle» a la máquina, una vez que ésta ha «aprendido» se convierte en una herramienta invaluable. Esto se muestra, por ejemplo, en la forma en que puede facilitar el trabajo del lingüista gracias a su gran capacidad de memoria y de análisis. Estas y otras tareas que antes hubieran requerido muchas horas de trabajo y muchas personas, pueden ser realizadas ahora en unos instantes, gracias a los avances de los procesos de cómputo.

Este libro habla precisamente de los progresos que han logrado los autores en la formación de grandes corpus —a partir de la Internet, por ejemplo— y en el análisis de textos muy amplios —como diccionarios—; de los recursos que han elaborado y de los planes que tienen para que los resultados se obtengan cada vez con mayor precisión y sencillez. Por eso me permito expresar sin reservas mi reconocimiento a la perseverancia que muestran. Su empeño, sin duda, ha dado frutos. Además, en estas páginas

no sólo se encontrará investigación original en el campo del uso y construcción de recursos léxicos: el libro también introduce al lector en los métodos característicos de la lingüística computacional, en sus diversos enfoques y en sus muchas tareas, por lo que posee una indudable utilidad didáctica.

Raúl Ávila

Índice general

Prólogo 5

Prefacio a la segunda edición 19
Prefacio a la primera edición 21

Capítulo 1. Introducción 23

Parte I. Problemas generales del procesamiento de lenguaje natural 35

Capítulo 2. Tareas y aplicaciones de PLN 37
Capítulo 3. Niveles de lenguaje y su reflejo en PLN 87
Capítulo 4. Problemas del uso de diccionarios en PLN 103

Parte II. Aplicaciones del PLN con recursos léxicos grandes 131

Capítulo 5. Análisis morfológico automático basado en un diccionario de raíces 133
Capítulo 6. Análisis sintáctico automático basado en un diccionario de patrones de manejo 155
Capítulo 7. Resolución de correferencia con un diccionario de escenarios 169

Capítulo 8. Desambiguación de sentidos de palabras utilizando recursos léxicos 183
Capítulo 9. Recuperación de documentos con comparación semántica suave 197
Capítulo 10. Comparación de los coeficientes de las leyes de Zipf y Heaps en diferentes idiomas 211

PARTE III. Construcción de recursos léxicos para el PLN 227

Capítulo 11. Compilación automática del corpus léxica y morfológicamente representativo 229
Capítulo 12. Construcción automática del diccionario de colocaciones basándose en un análisis sintáctico automático 243
Capítulo 13. Evaluación automática de la calidad de los diccionarios explicativos 259
Capítulo 14. Detección automática de las primitivas semánticas 267

Bibliografía 283
Índice analítico 303

Índice detallado

Prólogo 5

Prefacio a la segunda edición 19
Prefacio a la primera edición 21

Capítulo 1. Introducción 23
1.1. La lingüística y la computación 23
1.2. La temática del libro 26
1.3. La estructura del libro 27

PARTE I. Problemas generales del procesamiento de lenguaje natural 35

Capítulo 2. Tareas y aplicaciones de PLN 37
2.1. Ayuda en preparación de textos 39
2.2. Búsqueda de información 41
2.3. Manejo de documentos 45
Búsqueda de documentos 45
Representación y navegación por los documentos 48
2.4. Gestión inteligente de documentos 49

Búsqueda inteligente de documentos 49
Combinación de la información tabular y textual 51
Representación inteligente de documento 52
Representación inteligente de un conjunto de documentos 56
Navegación inteligente por los conjuntos de documentos 58
Categorización automática de documentos 60
2.5. Interfaces en lenguaje natural 60
2.6. Traducción automática 63
2.7. Generación de texto 67
2.8. Aplicaciones recientes y emergentes 68
Bibliotecas digitales 68
Extracción de información, filtrado y alerta 69
Generación de resúmenes 70
Minería de texto 70
Manejo inteligente de documentos oficiales (e-Gobierno) 72
Estudio de Internet como un corpus enorme 72
Aplicaciones multilingües 73
Tecnologías de voz 74
Conducción de diálogo 75
2.9. Problemas y métodos de análisis y representación de texto 76
Problemas 76
Conocimiento lingüístico vs. extralingüístico 78
2.10. Métodos 79

2.11. Procesamiento de lenguaje natural en México. 80
2.12. Conclusiones ... 83

Capítulo 3. Niveles de lenguaje y su reflejo en PLN ... 87
3.1. Modelos buenos y modelos malos ... 89
3.2. Niveles de lenguaje natural ... 92
Fonética / fonología ... 93
Morfología ... 94
Sintaxis ... 95
Semántica ... 96
Pragmática ... 97
Discurso ... 98
3.3. Implementación de un procesador lingüístico ... 99

Capítulo 4. Problemas del uso de diccionarios en PLN ... 103
4.1. Relaciones entre las definiciones ... 106
4.2. Separación de los significados en sentidos ... 110
Falta de sentidos específicos ... 110
Sistema de sentidos demasiado detallado ... 113
Sentidos demasiado generales ... 118
4.3. Otros tipos de verificación formal ... 120
Verificación de la ortografía y la estructura de los artículos ... 121
Verificación de las marcas de sinonimia y antonimia ... 123
4.4. Herramienta ayudante de lexicógrafo ... 126
4.5. Conclusiones ... 129

Parte II. Aplicaciones del PLN con recursos léxicos grandes ... 131

Capítulo 5. Análisis morfológico automático basado en un diccionario de raíces 133
5.1. Modelos de análisis morfológico automático 137
5.2. Modelo de análisis a través de generación 143
Proceso de generación 144
Proceso de análisis 145
5.3. Modelos usados 146
Morfología nominal 146
Morfología verbal 147
5.4. Preparación de los datos 149
5.5. Implementación 149
5.6. Cómo se puede mejorar el analizador 151
5.7. Conclusiones 152

Capítulo 6. Análisis sintáctico automático basado en un diccionario de patrones de manejo 155
6.1. Análisis sintáctico automático 155
6.2. Requerimientos en el análisis de lenguaje natural 160
6.3. Ambiente de desarrollo 163
El uso y la información que proporciona el ambiente 164
6.4. Conclusiones 168

Capítulo 7. Resolución de correferencia con un diccionario de escenarios 169
7.1. Algunos ejemplos de correferencia indirecta 171
7.2. Correferencia indirecta como referencia a un elemento del escenario 173
7.3. Condiciones sintácticas 176
7.4. El algoritmo y el diccionario 179

7.5. Conclusiones y trabajo futuro 180

Capítulo 8. Desambiguación de sentidos de palabras utilizando recursos léxicos 183
8.1. Descripción del problema 183
8.2. Antecedentes 187
8.3. Enfoque basado en métodos estadísticos 188
8.4. Enfoque basado en las fuentes adicionales de conocimiento 190
8.5. Método propuesto 192
8.6. Evaluación del método 194
8.7. Conclusiones 195

Capítulo 9. Recuperación de documentos con comparación semántica suave 197
9.1. El método 198
9.2. Diccionarios 201
Diccionario morfológico 202
Sinónimos más cercanos 203
Sinónimos más lejaos 203
Todos los sinónimos y antónimos más cercanos 204
Todos los sinónimos y antónimos más lejanos 204
9.3. Interfaz del usuario 204
Opciones de búsqueda 205
Resultados de búsqueda 206
9.4. Conclusiones 209

Capítulo 10. Comparación de los coeficientes de las leyes de Zipf y Heaps en diferentes idiomas 211
10.1. Resultados experimentales 214

10.2. La posible explicación de la diferencia 216
10.3. Conclusiones 217
10.4. Apéndice 1: valores de los coeficientes de las leyes de Zipf y Heaps 218
10.5. Apéndice 2: listas de textos utilizados en los experimentos 221

PARTE III. Construcción de recursos léxicos para el PLN 227

Capítulo 11. Compilación automática del corpus léxica y morfológicamente representativo 229
11.1. El diccionario de contextos 233
11.2. Compilación del diccionario a través de la Internet 236
11.3. Resultados experimentales 238
11.4. Conclusiones 239

Capítulo 12. Construcción automática del diccionario de colocaciones basándose en un análisis sintáctico automático 243
12.1. Combinaciones idiomáticas, colocaciones y combinaciones libres de palabras 247
12.2. Enriquecimiento automático del diccionario de colocaciones 250
12.3. Evaluación del enriquecimiento automático .. 256
12.4. Conclusiones 257

Capítulo 13. Evaluación automática de la calidad de los diccionarios explicativos 259
13.1. Los datos para el experimento 260
13.2. El experimento 262
13.3. Conclusiones 265

Capítulo 14. Detección automática de las primitivas semánticas 267
14.1. La estructura de datos 269
14.2. El algoritmo 270
Definiciones 270
Funcionamiento 272
Depuración inicial del grafo 272
14.3. La metodología experimental 275
14.4. Resultados y discusión 278
14.5. Trabajo futuro 281
14.6. Conclusiones 281

Bibliografía 283
Índice analítico 303

Prefacio a la segunda edición

A nuestra gran satisfacción, la primera edición de este libro ha tenido gran éxito, y la casa editorial nos solicitó una segunda edición del mismo. Eso demuestra que el área de procesamiento automático del lenguaje natural y de la lingüística computacional son campos de investigación crecientes y muy actuales en los tiempos modernos, ya que hoy en día el uso de las computadoras se vuelve cada vez más amplio. Consecuentemente, la necesidad de utilizar todo el conocimiento representado en los textos y la posibilidad de interactuar con las computadoras en el lenguaje natural se vuelven de mayor importancia.

Para la segunda edición, hemos ampliado y revisado el libro. Agregamos un capítulo nuevo y modificamos varios otros capítulos, así como ampliamos y actualizamos la bibliografía.

Los temas presentados en este libro siguen siendo actuales en el área del procesamiento automático de lenguaje natural. Estamos seguros de que resultarán útiles para los lectores tanto los temas generales presentados en la primera parte del libro, como los temas más específicos presentados en los dos últimas partes.

Este trabajo fue realizado con el apoyo del Gobierno de México (SNI, Conacyt) e Instituto Politécnico Nacional (SIP-IPN, COFAA-IPN, PIFI), más específicamente, de los proyectos Conacyt 50206-H, 83270 y proyectos 20091587, 20090772, 20080787 y 20082936 apoyados por la Secretaría de Investigación y Posgrado (SIP) del IPN.

Prefacio a la primera edición

Este libro examina algunas de las aplicaciones prácticas de la computación tanto en la investigación lingüística como en la tecnológica del lenguaje natural. El objeto de estudio de este libro pertenece a la ciencia de la lingüística computacional. En general, se puede decir que ésta es una ciencia que tiene por el momento más problemas que soluciones, pero es un campo de investigación muy importante y sugestivo, porque aparte de generar aplicaciones útiles nos permite entender mejor la herramienta más importante que usamos los seres humanos: el lenguaje natural. El libro, sin embargo, tiene un enfoque más práctico y técnico que teórico, presentando al lector, después de una debida y amplia introducción al tema, un conjunto de nuevas técnicas para la solución de varios problemas específicos de procesamiento de texto por computadora.

El libro será útil tanto para los especialistas y estudiantes que se dedican a los problemas de Procesamiento de Lenguaje Natural (PLN) y áreas afines, como para los que apenas están empezando a familiarizarse con esta área. Otro grupo muy importante al cual está dirigido este libro son los lingüistas, que encontrarán en él ejemplos útiles tanto del uso de las técnicas computacionales en sus labores, como de las aplicaciones a su investigación. El libro hace uso extensivo de nuestros trabajos previos publicados en

varias revistas y congresos, con las actualizaciones y adecuaciones necesarias según los numerosos comentarios que recibimos de sus lectores, a quienes ahora expresamos nuestro más profundo reconocimiento.

Por último, aclaramos que los nombres de los autores aparecen en estricto orden alfabético.

Capítulo 1

INTRODUCCIÓN

En este libro presentamos resultados recientes de nuestra investigación en el procesamiento de textos en español por medio de la computadora, basado en el uso de los recursos lingüísticos. Más abajo explicaremos la relación entre el estudio del lenguaje español y la computación y especificaremos la temática de este libro exponiendo brevemente cada uno de sus capítulos.

1.1 La lingüística y la computación

La ciencia que estudia el lenguaje humano se llama lingüística. En esta gran ciencia existen ramas que representan su intersección con otros campos, tanto del conocimiento científico —por ejemplo, la psicolingüística o la sociolingüística—, como de la tecnología, la educación, la medicina, el arte y otras actividades humanas.

En particular, existe una relación muy especial e interesante —de gran beneficio mutuo— entre la lingüística y la computación.

Por un lado, el conocimiento lingüístico es la base teórica para el desarrollo de una amplia gama de aplicaciones tecnológicas, de cada vez más alta importancia, en nuestra incipiente socie-

dad informática —por ejemplo, la búsqueda de información y el manejo de conocimiento, las interfaces en lenguaje natural entre el humano y las computadoras o los robots, y la traducción automática, entre un sinnúmero de aplicaciones de alta tecnología.

Por otro lado, las tecnologías computacionales pueden dotar al lingüista de herramientas con las que ni siquiera podían soñar los investigadores de tiempos tan cercanos como las dos décadas anteriores, y de las que, dado el prohibitivo costo de las computadoras, hace unos cuántos años los lingüistas no disponían para sus labores cotidianas. Entre estas herramientas se puede mencionar la inmediata búsqueda de ejemplos de uso de las palabras y construcciones en enormes cantidades de textos; estadísticas complejas conseguidas milagrosamente rápido; análisis, marcaje y clasificación casi instantáneas —si se compara con su obtención manual, a lápiz y goma de borrar— de cualquier texto; detección automática de la estructura en un lenguaje desconocido, por mencionar sólo algunas. Los buscadores avanzados de Internet han abierto la puerta a todo un mundo de lenguaje, a un corpus tan enorme que puede considerarse como la totalidad del lenguaje humano a disposición de cualquiera, en forma palpable y medible —a diferencia de un corpus tradicional que sólo representa una gotita del océano del uso colectivo del lenguaje.

Entre los beneficios destaca también la posibilidad de la verificación masiva de las teorías, las gramáticas y los diccionarios lingüísticos. Hace unos años, para verificar la gramática propuesta por un estudioso, el lingüista debía esforzar su intuición en busca de un ejemplo no cubierto por ella, y si no encontraba ese ejemplo, tenía que admitir que la gramática era completa —lo que no es un buen paradigma del método científico. Hoy en día, la implementación de la gramática en forma de un analizador automático permite no sólo verificar si una gramática es completa o no, sino, además, medir cuantitativamente en qué grado

es completa y exactamente qué productividad tiene cada una de sus reglas.

Pero el beneficio principal de las tecnologías computacionales para la lingüística general, en todas sus ramas —desde la lexicografía hasta la semántica y pragmática— es la motivación para compilar las descripciones de lenguaje completas y precisas, es decir, formales —lo que significa un estándar de calidad en cualquier ciencia. Se puede comparar con la relación entre la física y las matemáticas: son las matemáticas las que motivan a los físicos a manifestar sus observaciones y pensamientos en forma de leyes exactas y elegantes.

Más específicamente, esta relación se puede describir de la siguiente manera. La lingüística, como cualquier ciencia, construye los modelos y las descripciones de su objeto de estudio —el lenguaje natural. Tradicionalmente, tales descripciones fueron orientadas al lector humano, en muchos casos apelando —sin que siquiera los mismos autores lo notaran— a su sentido común, su intuición y su conocimiento propio del lenguaje. Históricamente, el primer reto para tales descripciones —reto que ayudó mucho a elevar la claridad y lo que ahora conocemos como formalidad— fue la descripción de las lenguas extranjeras, en la que ya no se podía apelar al sentido lingüístico propio del lector. Sin embargo, incluso estas descripciones muy a menudo se apoyaban implícitamente en las analogías con el lenguaje propio del lector, sin mencionar las persistentes referencias al sentido común.

La revolución computacional regaló al lingüista un interlocutor con propiedades singulares: no sabe nada de antemano, no tiene ninguna intuición ni sentido común y sólo es capaz —eso sí, enormemente capaz— de interpretar y aplicar literalmente las descripciones del lenguaje que el lingüista le proporciona. Nos referimos a la computadora. Así como un niño nos hace preguntas que nos llevan a pensar profundamente en las cosas que siempre hemos creído obvias —pero que de hecho son muy

difíciles de explicar— y en las cuales nunca hubiéramos pensado si no se nos hubiera preguntado, así la computadora hace al lingüista afinar y completar sus formulaciones a partir de la búsqueda de respuestas a preguntas tan difíciles de responder que antes resultaba más simple considerarlas «obvias». Así, la computación convierte a la lingüística —que era tradicionalmente una rama de las humanidades— en una ciencia exacta, además de presentarle nuevos retos y darle nueva motivación y nuevas direcciones de investigación. Esta transformación se puede comparar con la que en su época propiciaron las matemáticas en la física.

El amplio campo de intersección e interacción entre la lingüística y la computación se estructura a su vez en varias ciencias más específicas. Una de ellas es la lingüística computacional. Esta ciencia trata de la construcción de modelos de lenguaje «entendibles» para las computadoras, es decir, más formales que los modelos tradicionales orientados a los lectores humanos. Otra área es el procesamiento automático de lenguaje natural (PLN), que se ocupa más de los aspectos técnicos, algorítmicos y matemáticos de la aplicación de dichos modelos a los grandes volúmenes de texto, con el fin de estructurarlos según la información contenida en ellos, de extraerles la información útil, de transformar esta información —es decir, de traducirla a otro lenguaje—, etcétera. Estas dos disciplinas tienen el mismo objeto de investigación, aunque lo consideran desde enfoques diferentes. Como sabemos, en la investigación casi nunca existen casos «puros». Sin profundizar demasiado en la definición de estos dos términos, haremos notar que muchos investigadores consideran que en la práctica no hay gran diferencia entre ellos.

1.2 La temática del libro

Para acotar el tema de este libro sólo consideramos aquí el procesamiento automático de lenguaje natural, y sólo en relación con

los textos escritos. Es decir, no abarcamos el tema de reconocimiento de voz ni los recursos que se usan para esta tarea. Es más, nos enfocamos específicamente en el uso de los recursos léxicos grandes para el análisis de texto.

Por recursos léxicos grandes en este libro entendemos tanto los diccionarios —de diversos tipos: monolingües, bilingües, explicativos, diccionarios de sinónimos, tesauros, enciclopedias, diccionarios de colocaciones, etc.— como los corpus. Éstos últimos son colecciones muy grandes de textos, normalmente con alguna información lingüística adicional como las marcas morfológicas, sintácticas, las marcas de sentidos de palabras, referenciales, etcétera.

Los diccionarios son indiscutiblemente recursos léxicos, pero ¿lo son los corpus? Los corpus son fuente de información de todos los fenómenos del lenguaje —no sólo de los léxicos— y se usan para varios tipos de investigación lingüística —gramatical, sintáctica, pragmática, etc. En la actualidad, sin embargo, su uso principal consiste en la obtención de información léxica, según la famosa idea de Firth de conocer las palabras por su compañía, es decir, analizando sus contextos de uso. En este sentido, los corpus son recursos léxicos muy valiosos y ayudan a los lexicógrafos complementando su intuición.

1.3 La estructura del libro

El libro está estructurado de la siguiente forma. La Parte I da una introducción general a las tareas de procesamiento de lenguaje natural. Después, la Parte II presenta diferentes aplicaciones prácticas que se basan en los recursos léxicos grandes. Por último, la Parte III describe los problemas relacionados con la compilación automática y semiautomática de los recursos léxicos —los corpus y los diccionarios.

El resto del libro está compuesto por los siguientes capítulos:

Capítulo 2. Introduce al lector a las técnicas de procesamiento de lenguaje natural que permiten a las computadoras entender —hasta cierto grado— y procesar el texto o el habla en el lenguaje humano —por ejemplo, español— y realizar las tareas que requieren de tal comprensión. Entre éstas se pueden mencionar la búsqueda de información, la traducción automática de un lenguaje a otro —digamos, de inglés a español—, el diálogo con el usuario para escuchar sus órdenes y comunicarle respuestas, etcétera. Recientemente, en este campo de investigación se ha tenido mucha actividad. El capítulo da una presentación resumida de las tareas técnicas y problemas científicos que implica el procesamiento de lenguaje natural, las técnicas que han tenido más éxito, el estado actual y las soluciones propuestas según las tendencias más recientes. Además, se discuten algunos problemas que aún quedan por resolver.

Capítulo 3. Describe en términos muy generales los problemas que enfrenta en la actualidad la lingüística computacional. La intención del capítulo es mostrar qué tipos de problemas existen en los diferentes niveles del lenguaje natural —fonética/fonología, morfología, sintaxis, semántica, pragmática y discurso— en relación con el procesamiento automático de textos, así como dar algunos ejemplos de aplicaciones que usan el conocimiento lingüístico para el procesamiento automático del texto.

Capítulo 4. Discute algunos problemas acerca del uso de diccionarios y sus posibles soluciones. Un diccionario explicativo es un sistema complejo con numerosas relaciones, tanto entre los elementos localizados en diferentes lugares de su texto como entre las definiciones y el vivo uso de las palabras en el lenguaje. Debido a esta complejidad se hace muy difícil la detección manual de ciertos tipos de defectos en el diccionario, tales como círculos viciosos en el sistema de definiciones, inventario inconsistente de las palabras usadas en las definiciones —vocabulario

definidor—, definiciones inconsistentes o insuficientes, división incorrecta de los artículos en los sentidos específicos, marcas inconsistentes de sinonimia y antonimia, etc. En este capítulo explicamos cómo los algoritmos computacionales pueden ayudar en el desarrollo interactivo de los diccionarios y presentamos una herramienta computacional correspondiente.

Capítulo 5. Presenta un método para el análisis morfológico de textos. La mayoría de los sistemas de análisis morfológico están basados en el modelo conocido como morfología de dos niveles. Sin embargo, este modelo no es muy adecuado para los lenguajes con alternaciones irregulares de raíz —por ejemplo, el español. En este capítulo describimos un sistema computacional de análisis morfológico para el lenguaje español basado en un modelo distinto, cuya idea principal es el análisis a través de generación. El modelo consiste en un conjunto de reglas para obtener todas las raíces de una forma de palabra para cada lexema, su almacenamiento en el diccionario, la producción de todas las hipótesis posibles durante el análisis y, finalmente, su comprobación a través de la generación morfológica. En el sistema experimental implementado según este método, usamos un diccionario de 26,000 lemas. Para el tratamiento de palabras desconocidas se usa un algoritmo basado en heurísticas. El sistema, además, está disponible, sin costo, para el uso académico en www.cic.ipn.mx/~sidorov/agme; cualquier persona interesada en el desarrollo de algún fragmento de los sistemas para el lenguaje natural puede usarlo.

Capítulo 6. Analiza los requerimientos con que debe cumplir un analizador sintáctico de lenguaje natural y presenta un ambiente de desarrollo de analizadores de texto en español. Las ideas expuestas se ilustran con la descripción de un sistema que incluye un analizador morfológico y un parser sintáctico, basado en una gramática libre de contexto con elementos de unificación. Éste último incorpora tecnología para la ponderación de las variantes

de análisis usando la información sobre la compatibilidad y co-ocurrencia de las palabras. El programa brinda al usuario varias vistas que facilitan el desarrollo y la depuración de los recursos léxicos y la gramática.

Capítulo 7. Discute un algoritmo para la resolución de la llamada anáfora indirecta. El algoritmo está basado en el uso de un diccionario de los escenarios típicos asociados con cada palabra, así como en un tesauro —u ontología— de tipo estándar. Un ejemplo del fenómeno anafórico del tipo bajo la consideración: «Compré una casa. La cocina es muy grande», donde existe una relación oculta entre las palabras *cocina* y *casa*. Se describe la estructura del diccionario de los escenarios típicos: en el caso más simple, este diccionario presenta para cada palabra una lista de todas las palabras que denotan los típicos o posibles participantes y procesos involucrados en la situación relacionada con la palabra. Por ejemplo, religión es relacionada con iglesia, vela, sacerdote, oración, Biblia, etcétera. El algoritmo se basa en el descubrimiento de la intersección del escenario de la posible fuente de la anáfora con el del posible antecedente, o la inclusión de una de estas palabras en el escenario de la otra. También se discute la combinación de la anáfora indirecta con tales complicaciones, como el uso metafórico de palabras, las relaciones de la hiponímia, derivación, frases subordinadas, etcétera. El método puede usarse incluso para detectar la presencia de la relación anafórica, lo que es importante para la interpretación tema-remática de la oración y para la comprensión coherente del texto.

Capítulo 8. Discute uno de los problemas importantes en el PLN y en particular en los portales de recuperación de información en Internet y en bibliotecas digitales relacionada con la polisemia de palabras. Por ejemplo, un mecánico de autos busca «¿dónde comprar un gato?» y obtiene respuestas sobre los «gatos monteses», «gatos siameses», y otros. Un comerciante de frutas busca «producción de lima» y obtiene respuestas sobre la «ciudad de

Lima», «jugo de lima», «lima de uñas», y otros. Estas imprecisiones son debidas a los distintos sentidos que tienen las palabras. Las técnicas para resolver este problema se denominan como desambiguación de sentidos de palabras (DSP, en inglés WSD, *word sense disambiguation*). Este término se refiere a un procedimiento basado en la información lingüística para definir el sentido correcto de una palabra, usando el contexto donde se emplee. Se presenta el desarrollo relacionado con un nuevo método de desambiguación de sentidos de palabras usando grandes recursos léxicos, tales como diccionarios explicativos, diccionarios de sinónimos, *WordNet*, entre otros.

Capítulo 9. Describe una técnica, basada en el PLN, útil para la recuperación de información. La recuperación de información con comparación no exacta entre la petición y el documento permite mejorar considerablemente los resultados de la búsqueda, porque en muchos casos el usuario no sabe las palabras exactas que contienen los documentos de su interés, por ejemplo: los documentos que contienen las palabras cristianismo o sacerdotes pueden ser relevantes para la petición «documentos sobre la religión». Se propone un método de recuperación de documentos basado tanto en el uso de los diccionarios grandes de sinónimos y antónimos, como en la aplicación de análisis morfológico automático. El método fue implementado para el español en un sistema de recuperación de textos políticos de la base de datos documental del Senado de la República Mexicana, y aquí se describe brevemente la implementación del sistema.

Capítulo 10. Presenta datos experimentales que muestran que los coeficientes de dos importantes leyes del lenguaje natural —la ley de Zipf y la de Heaps— cambian de un idioma a otro, lo que tiene implicaciones tanto teóricas como prácticas. Por un lado, entender las razones de este hecho puede proporcionar más información sobre la naturaleza del lenguaje humano. Por otro lado, la ley de Heaps tiene aplicaciones prácticas importantes,

como el caso del desarrollo de las bases de datos documentales, para las que permite predecir el tamaño del archivo de índice.

Capítulo 11. Describe una técnica para la compilación de un corpus de textos con ciertas propiedades estadísticas útiles. Para estudiar muchas propiedades de las palabras —coocurrencias y colocaciones, marcos de subcategorización, etcétera— se emplea la investigación estadística de grandes cantidades de textos. Sin embargo, en los corpus tradicionales, aunque sean muy grandes, debido a la ley de Zipf mencionada antes, unas cuantas palabras tienen un número enorme de ocurrencias y ocupan la mayor parte del volumen del corpus, mientras que la inmensa mayoría de las palabras tiene un número estadísticamente insuficiente de ocurrencias. La solución a este problema es el uso del corpus más grande que se ha creado por la humanidad: Internet. Los programas basados en esta idea se llaman corpus virtuales. Sin embargo, éstos presentan problemas tales como sobrecarga de la red, respuesta lenta y —lo peor— resultados no reproductibles, debido a que la Internet cambia constantemente, un resultado obtenido ayer no se puede reproducir hoy. Para combinar las ventajas de los dos tipos de corpus, se propone un corpus que brinda los mismos resultados que un corpus virtual pero almacenados localmente, sin necesidad de descargarlos de la red. Se presenta además una discusión de los problemas que surgen y se describe el sistema para la compilación automática de tal corpus.

Capítulo 12. Expone un método para el enriquecimiento automático de un diccionario de colocaciones, que es una base de datos que incluye combinaciones de palabras de diferentes tipos. También se discuten los conceptos relacionados con las combinaciones de palabras —combinaciones idiomáticas, colocaciones, combinaciones libres. El método se basa en los resultados del análisis sintáctico automático (*parsing*) de oraciones. Se usa el formalismo de dependencias para la representación de los árboles sintácticos, que permite un tratamiento más sencillo de la infor-

mación de compatibilidad sintáctica. Se presenta la evaluación del método de enriquecimiento para el lenguaje español basada en la comparación de los resultados obtenidos automáticamente con los resultados de marcaje manual y, finalmente, se comparan los resultados con los del método que emplea bigramas.

Capítulo 13. Sugiere una forma de evaluar un aspecto de la calidad de los diccionarios explicativos. Esta evaluación consiste en la medición de la semejanza entre los diferentes sentidos de la misma palabra. Las palabras en un diccionario explicativo tienen diferentes significados (sentidos). Proponemos que un buen diccionario tiene los sentidos diferentes bien delimitados y realmente distintos, mientras que un mal diccionario los tiene similares y difíciles de distinguir. La semejanza entre dos sentidos se calcula por el número relativo de palabras iguales —o de sinónimos— en las definiciones del diccionario explicativo. En nuestros experimentos usamos el diccionario explicativo Anaya de la lengua española. Los resultados que obtuvimos demuestran que 10% de los sentidos de la misma palabra usados en Anaya son significativamente parecidos.

Capítulo 14. Muestra cómo se puede construir automáticamente un vocabulario básico para el sistema de definiciones de un diccionario. Un diccionario semántico computacional empleado para los sistemas de inferencia lógica e inteligencia artificial no debe contener círculos viciosos —ciclos— en su sistema de definiciones. Incluso en un diccionario orientado al lector humano tales círculos, cuando son muy cortos, resultan no deseables. Sin embargo, los diccionarios explicativos tradicionales los contienen. Aquí presentamos un algoritmo para la detección de tales ciclos y para la selección de un conjunto mínimo de las palabras primitivas, a través de las cuales se pueden definir todas las demás palabras. Se describe también una herramienta que ayuda al lexicógrafo a elegir tales palabras y a corregir los defectos relacionados con los ciclos en las definiciones del diccionario.

Parte I

Problemas generales del Procesamiento de Lenguaje Natural

Capítulo 2

TAREAS Y APLICACIONES DEL PLN

El recurso más importante que posee la raza humana es el conocimiento, es decir, la información. En la época actual, del manejo eficiente de la información, depende el uso de todos los otros recursos naturales, industriales y humanos.

Durante la historia de la humanidad, la mayor parte del conocimiento se ha comunicado, guardado y manejado en la forma de lenguaje natural —griego, latín, inglés, español, etc. La actualidad no es una excepción: el conocimiento sigue existiendo y creándose en forma de documentos, libros, artículos —aunque todos estos ahora se puedan guardar también en formato electrónico, o sea, digital. Éste es, precisamente, el gran avance: el que las computadoras se hayan convertido en una ayuda enorme para el procesamiento del conocimiento.

Sin embargo, lo que es conocimiento para nosotros —los seres humanos— no lo es para las computadoras. Para ellas son sólo archivos, secuencias de caracteres y nada más. Una computadora puede copiar un archivo, respaldarlo, transmitirlo, borrarlo —como un burócrata que pasa los papeles a otro burócrata sin leerlos. Pero no puede buscar las respuestas a las preguntas en este texto, ni hacer inferencias lógicas sobre su contenido, ni

generalizar ni resumirlo —es decir, hacer todo lo que las personas normalmente hacemos con el texto. Porque no lo puede entender.

Para resolver esta situación se dedica mucho esfuerzo, sobre todo en los países más desarrollados del mundo, al desarrollo de la ciencia que se encarga de habilitar a las computadoras para entender el texto. Esta ciencia, en función del enfoque práctico o teórico, del grado en el que se espera lograr la comprensión y de otros aspectos, tiene varios nombres: procesamiento de lenguaje natural, procesamiento de texto, tecnologías de lenguaje, lingüística computacional. En todo caso, se trata de procesar el texto por su sentido y no como un archivo binario.

El esquema general de la mayoría de los sistemas y métodos que involucran el procesamiento de lenguaje es el siguiente:

- Primero, el texto no se procesa directamente sino se transforma en una representación formal que preserva sus características relevantes para la tarea o el método específico (por ejemplo, un conjunto de cadenas de letras, una tabla de base de datos, un conjunto de predicados lógicos, etcétera).
- Segundo, el programa principal manipula esta representación, transformándola según la tarea, buscando en ella las subestructuras necesarias, etcétera.
- Tercero, si es necesario, los cambios hechos a la representación formal (o la respuesta generada en esta forma) se transforman en el lenguaje natural.

Es decir, para convertir a la computadora en nuestro verdadero ayudante en el procesamiento de textos, se necesita pasar un largo camino de aprendizaje de la estructura de textos y de su formalización; más abajo hablaremos de algunos problemas que se encuentran en este camino. Pero si es tan largo el camino, ¿existe una razón práctica para trabajar en esta área ahora? Sí,

existe, porque con cada paso obtenemos herramientas de gran valor práctico que ayudan en nuestras tareas cotidianas. ¿Para qué sirve el procesamiento automático de lenguaje natural? En la práctica, se puede emplear en un rango de tareas que va desde situaciones muy simples a situaciones muy complejas. Las tareas simples, con el estado actual de la ciencia, ya se pueden realizar, aunque no perfectamente. Las tareas más complejas son la meta a alcanzar en el futuro. Las tareas simples aprovechan los avances de la ciencia, como veremos a continuación.

En este capítulo consideramos algunas de las tareas de PLN, desde las más simples hasta las más complejas, los problemas que se enfrentan durante el procesamiento, las ideas y los métodos que se usan para resolverlos, y las tendencias recientes en estas aplicaciones e ideas. También se discutirá la situación actual de México en el desarrollo de esta ciencia.

2.1 Ayuda en preparación de textos

Este tipo de aplicaciones quizá es conocido hoy en día por toda la gente que ha usado la computadora al menos una vez. Hablamos de las herramientas que proporcionan los procesadores de palabras como Microsoft Word. Aquí, sólo nos interesan las herramientas que emplean el procesamiento complejo de texto y requieren conocimiento lingüístico.

Guiones. La tarea de determinar los lugares donde las palabras se pueden romper para empezar una nueva línea es una de las más simples en procesamiento de textos. Por ejemplo, se puede romper la palabra como *mara-villoso* o *maravillo-so*, pero no *maravil-loso* o *maravillos-o*.

A pesar de ser un problema simple, a veces requiere una información bastante profunda. Por ejemplo, se debe saber cuáles son el prefijo y la raíz de la palabra: *su-bir* y *sub-urbano*, pero no

sub-ir o *su-burbano*. O bien, el idioma de origen de la palabra: *Pe-llicer*, pero *Shil-ler*. También se debe conocer la estructura del documento, ya que no es recomendable usar guiones en los títulos y encabezados.

Ortografía. La tarea de averiguar si una palabra está escrita correctamente o con un error ortográfico es un poco más difícil que la de los guiones. Por lo menos se deben saber todas las palabras del idioma dado. Ya que no es posible conocerlas todas, se necesita saber en primer lugar las formas de las palabras, como *inteligentísimas*, *satisfechos*, *piensen*, etcétera.

Pero para detectar los errores de ortografía, o simplemente de escritura, se debe considerar el contexto de la palabra de una manera a veces bastante compleja. Ejemplos: *Sé que piensen en el futuro. El terremoto causó una gran hola en el mar. Se iba a cazar con su novia en la iglesia bautista*. Para detectar este tipo de errores, la computadora necesita, incluso, entender hasta cierto grado el sentido del texto.

Gramática. Los correctores de gramática detectan las estructuras incorrectas en las oraciones, aunque todas las palabras en la oración estén bien escritas —en el sentido de que son palabras legales en el idioma—, por ejemplo: *Quiero que viene mañana. El maestro de matemáticas, se fue. Me gusta la idea ir a Europa. Fuera magnífico si él venía a la fiesta*.

El problema para detectar los errores de este tipo es que hay una gran variedad de estructuras permitidas y enumerarlas todas resulta muy difícil. Para describir las estructuras de las oraciones en el idioma se usan las llamadas gramáticas formales —conjuntos de reglas de combinación de palabras y su orden relativo en las oraciones.

Estilo. Una tarea más complicada consiste en detectar en el texto los problemas de las palabras correctamente escritas y las oraciones correctamente estructuradas, pero poco legibles, ambiguas, mal estructuradas, inconsistentes en el uso de palabras de

diferentes estilos. Por ejemplo, en un texto científico no se debe usar caló; una carta a un amigo no se construye con oraciones demasiado largas, profundamente estructuradas, ni con muchas palabras científicas.

Un ejemplo de una oración ambigua es «*Detectar los errores en el texto con estructuras complejas*»: sería mejor decir «*Detectar los errores en el texto que tiene estructuras complejas*» o «*Detectar en el texto los errores que tienen estructuras complejas*» o bien «*Detectar a través de las estructuras complejas los errores en el texto*». Para este tipo de procesamiento, en ciertas circunstancias, hay que emplear un análisis bastante profundo.

Hechos y coherencia lógica. Probablemente, en el futuro los correctores de texto serán capaces de encontrar errores como éstos: «*Cuando voy a Inglaterra, quiero visitar París, la capital de este país*». «*Primero vino Pedro y después José; ya que José llegó antes de Pedro, tomó el mejor asiento*». Sin embargo, en el estado actual de procesamiento de lenguaje natural, aún no es posible crear herramientas de este tipo suficientemente desarrolladas como para ser útiles en la práctica.

2.2 Búsqueda de información

Vivimos en la época de la información. Hace tiempo, la pregunta principal sobre la información era «*cómo puedo obtener la información*» y la respuesta se buscaba en la naturaleza, en la sociedad, etc. Ahora la pregunta principal es más bien «*dónde puedo encontrar la información*», es decir, ya sé que está en algún lado —incluso en mi propia computadora— pero ¿dónde está? ¿Cuál de los miles de archivos en mi computadora contiene lo que busco?

La aplicación de procesamiento de lenguaje natural más obvia y quizá más importante en la actualidad, es la búsqueda de información (llamada también recuperación de información).

Por un lado, Internet y las bibliotecas digitales contienen una cantidad enorme de conocimiento que puede dar respuestas a muchísimas preguntas que tenemos. Por otro lado, la cantidad de información es tan grande que deja de ser útil al no poder ser encontrada fácilmente. Hoy en día la pregunta ya no es «¿si se sabe cómo...?» sino: «ciertamente se sabe, pero ¿dónde está esta información?».

Técnicamente, rara vez se trata de decidir cuáles documentos (así se llama a los archivos o textos en la recuperación de información) son relevantes para la petición del usuario y cuáles no. Usualmente, una cantidad enorme de documentos se pueden considerar relevantes en cierto grado, unos más y otros menos. Entonces, la tarea consiste en medir el grado de relevancia, para proporcionar al usuario primero el documento más relevante; si no le sirvió, el segundo más relevante, y así sucesivamente.

El problema más difícil de la recuperación de información es, sin embargo, no de índole técnica sino psicológica: entender cuál es la necesidad real del usuario, por qué formula su pregunta. Este problema se complica, ya que no existe un lenguaje formal en el cual el usuario pueda formular claramente su necesidad. La dirección más prometedora para resolver este problema es, nuevamente, el uso de lenguaje natural.

Las técnicas más usadas actualmente para la recuperación de información implican la búsqueda por palabras clave: se buscan los archivos que contienen las palabras que el usuario teclea. Es decir, la representación formal usada es el conjunto de las cadenas de letras (palabras), usualmente junto con sus frecuencias en el texto (número de ocurrencias). La claridad matemática de la tarea provocó un gran avance en la teoría de estos métodos. Las ideas más usadas son los modelos probabilísticos y los procedimientos iterativos e interactivos, es decir, los que tratan de adivinar qué necesita el usuario preguntándole cuáles documentos le

sirven. Una excelente revisión del estado del arte en este campo se puede encontrar en (Baeza-Yates y Ribeiro-Neto, 1999).

Sin embargo, los métodos que involucran sólo las palabras (como cadenas de letras) pero no el sentido del texto, son muy limitados en su capacidad de satisfacer la necesidad informática del usuario, es decir, de hallar la respuesta a la pregunta que tiene en mente. Se puede mejorar mucho aplicando las siguientes operaciones, desde las más sencillas hasta las más complejas:

– Coincidencia de las formas morfológicas de palabras: buscando *pensar*, encontrar *piénsalo*.

  Este problema es bastante simple de resolver en el lenguaje inglés, al cual se dedica la mayor parte de la investigación en el mundo. Sin embargo, para el español, se convierte en un problema moderadamente serio, debido a la gran variedad de las formas de las palabras en español.

  Los métodos de la morfología computacional —la rama del procesamiento de lenguaje natural que se encarga del modelado de las formas morfológicas de palabras— varían, y van desde el uso de diccionarios que especifican las formas para cada palabra, hasta las heurísticas que ayudan a adivinarlas (Gelbukh, 2000; 2003).
– Coincidencia de los sinónimos, conceptos más generales y más específicos: buscando *cerdo*, encontrar *puerco*, *mascota*, *animal*, etcétera.

  Este problema no depende de cuál es la lengua de la que se trata (es importe tanto para el inglés como para el español), aunque los diccionarios que se usan sí son específicos de cada lengua.

  La idea principal es, como ya se dijo, el uso de diccionarios jerárquicos (Gelbukh *et al*., 1999a; Gelbukh *et al*., 2002c), que especifican los sinónimos en el mismo nivel del árbol, y

los conceptos más específicos debajo de los conceptos más generales. Uno de los problemas que aún no tienen solución adecuada es el de medir las distancias en este árbol: ¿qué tan parecida es la palabra *cerdo* a *puerco*? ¿y a *mascota*? ¿*animal*? ¿*objeto*?

Una generalización de esta idea son los diccionarios de las palabras conceptualmente relacionadas, por ejemplo, *cerdo* y *tocino*; o *sacerdote*, *Biblia*, *iglesia* y *rezar*. Aquí, el problema de la medición de distancia es aún más difícil.

– Tomar en cuenta las relaciones entre las palabras en la petición del usuario y en el documento: buscando *estudio de planes*, rechazar como no relevante *planes de estudio*.

Para lograr este grado de calidad, se necesita reconocer (automáticamente) la estructura del texto y representarla en forma que permita la comparación necesaria, por ejemplo, en la forma de grafos conceptuales (Montes y Gómez *et al*., 2001a).

Recientemente, el desarrollo de la solución al problema de búsqueda de información avanzó hacia una perspectiva diferente: generación automática de respuestas. La idea es la siguiente: en lugar de presentarle al usuario el documento completo donde probablemente se contiene la respuesta a su pregunta (por ejemplo, ¿cuándo fue la revolución mexicana?), simplemente darle la respuesta (en este caso, generar «En 1910-1917» basándose en la información encontrada en los textos).

Una de las técnicas más usadas para esto es la extracción de información: transformación de algunas partes de los textos libres en un formato de base de datos, por ejemplo: evento — fecha, artículo — lugar — precio, etc. Otra técnica posible es el razonamiento lógico sobre las relaciones encontradas en el texto.

2.3 Manejo de documentos

Un área relacionada con la búsqueda de información es el área de gestión inteligente de documentos, que incluye tanto la búsqueda de documentos, como su organización y la navegación por sus conjuntos.

Búsqueda de documentos

Hay tres métodos principales que nos ayudan a enfrentar este problema: la estructuración de la información, los motores de búsqueda y las combinaciones de estos dos.

La estructuración de información es el método tradicional. Desde las bibliotecas de la antigua Babilonia se conoce la técnica de ordenar alfabéticamente. Uno de los mejores modos de estructuración inventado desde aquellos tiempos consiste en organizar la información en un árbol de rubros y subrubros, lo que se representa en las computadoras modernas con carpetas y subcarpetas. Por ejemplo, en una carpeta pongo los documentos sobre las finanzas, en otra los que tienen que ver con los empleados —dividiendo la carpeta en subcarpetas por departamentos— y en una tercera los referidos al equipo de cómputo —dividiéndola en equipo mayor, equipo de oficina y accesorios.

Sin embargo, este modo de estructuración presenta muchos problemas cuando se trata de una cantidad significativamente grande de documentos. Por ejemplo, ¿dónde pongo un documento que dice que María Pérez del departamento de compras adquirió una nueva impresora en 10 mil pesos? Como se trata de María Pérez, lo puedo guardar en la subcarpeta de *Compras del personal*. Pero al final del año ¿dónde lo busco para calcular los gastos en cómputo de mi empresa? O bien, lo puedo guardar en *Cómputo*, pero luego ¿dónde busco las actividades de María Pérez?

Otro problema que presenta la estructuración jerárquica, es el modo de dividir el mundo en rubros. ¿Qué pasa, si, por ejemplo, después de un largo tiempo necesito encontrar un documento del que sólo recuerdo que trata de una mujer que compró un equipo de cómputo? Primero, tengo muchas subcarpetas de la carpeta *Cómputo*, ¿dónde busco mi documento? Segundo, la subcarpeta *Personal* se divide en *Compras*, *Contaduría*, *Dirección*, etc. y no en *Mujeres* y *Hombres*.

Se ve que entre más profundamente elaboro mi clasificación, más difícil resulta encontrar los documentos al no saber exactamente dónde los puse. Es una señal de un problema serio con el método.

Los motores de búsqueda ayudan a evitar los problemas mencionados arriba. A diferencia de los métodos de estructuración de información —que se aplican también a los documentos en papel— estos motores sólo se aplican a información almacenada en formato electrónico.

Un motor de búsqueda lee todos los textos de los documentos en una colección dada y encuentra los que corresponden a la petición. Por eso, se encontrará exactamente el mismo documento usando cualquiera de las siguientes peticiones: «María Pérez», «impresora» o «10 mil pesos».

A pesar de ser mucho más inteligentes que las estructuras jerárquicas de almacenamiento, los motores de búsqueda presentan sus propios problemas. Estos problemas dieron inicio a toda una ciencia denominada *recuperación de información* (Baeza-Yates y Ribeiro-Neto, 1999).

Algunos de estos problemas están relacionados con el proceso de lectura de los documentos. Para decidir si el documento corresponde a la petición del usuario, la computadora debe, en cierta medida, entender el texto. A una mejor comprensión del sentido del texto y las relaciones entre las palabras por parte de

la computadora, corresponde un mejor funcionamiento del motor de búsqueda.

Otros problemas son semejantes a los problemas de los métodos tradicionales —principalmente a los casos en los que el usuario no puede formular su pregunta de tal forma que la respuesta se contenga en el documento de manera exacta. Por ejemplo, preguntando por «una mujer», «equipo de cómputo» o «presupuesto», el usuario espera encontrar el documento que literalmente menciona «María Pérez», «impresora» y «10 mil pesos». En este capítulo, se presentan algunas soluciones inteligentes a esta clase de problemas.

Cabe mencionar que los sistemas que se ofrecen en el mercado no proporcionan en realidad soluciones adecuadas a los problemas mencionados. Usualmente, sólo presentan una interfaz gráfica más atractiva y quizá más cómoda e integrada que la interfaz estándar de las carpetas de Windows, pero nada o muy poco más.

Los métodos combinados existen tanto para los documentos tradicionales (en papel) como para los documentos en formato electrónico. Desde hace mucho tiempo existen catálogos que permiten, efectivamente, guardar un documento tradicional —digamos, un libro— no en un sólo lugar sino —virtualmente— en varios lugares al mismo tiempo. Con fichas, se puede encontrar el mismo documento (mejor dicho, su ficha) tanto en el rubro *Personal* como en *Cómputo*. Este es un paso un poco más inteligente desde la estructura jerárquica de almacenamiento hacia la búsqueda.

Por otro lado, las fichas descriptivas pueden proporcionar información adicional que no está escrita en el documento, o proporcionarla en una mejor forma, más entendible. Esto da pie para que los motores de búsqueda modernos se aprovechen de las formas más tradicionales del manejo de documentos. En este capítulo también se presenta un sistema de búsqueda combinado.

Representación y navegación por los documentos

Entre más poderoso sea el motor de búsqueda y mayor la colección o flujo de documentos, mayor será también el papel del usuario en la filtración de los resultados de la búsqueda. Digamos, en una colección de 10 mil documentos, el motor de búsqueda puede encontrar 50 que son relevantes para la petición. Pero es difícil leer por completo todos éstos para decidir si es realmente lo que busco.

Es muy importante, entonces, el modo en el que el programa describe al usuario cada documento. El nombre del archivo o el título en muchos casos son insuficientes para que el usuario tome una decisión fácil y rápidamente.

Más aún, con una buena presentación de los documentos y las relaciones entre ellos, se puede evitar totalmente el proceso de búsqueda, volviendo —en un nivel más alto— a los simples y familiares esquemas de la estructuración jerárquica. En este caso, sin embargo, la jerarquía es inteligente. Primero, la computadora puede clasificar los documentos automáticamente. Segundo, la jerarquía se genera en el proceso de navegación y tomando en cuenta el perfil de los intereses del usuario específico, de tal manera que el usuario —en el modo interactivo— tiene control sobre esta jerarquía. Es decir, para diferentes usuarios o en diferente tiempo el sistema inteligente construye estructuras distintas.

Los métodos de generación automática e interactiva tienen la ventaja de que pueden construir la estructura de un conjunto de documentos previamente no estructurado o estructurado con criterios diferentes a los que quiere aplicar el usuario. Esto proporciona al usuario una información nueva y valiosa sobre el conjunto o flujo de documentos.

En el resto de este capítulo se presentarán algunas soluciones que están bajo desarrollo en el Laboratorio de Lenguaje Natural del CIC-IPN. Primero, se describirán los métodos inteligentes de

búsqueda, tanto los que usan únicamente el texto del documento, como los que combinan la información tabular (base de datos) o descriptiva con la información textual. Después, se darán unos ejemplos de la representación inteligente del contenido del documento y del conjunto de documentos parecidos. Luego, se presentará la idea de jerarquía inteligente para la navegación por los conjuntos grandes de documentos. Finalmente, se darán las conclusiones.

2.4 Gestión inteligente de documentos

Búsqueda inteligente de documentos

Como ya se mencionó, hay dos tipos de problemas en la búsqueda de textos relevantes: problemas de comprensión del texto por la máquina y problemas de la comparación aproximada entre la petición del usuario y el texto.

En cuanto al primer problema, en el Laboratorio estamos desarrollando métodos para la comprensión automática de texto. Éstos son de diferentes niveles de profundidad: reconocimiento de formas morfológicas de palabras (Gelbukh, 2000), reconocimiento de la estructura de las oraciones (Bolshakov, 2002), representación semántica de la petición del usuario y de los textos de documentos (Montes y Gómez *et al*., 2001b). Algunos temas mencionados se describen detalladamente en los siguientes capítulos.

Con los resultados de la aplicación de estos métodos, el motor de búsqueda puede encontrar por la petición «pensar» el texto que contiene «pienso» (morfología); o por la petición «introducción en programación lógica» el texto que contiene «introducción detallada en programación lógica», pero omitir el texto que contiene «introducción lógica en programación orientada a objetos» (representación semántica).

En cuanto al segundo problema —la comparación aproximada entre la petición y el texto— en el Laboratorio estamos desarrollando métodos que permiten usar generalizaciones: por ejemplo, por la petición «equipo de cómputo» encontrar el texto que contiene «impresora». Para esto, usamos un diccionario grande que puntualiza las relaciones entre las palabras más específicas y más generales. Un ejemplo del motor de búsqueda desarrollado en el Laboratorio que usa tal diccionario se muestra en la ilustración 1. En esta ilustración, la petición «motor de combustión interna»[1] (seleccionada en la columna izquierda) se está ejecutando sobre el conjunto de reportes. Como se ve en la columna derecha, se encontrarán los textos que contienen las palabras *acelerador*, *biela*, etcétera.

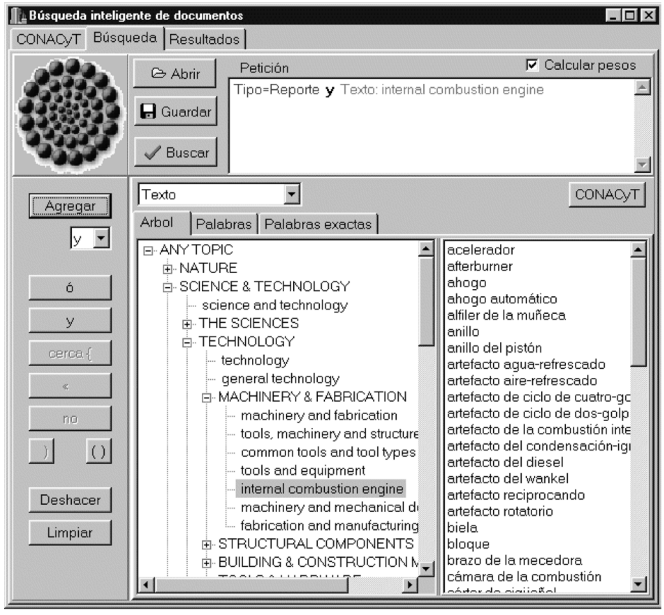


Ilustración 1. Motor de búsqueda con comparación entre palabras más generales y más específicas.

[1] En la versión actual, los conceptos se muestran en inglés para facilitar su promoción internacional. Está disponible también la versión totalmente en español.

Una generalización natural de la comparación aproximada es la búsqueda por un documento ejemplo. En este caso, el usuario no tiene que proporcionar las palabras clave de la búsqueda sino sólo un documento ya existente. El sistema desarrollado en nuestro Laboratorio encuentra todos los documentos parecidos. La comparación es aproximada: si el documento ejemplo contiene palabras «monitor» y «teclado», el programa puede encontrar otro que contenga las palabras «impresora» y «escáner», pues todas ellas están bajo el nodo «dispositivos de cómputo».

Combinación de la información tabular y textual

La búsqueda en los textos en el español libre es una herramienta muy poderosa pero no se puede aplicar correctamente a la información exacta, como las fechas, presupuestos, etc. Por ejemplo:

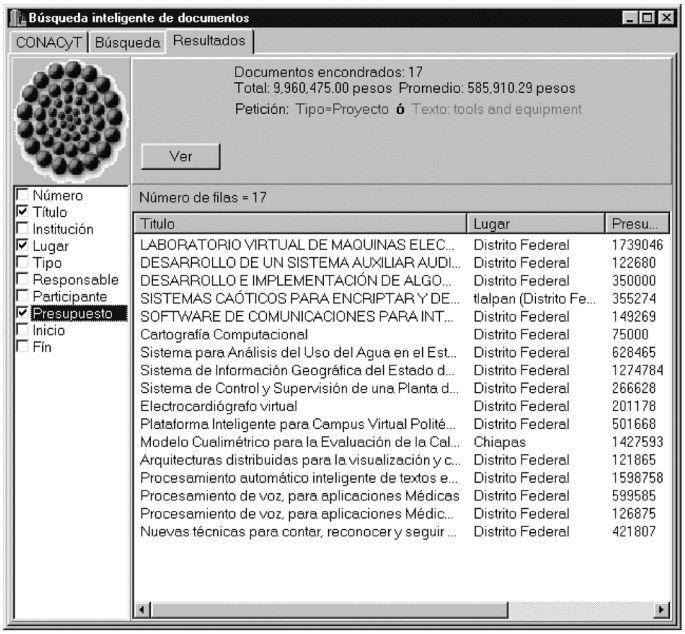


Ilustración 2. Resultado de búsqueda en una base combinada.

El resultado de la búsqueda se puede proporcionar mostrando los renglones relevantes del texto del documento, o a través de los valores especificados en sus fichas descriptivas, véase la Ilustración 3.

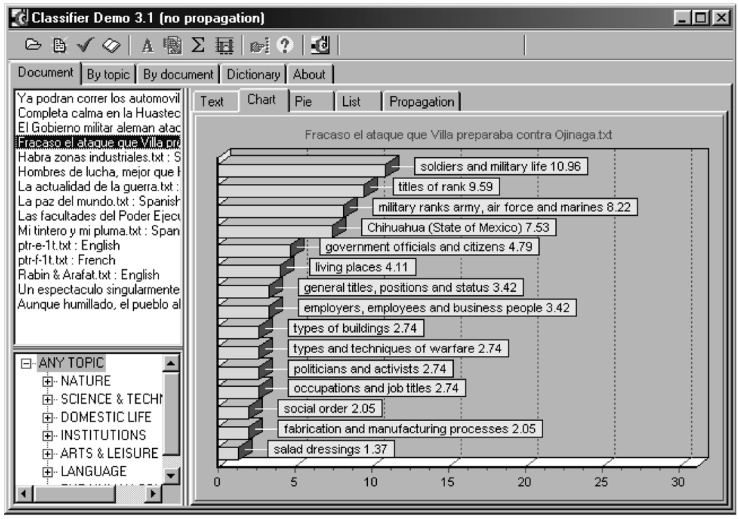


Ilustración 3. Resumen temático del documento.

Representación inteligente de documento

Hay dos situaciones en las que se necesita una representación del documento mucho más concisa que su propio texto completo.

Documentos largos. El usuario puede querer familiarizarse rápido con el documento sin leerlo completo. La mejor ayuda que la computadora puede proporcionar al usuario es construir el resumen del documento. Desgraciadamente, los programas existentes de este tipo funcionan todavía con un nivel de calidad inadecuado. Un ejemplo claro de esto es la herramienta Autorresumen proporcionada en Microsoft Word. Están en desarrollo, sin embargo, métodos mucho más inteligentes para construir resúmenes.

Otra posible solución —mucho más factible hoy en día— es el resumen temático. Este tipo de resumen no responde a la pregunta «¿qué dice este documento?» sino «¿sobre qué es este documento?» o bien, «¿qué temas trata?» (Gelbukh *et al.*, 1999). En la ilustración 3 se muestra en forma gráfica[2] el resumen temático del documento «Fracasó el ataque que Villa preparaba contra Ojinaga». También se puede ver este resumen en forma textual: «Este documento es principalmente sobre los soldados y vida militar y sobre los títulos de rango; también menciona los temas de la armada, fuerza aérea y marina, así como del Estado de Chihuahua (México)».

Uno de los primeros pasos para la clasificación, búsqueda y comprensión de documentos es determinar de qué temas trata un documento determinado. El sistema usa un diccionario jerárquico para hallar los temas principales, para comparar los documentos con un aspecto temático y para buscarlos por sus temas; todo se centra en la detección de temas de documentos. Las palabras se asocian con los nodos terminales del diccionario jerárquico y «votan» por algún tema.

Para detectar esto se hace análisis de frecuencias de palabras usando el diccionario jerárquico de conceptos, que se puede ver en ilustración 4. Las palabras «votan» por algún tema. La estructura del diccionario en forma de árbol permite hacer propagaciones de temas para los nodos no-terminales. En la parte derecha de la ilustración 4 están presentes las palabras que se asocian con el nodo terminal. En este caso están en inglés, pero el sistema por el momento soporta también el español y el francés. La descripción detallada se encuentra en (Gelbukh et al., 1999).

[2] Recordamos al lector que ahí se muestran los nombres de conceptos en inglés pero está disponible la versión en español.

Presentamos un fragmento del diccionario de conceptos:

ANY TOPIC
└ SCIENCE & TECHNOLOGY
└ THE SCIENCES
└ COMPUTERS
├ languages and programming
└ tools and hardware

etcétera.

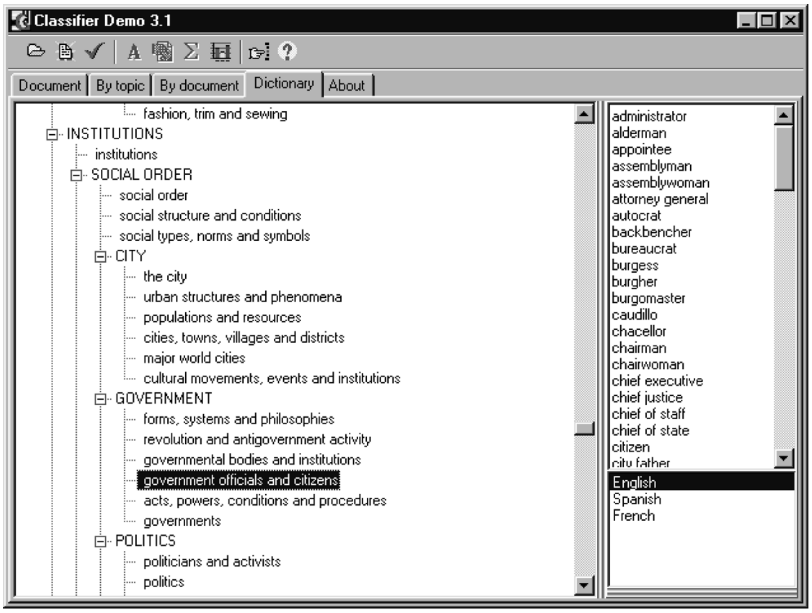


Ilustración 4. Diccionario jerárquico de conceptos.

La estructura en forma de árbol del diccionario permite hacer propagaciones de temas para determinar la contribución de los nodos no-terminales. Los conceptos no-terminales del árbol están en inglés, pero las listas de palabras asociadas con ellos pueden estar en cualquier lenguaje. El sistema por el momento funciona para el inglés, el español y el francés.

Como un ejemplo, las palabras que corresponden al nodo terminal *languages and programming* son:

> *programa, programador, programando, prueba, recopilador, rendimiento, residente de memoria, retorno, retrofit, rpg, shareware, sistema operativo*, entre otras.

En la ilustración 3 se presenta el resultado del funcionamiento del sistema. El sistema determinó los temas principales del documento, que están mostrados como un histograma. En la parte izquierda superior de la pantalla se encuentra la lista de los archivos procesados —«*Ya podrán correr los automóviles...*», «*Completa calma...*», «*El Gobierno militar...*», «*Rabin & Arafat*», etc. En la parte inferior izquierda se presenta el árbol jerárquico de conceptos con el nodo raíz *ANY TOPIC* (*cualquier tópico*) seleccionado. Nótese que si se elige un nodo en el árbol que sea distinto del nodo raíz *ANY TOPIC*, el sistema empieza a funcionar solamente tomando en cuenta los nodos debajo del nodo elegido, es decir, si elegimos el nodo *NATURE* (*naturaleza*), el sistema va a determinar los temas principales o clasificar los documentos solamente tomando en cuenta las palabras relacionadas con naturaleza.

Se nota que para el texto llamado «*Rabin & Arafat*», el tema principal es «soldados y vida militar», lo que corresponde a su contenido. Para detectar esto se hizo el análisis de frecuencias de palabras («votación») según se describió anteriormente.

Navegación por los documentos. Cuando el usuario navega por un conjunto grande de documentos, es indispensable que los documentos —digamos, en la lista de documentos presentada al usuario— se representen de una manera muy concisa, para que la lista quepa en la pantalla y sea manejable. Usualmente, para esto se utiliza el nombre del archivo o el título del documento. Un resumen temático, como en la ilustración 3 —o bien, una lista de palabras clave equivalente al resumen temático— construido

automáticamente, es una forma adicional (quizás mejor) de representación de los documentos en el conjunto.

Representación inteligente de un conjunto de documentos

A diferencia de la representación concisa de un documento en el conjunto, la tarea de representación inteligente de un conjunto de documentos (por ejemplo, los resultados de búsqueda) implica, primero, la representación integral del conjunto (como un objeto) y, segundo, la representación de la estructura interna del conjunto —de las relaciones de los documentos que lo constituyen.

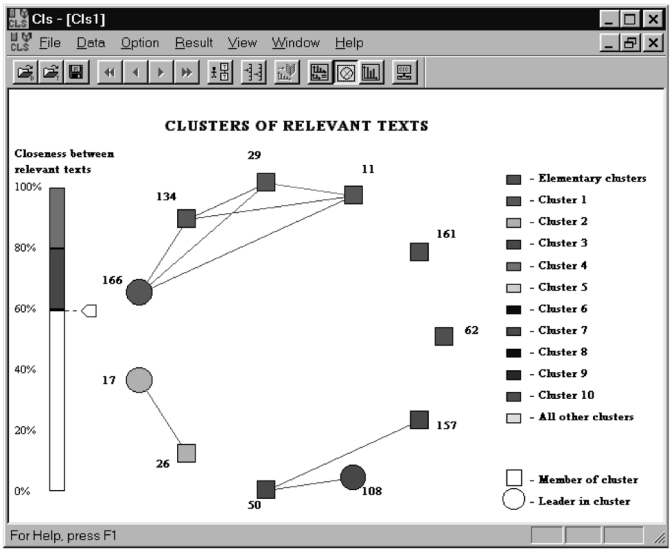


Ilustración 5. Estructura de un conjunto de documentos. Los círculos representan los documentos típicos en sus subconjuntos.

La manera tradicional de representación de un conjunto de documentos es una lista, donde los documentos individuales se representan, digamos, por sus títulos. La lista puede ser ordenada por relevancia, pero no puede representar las relaciones entre los

documentos. Tampoco se puede dar automáticamente un título significativo a la lista entera (como una entidad).

Esta representación tiene ciertas ventajas: es simple, es intuitiva, los usuarios están bien acostumbrados a ella y no necesitan entrenamiento adicional alguno para usarla. Sin embargo, tiene también desventajas importantes. La principal desventaja es la falta de diversidad, la cual dificulta la exploración de diferentes opciones que le ofrece al usuario una colección dada.

En efecto, muchas colecciones de documentos contienen grupos de documentos muy parecidos —por ejemplo, noticias sobre un evento. En la lista de los resultados de una consulta el grupo formalmente más relevante aparece al inicio, sin darle oportunidad al usuario de ver otros documentos, quizá más interesantes. La representación inteligente del conjunto alivia este problema.

La herramienta[3] que se presenta en la ilustración 5, divide automáticamente la colección de los documentos en grupos (*clusters*) de tal manera que los documentos que pertenecen al mismo grupo son, en cierto grado, parecidos entre sí. En la ilustración se muestran en un diagrama circular tres grupos de documentos, además de dos documentos que resultaron estar aislados (cada documento se representa con un pequeño cuadrado o círculo). En cada conjunto, se puede ver las relaciones de proximidad temática que existen entre sus documentos.

En cada grupo hay un documento marcado con círculo. Este documento es el más típico en su grupo, es decir, en promedio es más parecido a todos los demás documentos de su grupo. Este documento, entonces, puede servir como representante de su grupo cuando éste se debe constituir como una entidad. Uno de los posibles usos de ese documento representativo se describe en la siguiente sección.

[3] Esta herramienta profesional es para el usuario experto. Está bajo desarrollo la versión para el usuario común, donde los conjuntos se representan en una forma más familiar.

Otra técnica de descripción de un grupo de documentos es su resumen temático conjunto. Esto se concibe como si el programa construyera un sólo documento grande a través de la unión de todos los documentos en el conjunto y presentara sus temas principales.

Navegación inteligente por los conjuntos de documentos

La navegación por un conjunto de documentos es importante en dos casos: exploración del tema y precisión de búsqueda.

La exploración del tema es primordial cuando el usuario no conoce bien el tema y quiere explorarlo de modo interactivo. Digamos, para automatización de su oficina el gerente quiere saber más de cómputo: qué tipos de equipo hay, en qué le pueden ayudar, cómo se usan, etcétera.

La precisión de la búsqueda es necesaria cuando hay demasiados documentos encontrados, como en el caso del resultado de la petición para el motor de búsqueda. Digamos, el usuario busca por la frase «equipo de cómputo» y encuentra 10 mil documentos. El encontrar los documentos necesarios dentro de ese grupo es una tarea muy parecida a la anterior: el usuario tiene que explorar el conjunto de documentos encontrados para entender mejor lo que específicamente le interesa.

Uno de los métodos desarrollados en nuestro Laboratorio para la solución de este problema se basa en las técnicas de búsqueda, representación concisa y agrupamiento de documentos, presentadas en las secciones anteriores, y funciona de la siguiente manera.

En la etapa inicial de la navegación, el usuario puede especificar una petición usando el lenguaje y el diccionario jerárquico, aplicándola a una base de documentos específica o bien a Internet. Los documentos encontrados se presentan en

resúmenes temáticos, es decir, muestran sus temas principales. Si el conjunto de documentos es demasiado grande, el sistema aplica el proceso de agrupamiento de documentos.

Por ejemplo, el sistema puede informar al usuario que encontró 10 mil documentos, que se dividen en cinco grandes grupos. Cada grupo está representado por su documento típico, o bien por su resumen temático conjunto: por ejemplo, el primer grupo se representa por el documento *informe_2000.doc* (que se puede ver con un clic del ratón) y trata de *finanzas* y *presupuesto*, el segundo se representa por el documento *proyecto_IMP.doc* y trata de *petróleo* y *altas tecnologías*, etcétera.

El sistema proporciona al usuario experto la posibilidad de cambiar los parámetros que afectan el proceso de agrupamiento. Uno de los elementos de control se puede ver en la escala en la parte izquierda de la ilustración 5: con ésta, el usuario puede aumentar o disminuir el número de grupos en la división del conjunto (es decir, elegir el agrupamiento más fino o más grueso). También se puede cambiar el modo de comparación usado para el agrupamiento, etcétera.

Basándose en la información que le presenta el sistema sobre cada grupo, el usuario puede elegir uno (o varios) de los grupos encontrados que responden mejor a sus intereses. Desde este momento, el proceso se puede repetir: el nuevo grupo se divide en subgrupos más finos, el usuario elige uno de estos subgrupos y realiza el mismo procedimiento hasta que encuentra los documentos necesarios o el conocimiento que busca.

De este modo, el programa permite al usuario generar una jerarquía sobre el conjunto de documentos de manera interactiva (si el usuario lo prefiere) o automática. El proceso de navegación es muy parecido a la navegación familiar por las subcarpetas de Windows, pero es, en este caso, inteligente, y toma en cuenta los intereses individuales del usuario.

Categorización automática de documentos

La sección anterior describe el agrupamiento y división de un conjunto de documentos en el modo interactivo. Este modo se aplica cuando el usuario no tiene conocimiento previo sobre la colección de documentos, o bien cuando formula una nueva petición para el motor de búsqueda, una petición que no había formulado en ocasiones anteriores. Sin embargo, en la práctica, es frecuente la situación en la que la misma búsqueda o el mismo procedimiento de división se aplican a una colección de documentos que cambia con el tiempo, es decir, a un flujo de documentos. Por ejemplo, el correo que llega a una oficina, y que se debe clasificar para girarlo en la división de compras, división de vinculación o a la dirección.

Las herramientas desarrolladas en el Laboratorio permiten solucionar este problema de manera automática. Para esto, se construyen (en forma similar a la formulación de peticiones) las descripciones temáticas de los intereses de cada destinatario, y el sistema aplica los métodos matemáticos de clasificación para decidir adónde se debe enviar cada documento. El sistema proporciona un conjunto de herramientas que ayudan a la construcción óptima de las descripciones temáticas (Alexandrov *et al.*, 2000a).

2.5 Interfaces en lenguaje natural

Hoy en día, la interacción con las computadoras requiere que el usuario aprenda «cómo usar» la computadora: cómo comunicarle las órdenes o peticiones y cómo interpretar los símbolos que muestra en su pantalla. Una desventaja de esta circunstancia es que la interacción con la computadora no es fluida, por el contrario, es lenta y complicada, requiere la atención completa del usuario —por ejemplo, ocupa sus manos para teclear y sus ojos

para mirar la pantalla. Existe otra desventaja aún más grave: el uso eficiente de las computadoras requiere de conocimientos especiales que la mayor parte de la gente no posee, lo que le hace imposible disfrutar de todas las oportunidades y comodidades con las cuales las computadoras pueden revolucionar nuestra sociedad y nuestra vida cotidiana.

Un modo de interacción mucho más eficiente y cómodo para el hombre es el habla. La habilidad de escuchar, hablar y conversar con las personas de la misma manera en que lo hacemos nosotros, puede convertir a las computadoras —o sea, a los robots— en nuestras verdaderas ayudantes, sirvientes, amigas y colaboradoras.

Como ya hemos visto, en los apartados anteriores, lograr que las computadoras puedan escuchar, hablar y conversar como las personas no es fácil. Involucra todas las ramas de la lingüística computacional: el procesamiento de voz para que puedan escuchar y pronunciar las palabras, el análisis y la generación de las oraciones del lenguaje, el manejo de estrategias complejas de diálogo, la comprensión de lenguaje y el razonamiento lógico sobre las situaciones de las cuales se platica. Aunque no se puede decir que estamos cerca de lograr estos objetivos, con nuestro trabajo cotidiano de investigación y desarrollo tecnológico cada vez nos acercamos más a ellos.

Las computadoras están entrando en todos los campos de nuestra vida cotidiana: en las oficinas, en las tiendas, en las escuelas, en los servicios públicos. Sin embargo, la gran mayoría de la gente no tiene la preparación adecuada para usarlas y nunca la tendrá, por una simple cuestión de economía. Resulta más conveniente que las máquinas se adapten al modo de comunicación de las personas, a que todas las personas (sólo en el mundo hispanohablante son 400 millones), generación tras generación, aprendan cómo usar las máquinas —que aprendan, por ejemplo, el SQL para formular con precisión sus preguntas.

De situaciones como estas han surgido las ideas, ya muy conocidas, de las películas de ciencia ficción, en donde las personas pueden hablar con las máquinas (o sea, los robots) como hablarían con sus sirvientes humanos, dándoles órdenes en la forma cotidiana y escuchando sus respuestas.

Respecto al hecho de darles órdenes, no se trata de pronunciar los comandos especiales que normalmente escogeríamos del menú: *abrir*, *edición*, *copiar*, *guardar*, *salir* (de forma similar a como se le habla a un perro). Se trata de hablarle a la máquina como hablaríamos a un ser humano.

Un tipo específico de interfaces en lenguaje natural consiste en preguntas complejas a una base de datos (Pazos *et al*., 2002). Como ejemplo, podemos mencionar el sistema TRAINS, desarrollado en la Universidad de Rochester en Estados Unidos por James Allen. Este sistema vende los boletos de tren. El cliente —que puede ser cualquier persona sin ningún conocimiento sobre las máquinas— llama por teléfono a la estación de trenes y formula su necesidad: *tengo que ir mañana en la tarde a Nueva York*. El programa —sin que el cliente alcance siquiera a notar que habla con una máquina y no con una persona— descifra la pregunta e internamente la traduce a SQL, para ejecutar la búsqueda en su base de datos. Después, el programa conduce (por teléfono) el diálogo con el usuario, explicándole los precios y las condiciones, escuchando sus preguntas o comentarios sobre qué boleto le conviene más, etc. Finalmente, si llegan a un acuerdo, le reserva el boleto. Todo eso, enfatizamos nuevamente, no requiere del cliente ningún conocimiento previo sobre el manejo de los programas, sino sólo el manejo natural de lenguaje que cotidianamente usa para hablar con otras personas.

El problema más importante de este tipo de aplicaciones es que —a diferencia de las aplicaciones en la recuperación de información— se requiere entender exactamente la intención del usuario, ya que el costo del error puede ser muy alto. Si el robot

entiende incorrectamente el comando, pude realizar alguna acción destructiva o peligrosa. Si la pregunta a la base de datos se malentiende, la información proporcionada resultará incorrecta, lo que también puede causar graves consecuencias.

Por lo tanto, las interfaces en lenguaje natural requieren de representaciones de información más detalladas y complejas, así como de un análisis lingüístico más preciso y completo (Sag y Wasow, 1999).

2.6 Traducción automática

La traducción automática ha motivado el desarrollo de la lingüística computacional desde sus inicios en los años sesenta. En el mundo contemporáneo, cada vez más globalizado, es una tarea de enorme importancia para la raza humana, ya que permitirá romper las barreras de lenguaje y habilitar la comunicación fácil y fluida entre la gente de diferentes países y diferentes culturas. Ofrecerá a todos los pueblos del mundo el acceso fácil a la información escrita en las lenguas más desarrolladas —como el inglés, español, alemán, o francés— en los que están escritos los libros y revistas más importantes y en los que se difunden la mayor parte de las noticias.

Los creadores de los primeros programas de traducción automática se guiaron por una simple idea: una computadora puede sustituir, con gran velocidad, las palabras de un idioma con las palabras de otro, generando así la traducción. Sin embargo, esta sencilla teoría fracasó por completo: el texto generado no era legible, ni siquiera se podía entender. Los primeros estudios serios en lingüística computacional comenzaron con el análisis de las causas de este fenómeno.

Una razón obvia para el fracaso de la traducción mediante simple sustitución son las diferencias en el orden de palabras entre dos lenguajes dados. Por ejemplo, *an interesting book* se traduce

como *un libro interesante* y no como **un interesante libro*. Pero la tarea de traducción automática presenta un problema mucho más difícil de combatir: la ambigüedad. Digamos, para traducir la oración *John took a cake from the table and ate it* se necesita entender qué comió Juan —*la mesa* o *el pastel*— ya que si esto no se entiende tampoco se puede elegir la variante correcta al traducir la palabra *it*: *Juan tomó el pastel de la mesa y ¿la* o *lo*? *comió*. Sin entender la situación que describe el texto, es muy difícil tomar tal decisión. Por ejemplo, cambiando sólo una palabra, obtenemos una oración para la cual la correcta selección entre «*la*» y «*lo*» es contraria a la anterior: *John took a cake from the table and cleaned it*, *Juan tomó el pastel de la mesa y ¿la* o *lo*? *limpió*. Lo mismo sucede al revés. Para traducir el texto «*Juan le dio a María un pastel. Lo comió.*», hay que elegir entre las variantes *He ate it, She ate it, It ate him, She ate him*, entre otras.

Con esto se demuestra que la manera correcta de traducir un texto consiste en lo que hace un traductor humano: entenderlo —lo que en el caso de la traducción automática corresponde al análisis automático de lenguaje— y luego generar el texto con el mismo sentido en otro idioma —equivalente a la generación automática del texto. Aunque hoy en día esto no es posible en su totalidad, el desarrollo de la lingüística computacional —de los métodos de análisis y generación automática de textos— lo hace cada vez más factible.

Históricamente, el sueño de la traducción automática (en aquellos tiempos entre los idiomas ruso e inglés) motivó las primeras investigaciones en lingüística computacional. Como ya mencionamos, a primera vista, la traducción parece ser un trabajo bastante mecánico y aburrido, que puede fácilmente hacer la máquina: sustituir las palabras en un lenguaje con sus equivalentes en otro.

Sin embargo, con los avances en los programas de traducción se hizo cada vez más obvio que la tarea no es tan simple. Esto

se debe, en parte, a las diferencias entre los lenguajes, que van desde las muy obvias (por ejemplo, que el orden de las palabraza diferente), hasta las más sutiles (el uso de expresiones distintas y diferente estilo).

El esquema general de prácticamente cualquier traductor automático es (de acuerdo con el esquema expuesto más arriba) el siguiente:

- El texto en el lenguaje fuente se transforma a una representación intermedia.
- De ser necesario, se hacen algunos cambios a esta representación.
- Finalmente, la representación intermedia se transforma al texto en el lenguaje final.

En algunos sistemas, al texto generado con este esquema también se le aplican algunos ajustes previstos por las heurísticas de traducción.

De la calidad esperada de traducción y de la proximidad de los dos lenguajes depende qué tan profundas sean las transformaciones entre las dos representaciones, es decir, que tan diferente es la representación intermedia del texto en el lenguaje humano.

En algunos casos —se pueden mencionar traductores entre lenguajes tan parecidos como español y catalán (Canals-Marote, R. *et al*., 2001; Garrido-Alenda y Forcada, 2001), portugués y gallego, etc.— es suficiente con basarse en la representación morfológica: el análisis y generación de las palabras fuera del contexto. Por ejemplo, la palabra española *hijas* se analiza como *HIJO-femenino-plural*, se transforma (usando una tabla de correspondencias) a la representación *FILHO-femenino-plural*, de la cual se genera la palabra portuguesa *filhas* (aquí, *HIJA* y *FILHA* son claves de acceso a la base de datos que contiene las propiedades de las palabras en los lenguajes correspondientes).

En otros casos, cuando hay diferencias estructurales más profundas entre los lenguajes (que es el caso de casi cualquier pareja de idiomas, incluidos español—inglés, español—francés, etc.), se usan como representación intermedia (que en este caso se denomina interlingua) las estructuras formales de predicados lógicos o sus equivalentes, por ejemplo, redes semánticas (véase más abajo). Esta representación independiente del lenguaje es, en realidad, el sentido del mensaje que comunica el texto. Es decir, la transformación del texto a esta representación formal es comprensión del texto, y la transformación en sentido contrario es generación: teniendo una idea, decirla en lenguaje humano.

De esta discusión queda completamente claro que, en el caso de la traducción automática (de tipo interlingua), es mucho más importante aún que el programa comprenda el texto perfectamente y, por otro lado, que pueda verbalizar correctamente el sentido dado. Cualquier error de compresión del texto fuente causaría la traducción incorrecta, lo que, dependiendo de la situación del uso del texto traducido, podría resultar en consecuencias graves.

Entonces, este tipo de sistemas de traducción requieren de toda la fuerza de la lingüística computacional, de los métodos más precisos y completos de análisis de texto y representación de su contenido.

De forma adicional a los problemas de análisis de texto, la traducción automática enfrenta problemas específicos para la generación de texto. Uno de estos problemas es la selección de palabras. Por ejemplo, para traducir del inglés la frase *John pays attention to Mary*, no basta con encontrar en el diccionario el verbo *pay* 'pagar', ya que esta variante de traducción es incorrecta: **Juan paga atención a María*. La manera correcta de representar la palabra *pay* es tratarla como una *función léxica*: esta palabra significa la ejecución de la acción de *atención* por el agente (*Juan*). En español, la palabra que indica la ejecución de *atención* es *prestar*. Nótese que la selección de la palabra depende de la acción: para

culpa es *echar* (*Juan echa culpa a María*), para *propiedad* es *mostrar*, etcétera.

Otro ejemplo de una función léxica, es el significado de *mucho* o *muy*: *té cargado*, *voz alta*, *viento fuerte*, *gran vergüenza*, *alto mar*, *perfecto idiota*, *amigo incondicional*, *correr rápido*, *saber al dedillo*. Las funciones tienen sus denominaciones, por ejemplo, la función que significa *muy* tiene la denominación **Magn**, es decir, **Magn** (*saber*) = *al dedillo*. Se usan las combinaciones de estas funciones para formar otras funciones compuestas o para expresar las transformaciones equivalentes (Bolshakov y Gelbukh, 1998).

Nótese también que estas funciones son, en muchos casos, distintas en diferentes lenguas (como en el ejemplo de *pay attention* / *prestar atención*), lo que justifica su trato indirecto y separado en cada lenguaje durante el proceso de traducción o generación.

2.7 Generación de texto

¿Cómo puede la computadora comunicarle al usuario sus opiniones o pedirle información?

El complemento natural de la capacidad de entender el lenguaje es el segundo componente de la comunicación, la capacidad de producir el texto, o bien, el habla. En cierto grado es una tarea más simple que la comprensión, ya que por lo menos la computadora puede elegir las expresiones que sabe producir.

Uno podría pensar que para la generación de texto es suficiente con sólo saber las reglas de gramática, es decir, saber con que números, tiempos y géneros hay que usar las palabras en la oración y en que orden ponerlas. Sin embargo, hay algunos problemas más complicados en la generación de texto. Uno de ellos estriba en la necesidad de elegir las palabras y expresiones que «se usan» en un contexto dado.

El otro problema es que el texto producido con los métodos de fuerza bruta es aburrido, incoherente y a veces poco inteligible.

Hay que saber en qué ocasiones se deben usar pronombres y en qué otras las palabras completas, en qué ocasiones hay que explicar de qué se trata la oración y en qué otras es entendible para el lector. Esto se refiere a los métodos de la denominada *planificación textual*.

2.8 Aplicaciones recientes y emergentes

Ya que en este libro no tenemos espacio suficiente para describir todas las aplicaciones interesantes de las técnicas de procesamiento de lenguaje natural, sólo podemos mencionar aquí las que llamaron más la atención o recibieron mayor desarrollo en los últimos años.

Bibliotecas digitales

Como ya hemos discutido más arriba, el tesoro más valioso de la raza humana —su conocimiento y su cultura— se concentra en grandes acervos de textos (libros, revistas, periódicos) escritos en lenguaje natural. Tradicionalmente, tales acervos se llaman bibliotecas y han jugado un papel único en la difusión y conservación de la cultura y conocimiento.

Sin embargo, hasta ahora, la tecnología de mantenimiento de las bibliotecas era muy rudimentaria: se trataba de almacenes de libros con un soporte muy básico para encontrar un ejemplar si ya se conocían el autor y título. El «rendimiento» de tal difusión de conocimiento era muy bajo, incluso se puede decir que la mayor parte de la información contenida en los libros no era encontrada por quien la necesitaba ni en el momento en que se necesitaba.

Con el tratamiento digital de información la utilidad de las bibliotecas —que en este caso se llaman bibliotecas digitales— aumenta hasta convertirlas en servicios integrados y complejos de información cultural, científica y técnica. Obviamente, las facilidades de búsqueda inteligente proporcionadas por las tecnologías de lenguaje natural son sólo una parte de la solución integral, la cual involucra también aspectos técnicos, administrativos, legales y culturales.

Extracción de información, filtrado y alerta

Otra posibilidad que surgió con la aparición de grandes volúmenes de textos, que además crecen constantemente, es la creación de bases de datos específicos acerca de la información que se comunica en los textos. Por ejemplo, es posible crear una base de datos que guarde las atracciones turísticas por lugares, fechas y servicios, extrayendo esta información automáticamente de las descripciones en las páginas Web y la propaganda de las compañías turísticas. O bien, una base de datos de oferta y demanda de soluciones tecnológicas, que podría ser útil para una compañía de consultoría. Obviamente, este tipo de tareas requiere de cierto grado de comprensión del texto por parte de la máquina, aunque en un dominio acotado.

Otra tarea similar es el filtrado de información nueva, por ejemplo, de las noticias publicadas por las agencias. De muchos miles de noticias, el agente de filtrado selecciona sólo las que corresponden al perfil de intereses del usuario específico y las presenta en su escritorio. Si las noticias de este tipo aparecen muy raramente, la tarea se llama servicio de alerta: el agente advierte al usuario si aparece algo de su interés (digamos, la compañía cliente cambia de presidente).

Generación de resúmenes

Otro modo de filtrar la información relevante del mar de la irrelevante es la presentación resumida a través de generación automática de resúmenes de textos o colecciones de textos. Se trata de analizar un texto grande (o una colección grande de textos) y generar un informe corto de todo lo relevante que dicen estos textos. Así se da al lector una idea de su contenido sin la necesidad de que él tenga que leerlos completos.

Existen diferentes variantes de la tarea de generación de resúmenes. Por ejemplo, se puede buscar la opinión prevaleciente (más común) sobre el tema dado. Digamos, hay muchos artículos sobre el procesamiento de lenguaje natural, pero ¿cuáles son los problemas que más se discuten? ¿cuáles son las soluciones que más frecuentemente se proponen?

Una variante de la generación de resúmenes es la generación de resúmenes temáticos del texto: presenta un breve informe sobre los temas (aunque no las ideas) que se discuten en un texto dado (Gelbukh *et al*., 1999b). Por ejemplo: un texto habla sobre guerra, política y narcotráfico; otro texto habla sobre ciencia, tecnología y transporte. A pesar de la menor riqueza de esta presentación —en comparación con los resúmenes completos—, tiene algunas ventajas: es más simple de obtener y como consecuencia da resultados más seguros y estables; además, permita realizar operaciones matemáticas con los conjuntos (vectores) de temas obtenidos (Gelbukh *et al*., 1999c).

Minería de texto

La minería de texto consiste en descubrir, a partir de cantidades de texto grandes, el conocimiento que no está literalmente

escrito en cualquiera de los documentos. Esto incluye buscar tendencias, promedios, desviaciones, dependencias, etc. Es un área emergente, y muy interesante, del procesamiento de textos y minería de datos.

Por ejemplo, con los métodos de minería de texto, a partir de los textos de periódicos mexicanos encontrados en Internet, se podrían investigar preguntas como las siguientes: ¿Es positiva o negativa la opinión promedio en la sociedad sobre el tema del Fobaproa?, ¿aumenta o disminuye el interés en este asunto en los últimos meses? ¿Hay diferencias en la actitud hacia este asunto en el Distrito Federal y en Monterrey? ¿Cómo afecta la noticia de privatización de la industria eléctrica el interés social hacia el Fobaproa? (Montes y Gómez *et al*., 2001b).

En grandes cantidades de texto se puede no sólo encontrar lo que está escrito explícitamente en alguno de los textos, sino también descubrir cosas nuevas ¡de las cuales todavía nadie se dio cuenta! Digamos, detectar tendencias, relaciones y anomalías (Montes y Gómez *et al*., 2001c). Por ejemplo, descubrir que en el estado X la popularidad del gobierno empieza a caer (y al darse cuenta de esto, tomar medidas adecuadas a tiempo) —una tendencia. O bien, que en los estados donde las gobernadoras son mujeres hay más satisfacción de la población con el gobierno —una relación. O que el periódico X no publicó los informes sobre un evento que la mayoría de los periódicos discutió extensivamente —una anomalía.

Nótese que (a diferencia de las tareas de búsqueda, filtrado o extracción) esta información no está escrita explícitamente en algún texto, sino que se descubre con los métodos estadísticos. Véase también que la minería de texto es un modo distinto de la presentación resumida de información, aunque no de información explícita, sino implícita en los textos.

Manejo inteligente de documentos oficiales (e-Gobierno)

Las sociedades democráticas tienden a ser también burocráticas. Esto se debe, primero, al gran número de documentos que circulan, ya que cada ciudadano hace efectivos sus derechos de petición, apelación, opinión, etc., y, segundo, al gran número de personas involucradas en la consideración de tales documentos, de tal manera que el poder de decisión no se concentra en las manos de una o pocas personas. Esta situación causa retrasos y desorden cuando el flujo de documentos rebasa las capacidades del sistema burocrático.

Una solución eficiente a este problema, que permite ajustar la democracia con la eficacia, es el procesamiento automático de documentos, por lo menos en lo que se refiere a clasificación y distribución del flujo de documentos (Alexandrov *et al*., 2000b; Alexandrov *et al*., 2001; Makagonov *et al*., 2000), búsqueda de documentos relevantes y parecidos, etc. Por ejemplo, un sistema automático puede girar los documentos a los funcionarios o departamentos correspondientes. Puede agrupar los documentos que describen los casos parecidos para su consideración conjunta en una sola reunión. Puede facilitar al funcionario la búsqueda de los casos parecidos en el pasado, con el dictamen correspondiente, para que quede a su consideración si un dictamen similar podría aplicarse al caso en cuestión.

En México, como en algunos otros países, existen programas gubernamentales para el desarrollo de la infraestructura electrónica del manejo de documentos.[4]

Estudio de Internet como un corpus enorme

Todos los métodos de análisis de grandes cantidades de texto son específicamente útiles para la colección de información de Internet, que es fácil de obtener y es muy rica en contenido.

[4] www.e-mexico.gob.mx

También, en los años recientes, Internet comenzó a emplearse para construir sistemas de análisis de texto. Estos sistemas requieren de diccionarios muy grandes que indiquen las propiedades del lenguaje —tanto de las palabras como de las estructuras de oraciones y de texto completo. Usualmente, esta información se guarda junto con las estadísticas de uso: el hecho de que algunas estructuras se usen más frecuentemente que otras ayuda a entender el texto correctamente en los casos de ambigüedad. Obviamente, esta cantidad gigantesca de información no se puede compilar y codificar a mano. Por eso se aplican las técnicas de aprendizaje automático, para extraerla de grandes colecciones de textos llamadas corpus. Un corpus usualmente contiene marcaje especial o se prepara con las técnicas especiales para facilitar la extracción de la información necesaria.

Internet es la colección más grande de textos que ha creado la humanidad, y es una fuente muy rica de información no sólo sobre los hechos que se discuten allí, sino también sobre el propio lenguaje (aunque, por el momento, esto se aplica más al inglés que a otras lenguas). Sin embargo, Internet es un corpus muy especial, porque no cuenta con el marcaje y la estructura que usualmente ofrecen otro tipo de corpus, lo que resulta en el desarrollo de métodos especiales para su análisis (Gelbukh *et al*., 2002b).

Aplicaciones multilingües

Adicionalmente a las tareas de la traducción automática que ya hemos discutido, existe, y ha recibido recientemente un desarrollo considerable, un espectro de aplicaciones que involucran textos en diferentes lenguas sin traducirlos. La importancia de las aplicaciones multilingües aumentó mucho por las siguientes circunstancias:

- La formación de la Unión Europea. Las oficinas europeas manejan documentos en 12 lenguas oficiales de la Unión, y este número va a crecer más con la expansión de la unión a otros países europeos (tales como República Checa, Estonia, etc.). Obviamente, ningún empleado de estas oficinas maneja a la perfección tal cantidad de idiomas.
- El crecimiento de la democracia en los países multilingües. En estos países se fortalece la posición de las lenguas no oficiales pero muy usadas —como es el caso del español en los Estados Unidos.
- El desarrollo técnico de los países del tercer mundo. La revolución informática empieza a llegar a estos países donde, en muchos casos, hay decenas de lenguas usadas e incluso varias oficiales.

En situaciones como estas, muchos acervos de información son multilingües, por ejemplo: las bases de documentos oficiales de la Unión Europea contienen textos en muchas lenguas. De ahí que aparezcan tareas tales como la búsqueda *cross-lingual*: la pregunta se formula en el lenguaje que el usuario sabe mejor, pero se ejecuta sobre una colección de documentos en diferentes lenguajes. Todas las demás tareas de procesamiento de documentos —generación de resúmenes, minería, agrupamiento, etc.— también se pueden aplicar, con los métodos adecuados, a las colecciones de documentos multilingües.

Tecnologías de voz

El modo más natural de comunicación para un ser humano es hablar y escuchar, no escribir y leer. Tenemos que escribir y leer porque de esa forma realizamos las tareas principales de procesamiento de información: búsqueda y comparación. La voz

representa más información que el texto escrito: con entonaciones de la voz expresamos énfasis, propósitos, relaciones lógicas que se pierden en el texto.

Con la expansión de las áreas de aplicación de las computadoras a los servicios públicos, en los recientes años aumentó el desarrollo, y la investigación correspondiente, en las interfaces que utilizan la voz en lugar del teclado. Adicionalmente, se espera que con el uso creciente de las computadoras *palmtop* (pequeñas computadoras que caben en un bolsillo) la voz va a convertirse prácticamente en el único modo de comunicación con ellas, porque carecen de espacio para el teclado.

Las técnicas de reconocimiento de voz involucran áreas de dos ciencias diferentes: por un lado, los aspectos fisiológicos, físicos y acústicos de producción y procesamiento de la señal percibida y por el otro, los aspectos lingüísticos del contenido del mensaje comunicado. Actualmente, los sistemas prácticos carecen, en muchos casos, de la interacción apropiada de estas dos tecnologías, y prestan más atención a los métodos estadísticos de procesamiento de la señal acústica. Se espera de los sistemas futuros la interacción con la tecnología de procesamiento de lenguaje (las reglas gramaticales de la lengua y el razonamiento lógico sobre el contenido del mensaje) para la resolución de ambigüedades.

Conducción de diálogo

Bueno, si la computadora aprende a entender y producir el texto, ¿ya puede conversar con las personas?

El problema es que en las situaciones de conversación no hablamos con los textos, es decir, con los párrafos, capítulos y documentos. Hablamos con réplicas cortas, y la mayoría de la información omitida es clara en el contexto previo, en la situación,

en las acciones de los participantes y en el conocimiento general sobre el tipo de situación. Un diálogo en una cafetería podría ser: «¿*De manzana*?» — «*Piña*.» — «¡*Por favor*!» — «¿*Este*?» — «*El otro*.» — «*Dos pesos más*.» Claro que entender este tipo de conversación y participar en ella es una tarea muy diferente, y por supuesto más difícil, que entender un artículo con introducción, definición de los términos y un flujo lógico de ideas.

2.9 Problemas y métodos de análisis y representación de texto

Luego de haber dado suficientes argumentos para mostrar la importancia de la tarea del procesamiento automático de lenguaje natural, expongamos, en breve, el por qué no es trivial y cómo se hace.

Problemas

En este libro se mencionó ya varias veces el problema de la resolución de ambigüedades. Algunos científicos opinan que este problema es el principal en el procesamiento del lenguaje, en todos sus aspectos —desde el aspecto acústico hasta el semántico.

La ambigüedad, como indica el propio término, aparece cuando una unidad de lenguaje —un sonido, una palabra, una oración, etc.— se puede interpretar en más de una manera. Por ejemplo, la palabra *fuera* se puede interpretar como la forma morfológica de *ser* (en *como si fuera esta noche la última vez*), de *ir* (en *como si se fuera a la escuela*) o como adverbio (en *está fuera de la ciudad*). O bien, la oración *Veo al gato con el telelscopio* se puede interpretar como *uso el telelscopio para ver al gato* o *veo al gato que tiene el telescopio*.

Las ambigüedades se producen cuando una palabra, oración, etcétera se considera fuera de contexto —ya sea el contexto local explícitamente presente en el texto en cuestión o el contexto global de la experiencia de la vida humana o de los conocimientos escritos en otras fuentes. Es decir, para resolver las ambigüedades en cada nivel del lenguaje (al nivel de una palabra, una oración, etc.) se emplea el análisis de un nivel mayor, usando uno de los tres tipos de conocimiento o su combinación:

- Conocimiento lingüístico, o bien del lenguaje, aplicado al contexto cercano de la construcción en cuestión. Por ejemplo, no se pueden combinar en una oración dos verbos, de la manera como en *está fuera de la ciudad*: la palabra *está* indica que *fuera* no es un verbo, entonces, es adverbio.
- Conocimiento extralingüístico, o bien del mundo, aplicado al sentido del texto. Por ejemplo, en *veo al gato con el telescopio* se toma en cuenta que los gatos normalmente no usan telescopios.
- Conocimiento obtenido del mismo texto. Digamos, si el texto narra una historia de un gato mágico que sí sabe usar telescopios, la interpretación de la oración arriba mencionada cambia.

Leyendo un texto, las personas aplicamos todo este conocimiento fácilmente y sin darnos cuenta porque: 1) lo tenemos y 2) disponemos de mecanismos innatos muy eficientes para aplicarlo. Sin embargo, en la actualidad, los programas no disponen de tal conocimiento ni de mecanismos eficientes para su uso. Es labor nuestra proporcionárselos.

En el desarrollo de los algoritmos se usan métodos matemáticos y de la ciencia de la computación. En cuanto al conocimiento propio, el problema es quizá mucho más complejo, debido a la enorme cantidad de conocimiento requerido. Para obtener el co-

nocimiento lingüístico, adicionalmente a la codificación manual, se usan recientemente métodos de aprendizaje automático aplicados a los grandes corpus de textos. Para el extralingüístico están por desarrollarse las tecnologías de su construcción, diferentes a la mera codificación manual, que en este caso no es factible debido a la inconmensurable labor que representaría.

Conocimiento lingüístico vs. extralingüístico

Los diccionarios de propiedades gramaticales de las palabras no son el único conocimiento necesario para entender el lenguaje. También hace falta el conocimiento de las propiedades de las cosas y de las situaciones en el mundo real. El problema es que el texto no comunica toda la información necesaria para entenderlo, omite muchas ideas obvias que se pueden restaurar sencillamente por el humano que escucha... pero no por la computadora.

Consideremos una analogía. Cuando explicamos a alguien cómo ir al metro, le decimos algo como esto: «*Del Ángel vas por Reforma dos paradas en la dirección opuesta a la Diana, bajas en el Caballito y das vuelta a la derecha*». Tenemos en nuestra mente un mapa, y estamos seguros que quien oye también tiene en su mente un mapa igual al nuestro; lo único que necesitamos es darle unas pistas sobre su trayectoria en este mapa, unos puntos clave. Ahora bien, ¿qué sucederá si el que oye es un extranjero que no sabe ni qué son el Ángel o la Diana, ni como llegar allá, ni siquiera cómo usar los peseros, cómo son, dónde subir, ni cuánto hay que pagar? Ésta es la situación en la que se encuentran las computadoras: son extranjeras en nuestro mundo, no saben cómo usarlo, cómo se comportan las cosas en él. Las personas lo aprenden observando el mundo y participando en las situaciones. Las computadoras no tienen esta oportunidad.

Entones, nuestro texto es una línea de puntos, no es una trayectoria completa. Representa el modo humano de hablar. No lo podemos cambiar. Siempre suponemos que hablamos con alguien que sabe del mundo lo mismo que nosotros sabemos.

El problema es comunicarles a las computadoras el conocimiento humano del mundo real, por supuesto en la forma de diccionarios de relaciones entre objetos y de escenarios de las situaciones típicas. Estos diccionarios serán mucho más grandes que lo que hablamos en la sección anterior, y su compilación constituye una tarea a largo plazo, aunque ya existen algunos prototipos de diccionarios este tipo. Como una alternativa posible, se pueden desarrollar métodos de aprendizaje semiautomático de esta enorme cantidad de conocimiento, a partir de colecciones muy grandes de textos.

2.10 Métodos

Como ya se ha mencionado, el procesamiento de información en forma textual se conforma por dos tareas bastante independientes:

- El procesamiento de información propia según la aplicación específica: el razonamiento lógico, la búsqueda en la base de datos, etc. Dentro de un sistema, se realiza como un módulo especializado. Este módulo se toma como entrada, y genera como salida la información en una representación formal: predicados, tablas, números, etcétera.
- La traducción entre el texto en el lenguaje humano (una secuencia de letras o sonidos) y esta representación formal: la transformación del texto en la representación formal que sirve como entrada al módulo especializado y, de ser necesario, la transformación de su respuesta (representación formal) en texto.

La lingüística computacional sólo se ocupa de la última tarea: traducción entre el texto y la representación formal.

El punto crítico en el desarrollo de ese módulo consiste en seleccionar una representación formal lo suficientemente rica para reflejar el contenido del texto y, al mismo tiempo, lo suficientemente simple para no presentar problemas de comprensión. Esta representación, de manera particular, no debe presentar problemas de ambigüedad y, en la mayoría de casos, debe ser independiente del lenguaje humano específico (español, inglés, etcétera).

Entre las representaciones más prometedoras están las llamadas *redes semánticas* (Bolshakov y Gelbukh, 2004) —redes que representan las situaciones (acciones) y sus participantes según lo descrito en el texto. Un nodo en tal red es una situación o una entidad, y un enlace es el hecho de que la entidad (a veces, incluso, otra situación) participa en una situación. Varias entidades pueden participar en la misma situación y una entidad puede participar en varias situaciones. Esto cubre prácticamente cualquier aspecto del sentido comunicado en el texto. Existen también varias representaciones equivalentes, en lo esencial, a las redes semánticas, como los grafos conceptuales (Montes y Gómez *et al*., 2001a).

La tarea de transformación entre el texto y la red semántica es muy compleja, ya que estas dos representaciones de información son muy diferentes. Afortunadamente, se puede efectuar en varios pasos (etapas) según los niveles o capas de lenguaje (véase capítulo 3).

2.11 Procesamiento de lenguaje natural en México

Dada la importancia del procesamiento de lenguaje natural para la ciencia y educación (búsqueda), informatización de la

sociedad (interfaces en lenguaje natural), comercio (búsqueda y minería de texto), cultura (búsqueda y traducción) y otros aspectos de la vida social, el desarrollo de las herramientas para el procesamiento de lenguaje natural a nivel nacional es uno de los aspectos críticos de la independencia cultural, técnica y económica del país. Además, el procesamiento rápido y correcto (lo que implica procesamiento automático) de los documentos en las oficinas de gobierno es indispensable para el funcionamiento eficiente de la democracia. Estas consideraciones, entre otras, nos indican la pertinencia y prioridad del desarrollo del procesamiento automático de español en nuestro país.

Las investigaciones en torno a la aplicación de las computadoras en las tareas lingüísticas se empezaron en México en los setenta, cuando Luís Fernando Lara, de El Colegio de México[5], aplicó las técnicas estadísticas al análisis automático de un corpus de español mexicano, con el fin de desarrollar un diccionario de las palabras usadas en México. También en El Colegio de México, Raúl Ávila hace investigaciones sobre problemas de lexicografía computacional.

En 1996, en el CIC-IPN se fundó el Laboratorio de Lenguaje Natural y Procesamiento de Texto[6]. A lo largo de sus 14 años de historia, el Laboratorio ha desarrollado varios proyectos científicos y tecnológicos y ha efectuado más de 600 publicaciones —la mayoría internacionales— en las áreas de análisis sintáctico y semántico, aprendizaje automático de los recursos léxicos y compilación de diccionarios, minería de texto y resolución de anáfora. El Laboratorio ofrece estudios de Maestría y Doctorado apoyados con becas para proyectos de investigación. Actualmente, aparte de los autores de este libro, otros miembros del Laboratorio son I. Bolshakov y H. Calvo.

[5] www.colmex.mx

[6] www.gelbukh.com/lab.htm; algunas de las publicaciones del Laboratorio están en www.gelbukh.com y www.cic.ipn/ ~sidorov

Aproximadamente en 1998, en el Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS), de la Universidad Nacional Autónoma de México (UNAM), se formó un grupo de investigación en lingüística computacional liderado por Luís Pineda Cortés[7]. El grupo hace investigación fuerte en diálogos multimodales (los que emplean, adicionalmente al lenguaje, también gestos, imágenes, etc.), así como en formalismos gramaticales modernos, como HPSG. También ofrece estudios de Maestría y Doctorado. En la misma UNAM, otro grupo liderado por Gerardo Sierra trabaja en ingeniería lingüística —es decir, resolviendo los problemas lingüísticos desde el punto de vista más práctico.

En 2002, un grupo muy prometedor —el Laboratorio de Tecnologías del Lenguaje— se formó en el Instituto Nacional de Astrofísica Óptica y Electrónica (INAOE), en Puebla, orientado a los proyectos prácticos. Entre sus intereses están la recuperación de información, la minería de texto y el procesamiento de voz (habla). El grupo está formado por Luís Villaseñor Pineda, Aurelio López López y Thamar Solorio, entre otros. De este grupo también forma parte Manuel Montes[8] —uno de los primeros doctores en lingüística computacional formados en México, egresado del Laboratorio de Lenguaje Natural del CIC-IPN. El grupo también ofrece estudios de Maestría y doctorado.

Además, varios investigadores trabajan en diferentes instituciones en temas relacionados al Procesamiento de Lenguaje Natural —por ejemplo, Sofía Galicia Haro en la Facultad de Ciencias de la UNAM, Héctor Jiménez Salazar y David Pinto en la Benemérita Universidad Autónoma de Puebla; Everardo García Menier en la Universidad de Jalapa, Veracruz, entre otros.

Para coordinar los esfuerzos de los grupos mexicanos que trabajan en el Procesamiento de Lenguaje Natural, se está forman-

[7] leibniz.iimas.unam.mx/~luis
[8] cseg.inaoep.mx/~mmontesg

do la Asociación Mexicana para el Procesamiento de Lenguaje Natural (AMPLN)[9].

Finalmente, como una parte de la infraestructura de la investigación en lingüística computacional en México, el Laboratorio de Lenguaje Natural del CIC-IPN anualmente (en febrero de cada año) organiza el CICLing, Congreso Internacional de Lingüística Computacional y Procesamiento Inteligente de Texto[10]. El congreso reúne a los más conocidos especialistas en esta rama de la ciencia, para dar a los estudiantes mexicanos la oportunidad de escuchar a los más renombrados expertos del mundo. Las memorias del Congreso se publican por la casa editorial más prestigiosa de Alemania, Springer, en la serie *Lecture Notes in Computer Science* (hasta ahora han aparecido diez volúmenes, números 2004, 2276, 2588, 2945, 3406, 3878, 4394, 4919, 5449, 6008).

Varios investigadores mexicanos que trabajan en lingüística computacional son miembros del Sistema Nacional de Investigadores. La ciencia de la lingüística computacional también está representada en la Academia Mexicana de Ciencias por los autores de este libro.

2.12 Conclusiones

El procesamiento de lenguaje natural es importante para muchos aspectos de la vida de la sociedad, desde la informatización de los servicios públicos y el desarrollo de la democracia, hasta la ciencia, educación y cultura. El desarrollo de las herramientas correspondientes para la lengua nacional es indispensable para la dirección cultural del país.

---

[9] www.AMPLN.org
[10] www.CICLing.org

Entre las tareas principales del procesamiento de lenguaje natural se pueden mencionar:

- Manejo eficiente de la información (búsqueda, clasificación, agrupamiento, resúmenes, filtrado y alerta; bibliotecas digitales).
- Interfaces en lenguaje natural y tecnologías de voz, tanto en los equipos especializados como en los servicios públicos.
- Traducción automática y aplicaciones multilingües.
- Ingeniería de conocimiento: extracción de información, minería de texto.

En el aspecto técnico, como hemos visto, el procesamiento de lenguaje natural enfrenta la gran complejidad que implica el conocimiento involucrado. La compilación de este conocimiento es uno de los grandes retos para la ingeniería en sistemas lingüísticos; y una de las soluciones a este problema es el aprendizaje automático del conocimiento a partir de los grandes corpus de textos.

Otra solución al problema de complejidad es la partición del procesamiento en pasos (fases) que corresponden a los niveles (capas) del lenguaje: análisis morfológico (palabras), sintáctico (oraciones) y semántico (texto completo). Esta solución da origen a otro problema: la ambigüedad. Las ambigüedades que se presentan en un nivel (por ejemplo, *aviso*: ¿sustantivo o verbo?) se resuelven en otro nivel de análisis. La ambigüedad es probablemente el problema más importante en el análisis del lenguaje natural.

En México existen varios grupos que trabajan activamente en las tecnologías de lenguaje, tanto en los aspectos prácticos como teóricos. Las instituciones en que se encuentran estos grupos ofrecen estudios de Maestría y Doctorado, así como oportunidades

de colaboración en los proyectos aplicados. Para la difusión de los resultados y de los problemas actuales de la investigación, anualmente se organiza en México un congreso internacional de alto nivel.

Capítulo 3

NIVELES DE LENGUAJE Y SU REFLEJO EN PLN

Las computadoras juegan un papel imprescindible en la realidad actual; los cambios que ellas trajeron dieron la vuelta al mundo. Casi todas las áreas relacionadas con los seres humanos se modificaron desde entonces. Este cambio es notable especialmente en el área de la ciencia y la tecnología, pero lo es también en la sociedad, que tiene ahora acceso libre a la información a través de la Internet, y puede comunicarse rápidamente mediante el correo electrónico.

Las computadoras son como siervos fieles, pues ejecutan todo lo que les ordenamos. Pero son todavía siervos sordos y mudos (y, a veces, tontos). Es difícil hacerles entender nuestras órdenes, hay que saber comunicarse con ellas en su idioma: algún lenguaje de programación. Las excepciones a esta regla son, quizá, algunos programas comunes prediseñados, como por ejemplo hojas de cálculo, programas de correo electrónico o procesadores de texto, entre otros.

Por otro lado, los humanos poseemos una forma muy eficaz de comunicarnos: usamos el lenguaje natural. Incluso, para nosotros no es solamente la herramienta de comunicación, es un modo de pensar. ¿Pueden imaginar alguna actividad intelectual que no involucre palabras y frases?

Otro punto importante del lenguaje natural es su habilidad de representar el conocimiento. La mayor parte del conocimiento tiene una forma simbólica, es decir, la forma de texto escrito. Así que, si las computadoras entendieran el lenguaje natural, serían unos ayudantes mucho más valiosos y efectivos. ¿Podrían llegar a dominar el mundo? Todavía falta tanto, que podemos dormir tranquilamente...

Entonces, ¿cómo hacer que las computadoras entiendan lo que nosotros dominamos con una gran facilidad: la lengua natural? Vamos a analizar este problema desde el punto de vista de la ciencia del lenguaje humano: la lingüística. Aprendemos el lenguaje a los dos o tres años de edad, y toda la vida seguimos usándolo sin problema alguno. Incluso podríamos considerarnos a nosotros mismos unos especialistas en lingüística, pues usamos el lenguaje muy fácilmente, a diferencia, por ejemplo, de las matemáticas, que pueden resultarnos complejas. Usar el lenguaje es tan natural que, a veces, no nos damos cuenta de la complejidad de su sistema. Muchas veces, entendemos esta complejidad cuando tratamos de aprender algún otro idioma, lo que a partir de los diez años de edad ya cuesta muchísimo trabajo y difícilmente se alcanza a la perfección. Aquí cabe recordar una vieja broma:

> Un jeque árabe quiso aprender el francés e invitó a un maestro de Francia. Estudió el francés algún tiempo, después visitó París, regresó y dio órdenes de castigar severamente al maestro. «Pero ¿por qué?» —le preguntaron. «En París —contestó— ¡cualquier niño de la calle habla el francés mejor que yo!».

El jeque supuso que todos hablan su idioma —el árabe— porque para él ha sido algo natural el hablarlo; pero no reparó en que el aprendizaje de otros idiomas supone una dificultad superior y especial.

3.1 Modelos buenos y modelos malos

Entonces, ¿de qué se trata esta ciencia, la lingüística, la cual aplicamos tan fácilmente en el uso de nuestra propia lengua natal?

Toda ciencia se sustenta en el desarrollo de modelos. Un modelo es una construcción mental que refleja algunas características del objeto de investigación que son relevantes para una determinada investigación. Por ejemplo, los físicos desarrollan modelos de la naturaleza, o, en un caso más específico, del átomo, etcétera; los biólogos hablan de los seres vivientes, o de fotosíntesis, entre otros temas. El modelo depende totalmente del objeto que estamos modelando. Claro que un modelo ideal refleja todas las características del objeto, pero es casi imposible construir ese modelo para un objeto del mundo real. Por eso, para una investigación, tenemos que conformarnos con enfocarnos sólo en las características importantes.

De esta manera, la lingüística construye modelos del lenguaje o de diferentes fenómenos relacionados con el lenguaje. Desde ahora podemos precisar que la lingüística computacional trabaja con modelos que deben ser entendibles para las computadoras, esto es, modelos que tienen un grado adicional de precisión y formalización.

Los modelos, por supuesto, pueden ser de buena calidad o de mala calidad. ¿Cuál es un modelo bueno y cuál es un modelo malo? Recordemos que cualquier modelo, por definición, refleja algunas características del objeto. Sin embargo, un modelo bueno debe explicar, predecir, ser elegante y simple; no introducir conceptos innecesarios —principio conocido como navaja de Occam. Seguramente, todos estamos de acuerdo con esto, pero en los casos prácticos no es fácil seguir el camino correcto. Vamos a presentar un ejemplo muy simple relacionado con la lingüística.

¿Cuál sería el modelo que describe al acento gráfico en español? En las gramáticas se enseña que existen varios tipos de

palabras, clasificadas según el lugar donde se encuentra el acento fonético: agudas (la última sílaba), graves (la penúltima sílaba), esdrújulas (la antepenúltima sílaba) y sobreesdrújulas (antes de la antepenúltima sílaba). Acento fonético en español es un modo de pronunciar más fuerte una de las vocales (*a, e, o, u, i, y*) de la palabra, en relación con las otras. Según el tipo de palabra, se coloca el acento gráfico usando las tres reglas básicas siguientes (Álvarez, 1977):

1. Llevan acento gráfico las palabras agudas si terminan en *-n*, *-s*, o vocal.
2. No llevan el acento gráfico las palabras graves que terminan en *-n*, *-s*, o vocal; pero lo llevan todas las que terminan en las demás consonantes.
3. Todas las palabras esdrújulas y sobreesdrújulas llevan acento gráfico.

Hay otras reglas más específicas relacionadas, por ejemplo, con los diptongos específicos (por ejemplo, dos vocales contiguas donde por lo menos una vocal es cerrada —*u*, *i*, *y*; etc.); que no vamos a discutir aquí porque no afectan la parte principal de nuestra discusión en torno a la calidad de los modelos.

¿Es un modelo correcto? Sí, porque describe el fenómeno, pero ¿es un modelo bueno? ¿Explica el fenómeno? ¿Fundamenta su presencia en el español en comparación con otras lenguas? Desafortunadamente, tenemos que responder que no. Este modelo no presenta una explicación; cada palabra pertenece a una clase y sólo por eso tiene o no acento; es decir, el modelo no nos explica por qué existe el acento gráfico. Además, se agregan cuatro conceptos adicionales (agudas, graves, etcétera). ¿Qué tan necesario es tener esas clases de palabras? ¿No es una violación del principio de navaja de Occam? En total son cuatro reglas —pues, de hecho, la regla 2 contiene dos reglas—. Para nuestro gusto

esas reglas no parecen muy claras, porque mezclan los conceptos relacionados con el acento fonético (graves, agudas, etc.) con los conceptos relacionados directamente con el acento gráfico (como, por ejemplo, «terminan en *-n*, *-s*, o vocal»). Tal vez, por eso cuesta trabajo aprenderlas, a pesar de no ser tan complejas.

Ahora bien, trataremos de presentar un modelo mejor. ¿De que depende el acento gráfico? Claro que tiene cierta relación con el acento fonético; hay una razón por la que algunas palabras están resaltando el lugar de su acento fonético. Muy bien, ahora formulamos las dos reglas simples que debe cumplir el acento fonético en el español:

1. Si la palabra termina en *-n*, *-s*, o vocal, el acento fonético debe caer en la penúltima sílaba.
2. En caso contrario (si la palabra termina en alguna consonante que no es *-n* o *-s*), el acento fonético debe caer en la última sílaba.

La mayoría de las palabras en español cumplen con estas reglas, pero algunas palabras no tienen el acento fonético en el lugar indicado por ellas; y justamente es en esas palabras en las que se pone el acento gráfico. Entonces, ya podemos determinar en nuestro modelo la función del acento gráfico —marcar las excepciones de las dos simples reglas del acento fonético. Eso explica el porqué del acento gráfico. Nótese que ya no necesitamos los conceptos de palabras agudas, graves, etcétera. Además, en lugar de cuatro reglas, tenemos solamente dos.

Es claro que no inventamos las dos reglas, sólo presentamos las cuatro anteriores de otra manera y creamos un modelo que explica mucho mejor el fenómeno del acento gráfico en español.

Ahora bien, ¿por qué no existe el acento gráfico en inglés ni en ruso? Ya lo podemos explicar muy fácilmente: no existen reglas tan simples de acento fonético, el acento es muy cambiante

y puede estar en cualquier lugar de la palabra. De otra forma, tendríamos que acentuar casi todas las palabras, lo que no tiene mucho sentido. ¿Y en francés, donde el acento fonético siempre cae en la última silaba? El acento gráfico de tipo español sería redundante, ya que de antemano sabemos su posición.

¿Podría existir el español sin el acento gráfico? Lo podemos imaginar; no sucedería algo fatal. No obstante, hay consideraciones a favor del uso del acento gráfico:

- Son sólo dos reglas simples y la función del acento gráfico es muy clara: marcar excepciones.
- En caso de no usar el acento gráfico aparecería la homonimia en el sistema verbal en casos muy importantes y frecuentes como: *trabajo* – *trabajó* (primera persona en presente vs. tercera persona en pasado).

3.2 Niveles de lenguaje natural

¿Qué más es importante saber en lingüística para desarrollar modelos que sean aptos para las computadoras? Se puede tratar de desarrollar un modelo de lenguaje completo, sin embargo, es preferible dividir el objeto en partes y construir modelos más pequeños, y por ello más simples, de partes del lenguaje. Para eso se usa el concepto de *niveles del lenguaje*. Tradicionalmente, el lenguaje natural se divide en seis niveles:

1. fonética / fonología
2. morfología
3. sintaxis
4. semántica
5. pragmática
6. discurso

No existen criterios exactos para la separación de cada uno de los niveles; de hecho, las diferencias entre los niveles se basan en el enfoque de análisis de cada uno. Por eso pueden existir traslapes entre niveles sin presentar contradicción alguna. Por ejemplo, existen fenómenos relacionados tanto con fonología como con morfología, digamos, alternaciones de raíces (morfonología) *acordar – acuerdo*, *dirigir – dirijo*, entre otros casos.

A continuación vamos a describir brevemente cada nivel del lenguaje y sus avances computacionales.

Fonética / fonología

La fonética es la parte de la lingüística que se dedica a la exploración de las características del sonido —que es un elemento substancial del lenguaje. Eso determina que los métodos de fonética sean en su mayoría físicos; por eso su posición dentro de la lingüística es bastante independiente.

Los problemas en fonética computacional están relacionados con el desarrollo de sistemas de reconocimiento de voz y síntesis del habla. Aunque sí hay sistemas de reconocimiento de voz —esto es, la computadora puede reconocer las palabras pronunciadas en el micrófono—, el porcentaje de las palabras reconocidas correctamente es considerablemente bajo. Entre los sistemas de síntesis de habla hay mucho más éxito, existen sistemas que hablan bastante bien, incluso sin el acento *de robot*, pero aún no suenan completamente como un humano; se puede visitar el sitio de *Loquendo Vocal Technology and Services* [loquendo.com] para hacer pruebas con varios módulos de generación. Acerca de los sistemas de síntesis de habla hay que decir, además, que su área de aplicación es bastante restringida; normalmente es mucho más rápido, cómodo y seguro leer un mensaje que escucharlo;

los sistemas de síntesis de habla son útiles básicamente para las personas con deficiencias de la vista.

A la fonología también le interesan los sonidos pero desde otro punto de vista. Su interés se enfoca a la posición del sonido en relación con otros sonidos de algún idioma, es decir, las relaciones con los demás sonidos dentro del sistema y sus implicaciones. Por ejemplo, ¿por qué los japoneses no pueden distinguir entre los fonemas [l] y [r]? ¿Por qué los extranjeros hablan el español con un acento notable, digamos pronuncian [rr] en lugar de [r]? ¿Por qué los que hablan español usualmente tienen un acento hablando otros idiomas, como cuando no pueden pronunciar [l duro], como se pronuncia [l] en inglés? La respuesta es la misma en todos los casos: en los idiomas nativos no existen oposiciones entre los fonemas mencionados, y, por lo tanto, las diferencias que parecen muy notables en algunas lenguas, son insignificantes en otras; en japonés no existe el fonema [l], en la mayoría de los idiomas existe sólo un fonema para [r] – [rr] y, obviamente, no importa su duración (el español representa el caso contrario); por otra parte, en español no existe el fonema [l duro], sólo existe [l suave], por eso al hablar inglés, en donde el fonema [l] se pronuncia duro, hablantes de español lo pronuncian de manera suave, como en su idioma natal.

Morfología

El área de morfología es la estructura interna de las palabras (sufijos, prefijos, raíces, flexiones) y el sistema de categorías gramaticales de los idiomas (género, número, etc.). Hay lenguas que tienen muchas diferencias en relación con las reglas que tenemos en el español. Por ejemplo, en el árabe, la raíz contiene tres consonantes, y las diferentes formas gramaticales de la palabra se forman a partir de la inserción de vocales entre las consonantes

(*KiTaB* <*el libro*>, *KaTiB* <*leyendo*>, etc.). En el chino casi no existen las formas morfológicas de las palabras, lo que se compensa en el nivel de la sintaxis (orden de palabras fijo, palabras auxiliares, etc.). En los idiomas turcos los sufijos se pegan a la raíz expresando cada uno un solo valor de las categorías gramaticales, por ejemplo, en el azerbaijano una sola forma *baj-dyr-abil-dy-my* con los cuatro morfemas gramaticales significa *¿si él pudo obligar a ver?*, los morfemas expresan *posibilidad (poder), obligación, pasado* e *interrogación*; no se puede traducirla con una sola palabra en español, porque los morfemas que son gramaticales en el azerbaijano y se encuentran dentro de la palabra, corresponden a los verbos auxiliares y a las palabras auxiliares en el español; nótese que pueden existir palabras con más de diez morfemas.

Los problemas de morfología computacional están relacionados con el desarrollo de sistemas de análisis y síntesis morfológica automática. El desarrollo de tales módulos es aún bastante fatigoso, porque hay que hacer grandes diccionarios de raíces (que deben contener alrededor de cien mil elementos). En general, existe la metodología de ese desarrollo y existen sistemas funcionando para muchos idiomas. Lo que hace falta es un estándar de tales módulos. En el CIC hemos desarrollado un sistema de análisis morfológico para el español que está disponible a todo el que lo necesite (véase Capítulo 5).

Sintaxis

La sintaxis se dedica a analizar las relaciones entre las palabras dentro de la frase. Existen dos modelos principales para la representación de tales relaciones: 1) dependencias, donde las relaciones se marcan con flechas y una palabra puede tener varias que dependen de ella, y 2) constituyentes, donde las relaciones existen en forma de árbol binario.

La sintaxis computacional debe tener métodos para análisis y síntesis automática, es decir, construir la estructura de la frase, o generar la frase basándose en su estructura. El desarrollo de los generadores es una tarea más fácil, y es claro qué algoritmos se necesitan para estos sistemas. Por el contrario, el desarrollo de los analizadores sintácticos (también llamados *parsers*) todavía es un problema abierto, especialmente para los idiomas que no tienen un orden de palabras fijo, como el español. En el inglés el orden de las palabras es fijo, por eso las teorías basadas en el inglés no son tan fácilmente adaptables para el español. Vamos a presentar un ejemplo de *parser* en las siguientes secciones.

Semántica

El propósito de la semántica es «entender» la frase. ¿Pero qué significa «entender»? Hay que saber el sentido de todas las palabras e interpretar las relaciones sintácticas. Los investigadores están más o menos de acuerdo que los resultados del análisis semántico deben ser *redes semánticas*, donde se representan todos los conceptos y las relaciones entre ellos. Otra posible representación es algo muy parecido a las redes semánticas: los *grafos conceptuales*. Entonces, lo que se necesita saber es cómo hacer la transformación de un árbol sintáctico a una red semántica. Ese problema todavía no tiene una solución general.

Otra tarea de la semántica (o más bien, de sus subdisciplinas llamadas lexicología y lexicografía) es definir los sentidos de las palabras, lo que representa de por sí una tarea muy difícil, aun cuando se realiza manualmente. Los resultados de la definición de los sentidos de las palabras existen en forma de diccionarios. Aquí el problema principal es que siempre[11] existe un círculo

[11] Si no existe el círculo vicioso, entonces algunas palabras no están definidas.

vicioso en las definiciones de las palabras, porque las palabras se definen a través de otras palabras. Por ejemplo, si definimos *gallo* como «el macho de la gallina» y *gallina* como «la hembra del gallo», no ayudaremos a alguien que quiere averiguar qué cosas son. En este ejemplo, el círculo vicioso es muy corto, normalmente los círculos son más largos, pero son inevitables. La semántica computacional puede ayudar a resolverlo buscando un conjunto de palabras a través de las cuales se definirán todas las demás palabras: el *vocabulario definidor*. Otro problema específico es evaluar automáticamente la calidad de los diccionarios. Todos usamos los diccionarios y sabemos que hay buenos y malos.

Una aplicación importante del análisis semántico es la *desambiguación automática de sentidos de palabras*. Por ejemplo, *un gato* puede ser *un felino*, o *una herramienta*, o *una persona*. Para saber cuál de los sentidos se usa en un contexto dado se pueden aplicar diferentes métodos con el fin de analizar las demás palabras presentes en el contexto. Por ejemplo, en la frase *El gato se acostó en el sillón y estaba maullando*, las palabras *acostarse* y *maullar* indican que es *un felino*; mientras que en la frase *El mecánico usó un gato para subir el automóvil*, las palabras *mecánico, subir* y *automóvil* dan la preferencia al sentido *una herramienta.* Sin embargo, en la frase *El mecánico compró un gato y lo llevó en su carro*, no se puede definir el sentido. Ni siquiera nosotros mismos lo podemos hacer sin un contexto más amplio.

En suma, los problemas de semántica computacional son muy interesantes, pero todavía queda mucho por investigar.

Pragmática

Usualmente se dice que la pragmática trata de las relaciones entre la oración y el mundo externo. Un ejemplo famoso es el siguiente: usted y yo estamos comiendo juntos y yo le pregunto

a usted si puede pasarme la sal, usted contesta que sí... y sigue comiendo. Seguramente la respuesta es formalmente correcta, porque usted realmente puede pasarme la sal y eso es lo que contiene literalmente la pregunta, pero la intención fue pedir la sal y no preguntar sobre la posibilidad de pasarla. De otra manera, se puede decir que lo que interesa a la pragmática son las intenciones del autor del texto o del hablante.

Otro ejemplo del dominio de la pragmática es la clase de oraciones que tienen como característica particular ser acciones por sí mismas (se llaman performativas). Por ejemplo, decir *prometo* es precisamente la acción de *prometer*.

Como nos tropezamos con muchos problemas ya en el nivel semántico, normalmente es difícil continuar la cadena de análisis en el siguiente nivel –pragmático–, aunque siempre hay que tomarlo en cuenta.

Discurso

Normalmente no hablamos con una oración aislada, sino con varias oraciones. Esas oraciones tienen ciertas relaciones entre sí. Las oraciones hiladas forman una nueva entidad llamada *discurso*.

En el análisis del discurso existe un problema muy importante: la resolución de *correferencia*. Las relaciones de correferencia también se llaman anafóricas.

Por ejemplo, en el discurso «*He visto una nueva casa ayer. Su cocina era excepcionalmente grande*» (*su = de la casa*); o «*Llegó Juan. Él estaba cansado*» (*él = Juan*). Esas son relaciones de correferencia, y la computadora tiene que interpretarlas correctamente para poder construir las representaciones semánticas.

Existen algoritmos de resolución de correferencia bastante buenos, donde se alcanza hasta 90% de exactitud, sin embargo, resolver el 10% restante todavía es una tarea difícil.

3.3 Implementación de un procesador lingüístico

La estructura general del procesador lingüístico —el programa que hace el análisis de los textos— corresponde a los niveles del lenguaje; la excepción es el nivel fonético, porque el texto ya está representado con palabras escritas y no con sonidos.

Fase 1. Transformación morfológica entre las palabras. En este paso se resuelven las secuencias de letras en la llamada representación morfológica del texto: la secuencia de las estructuras de palabras en la forma del lema (que puede servir como una clave a una base de datos que guarda todas las propiedades de la palabra) y las propiedades específicas en el texto:

fuera ↔ *SER, subjuntivo, tercera persona, singular*

La representación morfológica del texto es, entonces, una tabla que da los lemas y las propiedades de cada palabra del texto, correspondiendo a una palabra un renglón en la tabla.

Para este paso, se usan los diccionarios morfológicos y los métodos matemáticos de autómatas de estados finitos.

Fase 2. Transformación sintáctica entre la representación morfológica y la representación sintáctica. Ésta última es una secuencia de estructuras de oraciones, siendo una estructura de oración un árbol sintáctico que representa qué palabras están relacionadas sintácticamente a cuáles otras en la misma oración.

Para este paso se usan los diccionarios sintácticos y los métodos matemáticos de gramáticas libres de contexto; los algoritmos que aplican tales gramáticas para el análisis del texto se llaman *parsers*.

Fase 3. Transformación semántica entre la representación sintáctica (la secuencia de árboles) y la representación semántica (la red semántica). En este paso se identifican las palabras que refieren a la misma entidad (o situación). Por ejemplo, en el texto:

Juan sacó 8 en el examen. Esto desanimó mucho al pobrecito se tiene que detectar que quien fue *desanimado* es *Juan*, lo que técnicamente consiste en mapear las dos frases —*Juan* y *el pobrecito*— al mismo nodo (entidad) de la red semántica; también se debe mapear *esto* y *sacó 8 en el examen* al mismo nodo (situación).

Los tipos más comunes de tal correferencia son la anáfora directa —expresada con pronombres— e indirecta —expresada con artículos (véase capítulo 6)—; ambos tipos se pueden ver en el ejemplo anterior.

Para la transformación semántica se usan los diccionarios semánticos, las reglas de transformación y los métodos de inferencia lógica.

De hecho, en cada paso la representación puede reflejar la ambigüedad:

f u e r a	↔	*SER, subjuntivo, tercera persona, singular* *IR, subjuntivo, tercera persona, singular* *FUERA, adverbio*

Como ya seguramente se dio cuenta el lector, es precisamente esta separación de todo el proceso en pasos, o fases —aunque simplifica muchísimo la tarea—, lo que da origen a los problemas de ambigüedad. Las ambigüedades en cada paso sólo se pueden resolver en alguno de los siguientes pasos.

En un paso posterior del análisis, sólo una variante se preservará y todas las demás se eliminarán. Eso permite subir, empezando el análisis en niveles más simples y construyendo las representaciones de un nivel dado con base en los niveles anteriores.

Hablando de la estructura del procesador lingüístico, asumimos que debe contener los siguientes módulos:

- El módulo *morfológico* reconoce las palabras y las convierte, de cadenas de letras, en referencias al diccionario y en marcas de tiempo, género y número, entre otras. Toma como entrada el

texto y pasa su salida —*representación morfológica*— al módulo siguiente.

- El módulo *sintáctico* reconoce las oraciones y las convierte, de cadenas de palabras marcadas, en estructuras de oraciones con marcas de sujeto, objeto y otras, y también reconoce las relaciones entre las palabras en la oración. Toma como entrada la representación morfológica y pasa su salida —*representación sintáctica*— al módulo siguiente.
- Los módulos *semántico*, *pragmático* y *discursivo* reconocen la estructura completa del texto y lo convierten en una *red semántica*. Resuelven las relaciones entre los pronombres y sus antecedentes, etcétera, y reconocen las intenciones del autor. Toman como entrada la representación sintáctica y generan la salida del procesador lingüístico.

En cada paso existen problemas técnicos y teóricos, algunos ya resueltos en cierto grado y algunos por resolverse en el transcurso del desarrollo de la lingüística computacional.

Es obvio que para realizar todo el proceso correctamente el sistema debe entender el texto, es decir, construir las redes semánticas. Esto es algo que los sistemas modernos no saben hacer todavía. Aunque en la época actual existen las aplicaciones mencionadas, normalmente se basan en heurísticas, en especial si se trata de los niveles semántico o pragmático; es decir, las aplicaciones funcionan y son útiles, pero todavía no alcanzan la calidad que se desea.

Capítulo 4

PROBLEMAS DEL USO DE DICCIONARIOS EN PLN

Los diccionarios explicativos son el corazón de la descripción lexicográfica de una lengua, la máxima autoridad que determina uso y comprensión correctos y precisos de sus palabras, que contiene el acervo de la sabiduría de todo un pueblo. Los diccionarios son elaborados con gran esmero por equipos de profesionales durante muchos años, para garantizar su impecable calidad.

Sin embargo —como se verá en los ejemplos que presentaremos— es muy difícil garantizar la calidad con los métodos tradicionales (Gelbukh y Sidorov, 2003b). Esta dificultad se debe a que un diccionario es un sistema complejo de elementos interrelacionados y al vivo uso del lenguaje. Como en el caso de cualquier sistema complejo, su calidad no se puede evaluar —ni mucho menos garantizar— observando y analizando sus elementos —los vocablos— aislados, uno por uno. La evaluación se debe llevar a cabo tomando en cuenta las relaciones entre los elementos que se encuentran en lugares muy diferentes del diccionario completo.

Por ejemplo, en el diccionario más popular del idioma ruso (Ozhegov, 1990), *gallina* (en ruso, *kúritsa*) se define como *la hembra del gallo* y, a cientos de hojas de esta definición, encontramos la

definición para *gallo* (en ruso, *petuj*) como *el macho de la gallina*. Aunque ambas definiciones son igualmente correctas y válidas, obviamente no son compatibles dentro del mismo sistema lógico, ya que a quien no sabe que son *kúritsa* y *petuj*, no le proporcionan información suficiente.

Para los humanos es muy difícil, por no decir imposible, detectar manualmente los problemas de esta naturaleza en un diccionario grande. Por lo tanto, es allí donde podemos obtener una ayuda indispensable de la computadora —la infatigable colaboradora capaz de analizar, sin desfallecer, palabra por palabra, comparando las definiciones esparcidas entre cientos de hojas diferentes, calculando estadísticas y verificando exhaustivamente todos los pormenores. Obviamente la máquina no puede sustituir al experto humano, pero sí puede atraer su atención hacia las anomalías y presentarle información que le facilite tomar una decisión más precisa y mejor fundamentada.

Dentro de la lingüística computacional ya se están desarrollando los métodos que permiten automatizar parcialmente el análisis del léxico (Saint-Dizier y Viegas, 1995; Vossen, 2001). Esos métodos pueden ayudar al lexicógrafo en el desarrollo de las definiciones y en la evaluación formal de los diccionarios explicativos.

En este capítulo presentamos varias ideas que para la creación de una herramienta computacional que ayudaría al lexicógrafo a detectar los defectos en la estructura del diccionario y a proponer posibles cambios, específicamente en los casos donde se trata de inconsistencias entre vocablos distantes en el texto.

Aquí abordaremos dos problemas relacionados con la calidad de los diccionarios explicativos:

- Relaciones entre las definiciones en el diccionario.
- División de los vocablos en sentidos, en los casos de polisemia.

El primer punto se refiere a todo un conjunto de problemas que van desde la selección de las palabras, a través de las cuales se tienen que definir otras palabras, hasta la lógica propia de las definiciones. Estos temas se han tratado en la literatura. Las palabras apropiadas para usarse en las definiciones conforman lo que se conoce como vocabulario definidor (LDOCE, OALD) o, en el contexto más teórico, primitivas semánticas (Wierzbicka, 1996). Esas palabras no son únicas, hay muchas maneras posibles de elegirlas. Sin embargo, según nuestro conocimiento, hasta ahora se eligen de forma artesanal para cada diccionario, por prueba y error, sin criterios bien definidos. En la sección 4.2, presentamos un método que da al lexicógrafo la información necesaria para formar un mejor vocabulario definidor.

Acerca de la construcción de definiciones, en la literatura de la lexicografía tradicional (Hartmann, 2001; Landau, 2001; Singleton, 2000) normalmente sólo se dan recomendaciones de carácter muy general acerca de cómo hay que redactarlas. El principio básico es la idea aristotélica de que la definición debe contener el género y las diferencias. Algunas ideas más específicas se basan en el trabajo clásico (Zgusta, 1971): no definir palabras más simples a través de palabras más complejas —más difíciles de entender—; definir, a su vez, todas las palabras empleadas en la definición; evitar el uso de la palabra —o sus derivadas— en su propia definición o en la definición de las palabras definidas a través de ésta (como en nuestro ejemplo con *gallina* y *gallo*); empezar la definición con la parte más importante; hacer las definiciones simples y breves, etcétera (véase, por ejemplo, Landau, 2001: 156–171). En las secciones 4.1 y 4.2, presentaremos los métodos para verificar automáticamente algunos de estos requerimientos y demostraremos que estos requerimientos, en efecto, no son absolutamente compatibles.

El segundo tipo de problemas —el tratamiento de homonimia y polisemia— es aún más difícil de manejar de modo uniforme y

consistente, ya que cada palabra presenta sus propias peculiaridades y, por otro lado, cada lexicógrafo tiene sus propios gustos y experiencia sobre el uso de las palabras específicas. Efectivamente, los pocos ejemplos de uso de cada palabra que una persona puede escuchar o leer en su vida no le dan una información estadísticamente significativa de todos sus usos, y mucho menos de los matices sutiles de su significado. Aquí, el análisis automático de grandes cantidades de texto —tan grandes que no podría una persona leerlos en toda su vida— es una ayuda indispensable.

Algunas consideraciones acerca del problema de la división de palabras en sentidos se presentan en la sección 4.2. En esa sección también analizamos en breve cómo reflejar en el diccionario la polisemia regular (Apresjan, 1974). Otros tipos de verificación automática del diccionario, tales como la verificación de ortografía y la verificación del sistema de marcas de sinonimia y antonimia, se presentan en la sección 4.3.

Finalmente, en la sección 4.4 describimos las funciones de la herramienta «ayudante del lexicógrafo», que está bajo desarrollo en el Laboratorio de Lenguaje Natural del CIC, IPN. En la sección 4.5 presentamos las conclusiones.

4.1 Relaciones entre las definiciones

Para el análisis formal de los diccionarios se usa comúnmente la representación del diccionario como una red semántica o lo que en las matemáticas se llama un grafo dirigido, ya que los métodos son orientados a la estadística o a la teoría de grafos (Kozima y Furugori, 1993; Evens, 1988; Gelbukh y Sidorov, 2002). La descripción detallada de uno de esos métodos se presenta en el capítulo 14. En este capítulo sólo mencionaremos algunas ideas fundamentales de este tipo de análisis, necesarias para entender sus aplicaciones prácticas.

Para el usuario humano, un diccionario tiene como propósito explicar la palabra, maximizando la probabilidad de que la definición contenga otras palabras que el usuario ya conoce. Nótese que, con esto, las paráfrasis sinonímicas en la definición aumentan la probabilidad de que el usuario entienda por lo menos una variante (mientras que «para la computadora» son confusas e inútiles). Otro modo de aumentar la probabilidad de comprensión es usar, en las definiciones, sólo un número restringido de palabras más simples y conocidas (vocabulario definidor). En la práctica es recomendable que sólo se usen alrededor de dos mil palabras, como, por ejemplo, en los diccionarios de inglés de Longman (LDOCE) o de Oxford (OALD).

Para maximizar la probabilidad de que el usuario entienda la definición no debe haber círculos viciosos cortos en el sistema de definiciones. Por ejemplo, el diccionario *Anaya* (Grupo Anaya, 1996) propone las siguientes definiciones:

abeja: insecto que segrega miel.
miel: sustancia que producen las abejas.
convenio: pacto, acuerdo.
acuerdo: pacto, tratado.
tratado: convenio.

En el primer caso, una palabra se define a través de otra y aquella a través de la primera, así que un usuario que no sabe qué son *abeja* y *miel* —y consulta el diccionario para saberlo— no tiene ninguna forma de entender las dos definiciones. En el segundo caso el círculo es de longitud 3: *convenio — acuerdo — tratado —* y nuevamente *convenio*; una persona que no sabe de antemano ninguna de esas tres palabras no entenderá las definiciones. Sin embargo, si el círculo es bastante largo, la probabilidad de que el usuario no conozca ninguna de las palabras es baja, por lo que los círculos largos —a diferencia de los cortos— no son

problemáticos para el uso tradicional del diccionario explicativo del léxico general.

No sucede así con los diccionarios terminológicos explicativos, de voces especiales o técnicas, donde es altamente probable que el usuario no sepa ninguna de las palabras en una cadena de términos explicados uno a través de otro. Así se desarrolla la exposición de la geometría escolar: todos los términos se construyen, aunque indirectamente, de los tres términos «básicos» —*punto*, *recta*, *pertenecer*—, que no se definen, sino se ilustran con dibujos o ejemplos. Nótese que para no crear círculos viciosos, algunas de las palabras usadas en las explicaciones no tienen explicación (ya que en un grafo donde cada nodo tiene vínculos salientes, necesariamente hay ciclos); es decir, las recomendaciones de la lexicografía tradicional de no formar ciclos y explicar cada palabra usada, son contradictorias.

El concepto de palabras «básicas» es acorde con la tradición lexicográfica donde se pretende definir (aunque indirectamente) todas las palabras a través de un conjunto muy restringido de las llamadas *primitivas semánticas* (Wierzbicka, 1996). La diferencia entre el vocabulario definidor y las primitivas semánticas es que las palabras del vocabulario definidor son las únicas palabras que pueden aparecer en las definiciones, y no importa que relaciones se establezcan entre ellas mismas. En cambio, las primitivas semánticas son independientes: no se puede definir unas a partir de otras. Lo que significa que su conjunto es mínimo: no se puede remover de él ninguna palabra (primitiva semántica) sin perder la posibilidad de definir todas las demás palabras en el diccionario. Representando el diccionario como un grafo dirigido (véase capítulo 14), la diferencia es que las palabras del vocabulario definidor deben ser accesibles en un sólo paso por los vínculos del grafo, mientras que las primitivas semánticas pueden ser accesibles en varios pasos. Eso se debe al hecho de

que las palabras del vocabulario definidor están presentes físicamente en las definiciones de las palabras (por eso el nombre de vocabulario definidor), mientras que las primitivas semánticas se presentan virtualmente en las definiciones, por el hecho de ser accesibles en el grafo, pasando tal vez por varios nodos.

Existe una aplicación muy importante, aunque menos tradicional, de los diccionarios, en la cual —al igual que en los diccionarios terminológicos— los círculos, no importa qué tan largos sean, están prohibidos. Aparte de su uso tradicional como fuente de referencia para los usuarios humanos, los diccionarios se pueden utilizar como fuente de información sobre el lenguaje y el mundo real en los sistemas computacionales de inteligencia artificial basados en inferencia lógica. En esta aplicación, no se espera que el sistema experto sepa de antemano palabra alguna, ya que su única fuente de conocimiento sobre el lenguaje es el mismo diccionario, por lo que los círculos viciosos destruyen el sistema de razonamiento lógico al hacerlo entrar en ciclos infinitos. En el uso del diccionario explicativo «para las computadoras» es necesario seleccionar algunas palabras como primitivas semánticas, eliminar sus definiciones (para romper los círculos) y definirlas por medio de programación, no de explicación; de manera semejante a lo que se hace con los términos *punto*, *recta* y *pertenecer* en la geometría escolar.

Entonces, en el análisis y la evaluación de la calidad de los diccionarios con respecto a las relaciones entre las definiciones, surgen los siguientes problemas:

- ¿Cómo escoger las palabras usadas en las definiciones (vocabulario definidor)?
- ¿Cómo escoger las primitivas semánticas (para el uso computacional)?
- ¿Cómo evitar los círculos viciosos (cortos) en las definiciones?

En este sentido, los criterios para mejorar el diccionario serían:

- Tener el menor número de palabras en el vocabulario definidor,
- Tener el menor número de círculos viciosos cortos (en el caso de que las palabras del vocabulario definidor también sean definidas).

4.2 Separación de los significados en sentidos

En la tarea de separación de sentidos de las palabras hay tres posibles problemas:

- El diccionario no contiene algún sentido presente en el texto.
- Varios sentidos del diccionario corresponden a un solo sentido en el texto (y no se trata de neutralización de algunas características).
- Un sentido del diccionario corresponde a varios sentidos en los textos.

Esos casos se analizan en las siguientes secciones.

Falta de sentidos específicos

Uno de los problemas del diccionario se presenta cuando éste no contiene algún sentido específico de una palabra. Por ejemplo, para la palabra *gato* se dan sentidos correspondientes al

1. *animal doméstico que maúlla*
2. *animal felino*

pero no a la

3. *herramienta mecánica para la reparación de carros*.

Este tipo de problemas, a diferencia de algunos otros, no se puede detectar automáticamente con tan sólo analizar el diccionario, sino que es necesario comparar el diccionario con el uso real del lenguaje. Aparte de la introspección del lexicógrafo (que no discutimos aquí), el método más adecuado es verificar si todas las palabras en un corpus grande corresponden a algún significado específico en el diccionario. Dicha verificación se puede hacer de dos maneras: manual y automática. Como siempre, la ventaja de la verificación manual es la calidad y la ventaja de la verificación automática es la rapidez.

Para la verificación manual, en una selección grande de ejemplos del uso de la palabra, cada ocurrencia se marca, manualmente, con uno de los sentidos seleccionados del diccionario. El hecho de que el anotador no encuentre ningún sentido adecuado (como sería con el ejemplo de *gato* arriba mencionado y con el texto *Para reparar su carro Juan tuvo que comprar un gato neumático*) indica el problema en el sistema de los sentidos.

Para facilitar la anotación manual, en nuestro Laboratorio fue desarrollada una herramienta computacional (Ledo-Mezquita *et al*., 2003) que selecciona automáticamente cada palabra significativa del texto, una por una (pasando por alto las palabras funcionales, como preposiciones) y presenta al usuario la lista de posibles significados de la palabra previstos en el diccionario, de entre los cuales el lexicógrafo puede escoger uno o, en su caso, marcar la palabra cuando tiene un sentido no previsto en el diccionario. La herramienta facilita la labor del anotador usando los métodos de lingüística computacional para seleccionar automáticamente el sentido más probable en el contexto dado, para que el anotador, en la gran mayoría de los casos, pueda

simplemente confirmar. Si la preselección automática fue errónea, el programa ofrece al usuario el siguiente sentido más probable (según las heurísticas computacionales usadas), etcétera.

Ya que la labor manual es costosa y aburrida, la manera más económica —aunque no más simple técnicamente— es la verificación puramente automática. En este caso, sólo se verifica que las heurísticas usadas para elegir el sentido de cada palabra lo puedan hacer con un nivel mínimo de certeza. Los métodos correspondientes, en el estado de desarrollo actual, cometen una cantidad significativa de errores de dos tipos. Por un lado, a una palabra se le puede asignar, erróneamente, un sentido no pertinente en el contexto dado, con lo cual queda sin detectar una verdadera falta de sentido de acuerdo al diccionario. Por otro lado, en algunos casos, el error se puede reportar no por un problema real en el diccionario, sino por un fallo de las heurísticas o porque el contexto no presenta la información suficiente para la selección del sentido. Sin embargo, la ventaja de los métodos automáticos es la posibilidad de procesar una gran cantidad de textos prácticamente sin costo alguno. Sólo de esta manera es factible encontrar y considerar los sentidos de frecuencia baja y muy baja.

Entre los métodos para la selección automática de los sentidos de las palabras en el contexto se pueden mencionar diferentes variantes del método de Lesk (Lesk, 1986). La idea básica de este método es buscar automáticamente en el contexto inmediato de la palabra, las palabras usadas en su definición. Por ejemplo, en el contexto *mi gato maúlla cuando ve al perro* está presente una palabra de la definición del primer sentido de nuestro ejemplo. Pero el contexto *Juan no pudo reparar su coche sin un gato* sólo es compatible con el tercer sentido del mismo ejemplo. Existen modificaciones de este método que usan diccionarios de sinónimos (Banerjee y Pedersen, 2002; Sidorov y Gelbukh, 2001) y métodos lingüísticos para la comparación de las palabras; por

ejemplo, en el último contexto *coche* es sinónimo de *carro* y *reparar* es una derivación de *reparación*.

Sea la verificación manual o automática, es importante que en el corpus aparezca un número suficiente de ejemplos de uso de la palabra en cuestión, lo que es muy difícil de lograr para la mayoría de las palabras del diccionario. Efectivamente, según la famosa ley de Zipf (véase capítulo 10), en cualquier texto unas cuantas palabras se repiten muchas veces, mientras que la mitad de las palabras que aparecen en el texto, aparecen en él sólo una vez. Con eso podemos deducir que incluso en un corpus muy grande, casi todas las palabras de un diccionario lo suficientemente completo aparecen muy pocas veces o ninguna. Entonces, el aplicar los métodos descritos arriba a un gran corpus tradicional parece un gran desperdicio de esfuerzo: se procesan muchísimas ocurrencias de unas cuantas palabras del diccionario y muy pocas de casi todas las demás.

Este problema se puede resolver con un corpus de un tipo específico, que llamamos un corpus representativo respecto al vocabulario dado (véase capítulo 12), equivalente a una concordancia de palabras en contexto. Este tipo de corpus se colecciona automáticamente de Internet —el repositorio de textos más grande creado hasta ahora por el ser humano. Para cada palabra del diccionario se colecciona un cierto número de contextos. Así, incluso para las palabras más raras, encontraremos en ese corpus un número de contextos suficiente para la investigación estadística. Lo que soluciona el problema que la ley de Zipf presenta para toda investigación basada en corpus.

Sistema de sentidos demasiado detallado

Otro posible problema se presenta cuando los sentidos son demasiado finos, es decir, a un sentido del texto pueden corresponder

varios sentidos del diccionario e incluso un humano tiene dificultades al elegir el sentido correcto.

Es importante decir que a veces la imposibilidad de escoger un sentido predeterminado está relacionada con la neutralización de algunas características semánticas, cuando el contexto no tiene la suficiente información para elegir un sentido predeterminado.

Por ejemplo, en el diccionario *Anaya* tenemos dos sentidos de la palabra *ventana*:

1. *Abertura, vano en un muro para iluminar y ventila.*
2. *Armazón, marco con cristales para cerrarla*

Ahora vamos a ver los siguientes ejemplos:

3. *Juan saltó por la ventana* $_{1 \text{ (pero no 2)}}$
4. *Juan rompió la ventana* $_{2 \text{ (pero no 1)}}$
5. *Juan saltó por la ventana* $_{2 \text{ (pero no 1)}}$ *y la* $_{2 \text{ (pero no 1)}}$ *rompió*
6. *Juan está mirando a través de la ventana* $_{1 \text{ ó } 2}$
7. *Juan abrió la ventana* $_{1 \text{ ó } 2}$

En los dos primeros ejemplos está muy claro de qué ventana —en el primer sentido o en el segundo— se está hablando, mientras que en el cuarto y quinto ejemplos, cualquiera de los dos sentidos es aceptable. Digamos, en el quinto ejemplo puede ser que se abrió el espacio o que se movió el marco. Como vemos, la interpretación exacta depende del enfoque del hablante u oyente —en este caso el contexto no contiene la suficiente información para elegir. Sin embargo, eso no significa que no existan los dos sentidos, porque sí hay otros contextos donde ambos se distinguen.

Lo interesante del tercer ejemplo está relacionado con el hecho de que el contexto que se encuentra antes de la palabra *ventana* es igual al del primer ejemplo, sin embargo, el sentido de la palabra

es diferente. Es así, porque la segunda parte del contexto contiene la restricción para elegir el sentido —la ventana se rompe, lo que es aplicable solamente a la *ventana*$_2$. Es decir, el contexto contiene datos (el conocimiento del mundo) que solamente son compatibles con *ventana*$_2$. Sin embargo, también se puede argumentar que es el caso similar al primer ejemplo.

Ahora bien ¿cómo un humano escoge el sentido que corresponde al contexto? Se analiza el contexto y se aprovecha el conocimiento del mundo. Si hay algo que sólo es compatible con uno de los sentidos, se puede escoger este sentido, en caso de que la información no esté disponible, se neutraliza la diferencia entre los sentidos. Digamos, en el ejemplo 1 el conocimiento indica que se puede saltar por algún espacio. En el ejemplo 2 se sabe que normalmente las ventanas se hacen de algún material como vidrio que se puede romper y que es parte del armazón que cubre las ventanas. En el ejemplo 3, se sabe que se puede romper la ventana, por lo tanto se abrirá el espacio, y se puede saltar por el espacio abierto. Tal vez, el ejemplo 3 sea el uso metafórico del sentido 2, en lugar del sentido 1. Es interesante que el pronombre refiera a un sentido distinto que el antecedente. En los ejemplos 4 y 5 no hay información adicional, entonces no se puede elegir uno de los sentidos. De hecho, no está claro que tan relevante es la diferencia en el caso de esos ejemplos.

Veamos otro ejemplo. La palabra *agobiarse* tiene dos sentidos en el diccionario *Anaya*:

1. *Causar molestia o fatiga.*
2. *Causar angustia o abatimiento.*

Sin embargo, en el contexto *Él se agobió*, no hay la posibilidad, obviamente, de escoger uno de esos dos sentidos.

En el caso de *ventana* existe una polisemia regular (Apresjan, 1974). Cuando los objetos son «un espacio plano limitado de

los lados», como *puerta*, *esclusa*, etc., puede uno referirse a este objeto como a un espacio, y al mismo tiempo como a un objeto que cubre este espacio. Es decir, de un sentido siempre se puede inferir el otro. En el caso de *agobiarse* no existe el fenómeno de polisemia regular.

Proponemos que la solución al problema de sentidos demasiado finos (y la neutralización de sus diferencias) puede ser la representación del sentido como una jerarquía —en los niveles altos, se definen los sentidos más generales, y en los niveles más profundos, se especifican los sentidos más a detalle.

En el caso de polisemia regular el nivel más alto es la unión de los sentidos de nivel más bajo. Los sentidos en este caso son muy diferentes, por lo tanto, no tienen un sentido generalizado. También la definición debe dar la referencia de que contiene el fenómeno de polisemia regular.

En el caso de *agobiarse* es necesario generalizar los dos sentidos, como, por ejemplo:

Causar una sensación desagradable en el cuerpo humano.

Nótese que no se especifica si el sentimiento está relacionado con el estado físico o el estado psicológico. En el nivel más bajo, se dan las definiciones como están en el diccionario.

La profundidad posible de la jerarquía es el objeto de investigaciones futuras.

Entonces, en algunos contextos se puede determinar cuál de los sentidos de nivel más bajo se usa, y si no es posible, se hace la referencia al sentido generalizado.

Es importante mencionar que el fenómeno que tratamos no es el caso de falta de precisión (*vagueness*, en inglés). La diferencia entre la ambigüedad (en nuestro caso, de sentidos) y la falta de precisión es que, en el caso de la ambigüedad, algo puede tener varios sentidos, y lo que no está claro es cual sentido se usa en el contexto. Mientras que la falta de precisión se refiere al hecho de que un concepto no está bien definido. En los casos que vimos se trata de un problema de ambigüedad.

Ahora bien, ¿cómo se puede aplicar el análisis automático para ayudar al lexicógrafo a detectar las situaciones de sentidos potencialmente similares, que pueden ser tanto casos de polisemia regular como requerir la generalización? Recordemos que es el lexicógrafo quien toma las decisiones y el sistema sólo trata de ayudarle.

Hemos desarrollado un método que permite prever la similitud entre los sentidos de la misma palabra (Gelbukh *et al*., 2003a). Brevemente, la idea es calcular la similitud de los sentidos usando la medida de semejanza entre las definiciones, muy parecida a la medida de similitud de los textos conocida como el coeficiente de *Dice* (Rasmussen, 1992). El coeficiente de *Dice* representa la intersección normalizada de las palabras en los textos. Es decir, se toma de dos textos la intersección textual medida en palabras y se divide entre la suma total. De preferencia, las palabras deben estar normalizadas, por ejemplo, *trabajabas*, *trabajar*, y *trabajaron* se refieren a la misma palabra (lema) *trabajar*.

La medida modificada toma en cuenta adicionalmente los sinónimos de las palabras, porque por definición los sinónimos expresan los mismos conceptos, y para algunas tareas se pueden ignorar los matices de sentido que normalmente tienen los sinónimos. La medida propuesta es como sigue:

$$S(t_1, t_2) = \frac{|W_1 \cap W_2| + |W_1 \circ W_2|}{\max(|W_1|, |W_2|)}$$

donde W_1 y W_2 son conjuntos de palabras en los textos t_1 y t_2, $|W_1 \cap W|_2$ significa que se calcula el número de las palabras (por lemas; recordemos que aplicamos la normalización morfológica automática) que se encuentran en definiciones de ambos sentidos de la palabra y $|W_1 \circ W_2|$ representa el número de intersecciones usando los sinónimos. Es decir, para cada palabra se toma su lista de sinónimos y cada sinónimo de esta lista se busca en el otro texto. En caso de que este sinónimo se encuentre allá, se aumenta

el número de intersecciones. El algoritmo está diseñado de tal manera que cuenta cada intersección sólo una vez —si la palabra o su sinónimo ya se encontró, no se buscan más sinónimos de esa palabra. Eso significa que el número de intersecciones no puede ser mayor que el número máximo de las palabras en uno de los textos (el que contiene más palabras) —el valor que aparece en el denominador. El denominador sirve para una normalización, que significa que el resultado no depende del tamaño del texto.

Aplicamos este algoritmo al diccionario *Anaya* (para la descripción detallada del algoritmo véase capítulo 13), comparando los pares de sentidos de cada palabra, y obtuvimos que cerca de 1% de todos los pares de sentidos son muy parecidos (contienen más de 50% de los mismos conceptos) y cerca de 10% de los pares son sustancialmente parecidos (contienen más de 25% de los mismos conceptos). Consideramos que, por lo menos, para ese 10% de los sentidos parecidos, el lexicógrafo debería de evaluar si sus definiciones son válidas.

Sentidos demasiado generales

Un tercer problema posible se presenta cuando el mismo sentido del diccionario cubre usos claramente diferentes de las palabras. Por ejemplo, la definición de *llave* como *un objeto que se usa para abrir o cerrar algo* cubre tanto el contexto *Juan sacó la llave de su bolsillo y abrió la puerta* como *Juan entró al baño y abrió la llave del agua caliente*. Sin embargo, los hablantes tendemos a considerar «la llave para la puerta» y «la llave para el agua» como cosas muy diferentes, al grado de que el uso de la misma palabra para cosas tan diferentes parece ser pura coincidencia.

El procedimiento descrito anteriormente no detectará ningún problema con esta definición, ya que en ambos contextos se le asignará un sentido del diccionario a la palabra en cuestión.

Tampoco es simple detectar el problema manualmente comparando los contextos en los cuales a la palabra se le asignó el mismo sentido, ya que están en lugares distantes en el corpus, además del alto costo de la gran labor manual necesaria para tal comparación.

Se pueden usar varios algoritmos para la verificación automática de la homogeneidad del conjunto de los contextos en los cuales se marcó la palabra con el mismo sentido. Aquí discutiremos dos métodos basados en el agrupamiento (*clustering*) automático de los contextos. Por contexto de una palabra entendemos las palabras que la rodean en el texto; este concepto se puede precisar de diferentes maneras, desde la oración que la contiene, hasta las palabras que están dentro de una cierta distancia desde la palabra en cuestión. Los dos métodos tratan de analizar diferentes sentidos de una palabra dependiendo de su contexto.

En el primer método, para cada sentido de la palabra se seleccionan los contextos y se agrupan, según una medida de semejanza entre los textos (Alexandrov y Gelbukh, 1999; Alexandrov *et al*., 2000a), en dos grupos, de tal manera que la distancia entre los elementos dentro de cada grupo se minimiza y la distancia entre los dos grupos se maximiza. Esta última distancia da una medida de la calidad de la definición. En el caso de una definición mala los contextos se dividirán claramente en dos o más grupos no parecidos entre sí. En el caso de nuestro ejemplo con la palabra *llave*, un grupo se caracterizará por las palabras *puerta*, *llavero*, *bolsillo*, *insertar*, *olvidar*, mientras que el otro por las palabras *agua*, *caliente*, *fría*, *baño*, *lavar*. Nótese que nuestro método no penaliza indiscriminadamente los sentidos generales: aunque la palabra *objeto* (en el sentido de *cualquier cosa*) es muy general y en consecuencia los contextos de su uso son muy diversos, éstos no se dividen en grupos claramente distinguibles, sino que llenan uniformemente un área amplia.

Otro método (Jiménez-Salazar, 2003) ayuda a verificar todo el conjunto de los sentidos de una palabra en el diccionario.

Los contextos de la palabra dada, encontrados en el corpus, se agrupan automáticamente, también usando alguna medida de semejanza entre dos contextos —por ejemplo, el número de palabras que ambos contextos comparten. La hipótesis del método es que diferentes sentidos de la palabra se usan en diferentes contextos, entonces, los grupos de contextos tales que los contextos son parecidos dentro de cada grupo y diferentes entre grupos diferentes, representan los sentidos diferentes de la palabra. Usando los métodos descritos más arriba, como las distintas modificaciones del método de Lesk, se puede incluso asociar los sentidos presentes en el diccionario para la palabra dada con los grupos de contextos detectados en el corpus. La buena correspondencia indica que el sistema de los sentidos está bien hecho, mientras que la mala es una alarma. Nótese que en este caso el procedimiento de evaluación es puramente automático, pero la resolución de los problemas encontrados necesita la intervención del lexicógrafo.

Resumiendo, el primer método usa las técnicas de clasificación automática y sólo se analiza un sentido de la palabra a la vez, para precisar si la definición del sentido es buena o no. Se supone de antemano que todos los contextos corresponden al mismo sentido. El segundo método usa las técnicas de desambiguación de sentidos de palabras y trata de asociar cada contexto con algún sentido. En caso de no encontrar un sentido apropiado se reporta un posible problema.

4.3 Otros tipos de verificación formal

Aunque no lo discutimos a detalle en este libro, hay muchos otros aspectos del diccionario explicativo que se pueden verificar automáticamente. La base de tal verificación son las propiedades formales (parecidas a lo que en el contexto de las gramáticas for-

males o bases de datos se llaman *restricciones*) que demuestran las relaciones entre los elementos de este sistema tan complejo que es el diccionario explicativo. Aquí sólo damos unos pocos ejemplos.

Verificación de la ortografía y la estructura de los artículos

A diferencia de otros tipos de verificación que discutimos en este capítulo, en esta sección mencionamos brevemente dos tipos de verificación local, que no involucra ninguna comparación de los elementos distantes en el texto del diccionario: la verificación de la ortografía y la verificación de la estructura.

La verificación de ortografía y gramática se aplica a cualquier texto, sin que un diccionario explicativo sea la excepción. Existe una vasta cantidad de libros y una gran variedad de métodos y heurísticas utilizados para este tipo de verificación (Kukich, 1992). Incluso, cualquier procesador de palabras moderno (como Microsoft Word™) contiene herramientas de esta naturaleza, por lo que no dedicaremos más espacio en este libro a la presentación de los métodos de verificación de ortografía y gramática.

Sin embargo, haremos notar que debido a la gran importancia de la perfección de los diccionarios, tiene sentido aplicar métodos que garanticen una mayor calidad de verificación que los tradicionales, es decir, verificación más exhaustiva. Aquí el punto clave es el balance entre el número de errores omitidos y las alarmas falsas (lo que en la literatura especializada se llama la relación entre especificidad —*recall*, en inglés— y precisión). Los métodos de verificación que producen un número demasiado alto de alarmas falsas (de baja precisión) —es decir, los que reportan un posible error que la verificación manual no confirma, muy característico de los métodos de verificación exhaustiva de

alta especificidad (*recall*) — no son prácticos en el uso cotidiano, sin embargo, pueden ser de gran utilidad en la verificación de diccionarios y otros textos importantes.

Entre los métodos de este tipo podemos mencionar la detección de malapropismos. El malapropismo es un tipo de error de la palabra existente en un lenguaje (*real-word errors* en inglés) que consiste en sustituir, por accidente, una palabra con otra igual de correcta y válida en el mismo lenguaje. Lo que en algunos casos resulta en una palabra de una categoría gramatical distinta, tales casos son simples de detectar con un análisis puramente gramatical, por ejemplo: *este artículo es interesante* (en vez de *artículo*). Sin embargo, en otros casos —precisamente los malapropismos— sólo las consideraciones semánticas permiten detectar el error, por ejemplo: *centro histérico de la ciudad*, *en la reserva la casa de venados está prohibida / mi caza tiene tres pisos y está pintada de blanco*. Los métodos existentes de detección de malapropismos (Hirst y Budanitsky, 2003; Bolshakov y Gelbukh, 2003) demuestran usualmente muy baja precisión cuando están configurados para una especificidad (*recall*) razonablemente alta. Eso limita su uso en los procesadores de palabras comunes, pero todavía pueden ser útiles para una verificación más exhaustiva de los diccionarios.

Otro tipo de verificación local es el análisis de la estructura de los artículos. Por ejemplo, verificar que cada palabra significativa (no funcional) usada en el texto del diccionario tenga definición en éste, y en su caso proporcionar al lexicógrafo la lista de palabras usadas sin ser definidas (lo que se llama el vocabulario definidor). También se puede verificar que cada artículo contenga las partes obligatorias —por ejemplo, pronunciación, etimología, explicación y ejemplos. Igualmente se puede observar la numeración correcta de los sentidos y subsentidos, el orden de los elementos del artículo, el orden alfabético de los artículos, las fuentes tipográficas correspondientes a diferentes elementos del artículo, etcétera.

Verificación de las marcas de sinonimia y antonimia

Usualmente los diccionarios explicativos marcan las relaciones básicas entre palabras, tales como sinonimia y antonimia, y en algunos casos —como, por ejemplo, *WordNet* (Fellbaum, 1998)— otras relaciones, tales como meronimia, etc. En el sistema de estas relaciones existen ciertas propiedades (restricciones), por ejemplo:

- simetría: si la palabra *A* es sinónima de la palabra *B* entonces normalmente *B* es sinónima de *A*,
- transitividad: si la palabra *A* es sinónima de la palabra *B* y *B* es sinónima de *C* entonces es probable (aunque en muchos casos no cierto) que *A* sea sinónima de *C*.

Como en otros casos de las propiedades de las relaciones entre las palabras colocadas distantemente en el texto del diccionario, es muy difícil (o por lo menos laborioso) verificar tales restricciones manualmente. Es más fácil hacer que un programa las verifique —y atraiga la atención del lexicógrafo a los posibles problemas detectados. Nótese que se pueden tratar de manera semejante otras relaciones, tales como antonimia, meronimia, etc. Incluso se pueden combinar las verificaciones que involucran relaciones diferentes: por ejemplo, el antónimo de una palabra normalmente no debe ser su merónimo, ni su sinónimo, ni un sinónimo de su sinónimo, etcétera.

Uno puede argumentar que el autor del diccionario, en su sano juicio, no puede marcar la palabra *A* como sinónima de *B* y a la vez marcar la *B* como antónima de *A*, y que entonces no tiene caso en la práctica aplicar las heurísticas que aquí discutimos. Sin embargo, la aplicación de tales heurísticas no sirve al programa para argüir con el autor del diccionario sobre los asuntos lingüísticos, sino para detectar posibles errores mecanográficos

o incluso errores puramente ortográficos, de manera semejante a la detección de malapropismos. Por ejemplo:

cuerdo < ... >. Antónimo: *poco*

en lugar de *loco*. Aquí, el error probablemente ocurrió porque el dedo tocó la tecla *p* en lugar de la cercana *l*, lo que puede suceder en el proceso de preparación del texto. Sin embargo, la única manera que podemos imaginar para detectar automáticamente este error no es la verificación de la ortografía, por muy exhaustiva que esta sea, sino el atraer la atención del lexicógrafo hacia el hecho de que en la definición de la palabra *poco* no se indica, como se esperaba, que tenga como antónimo *cuerdo*.

Otra posible técnica para la verificación de las marcas de sinonimia o antonimia es la comparación de las definiciones. En este caso, más bien se trata de determinar automáticamente qué palabras son sinónimas y verificar si así están marcadas en el diccionario. La hipótesis que aquí se verifica es que las palabras cuyas definiciones son semejantes deben ser marcadas como sinónimas (o antónimas, ya que es difícil interpretar las negaciones automáticamente), sólo esas y ningunas otras. El incumplimiento de esta hipótesis para un par dado de palabras puede significar la marca de sinonimia mal puesta, o bien —mucho más probable— algún problema en las definiciones. Por ejemplo, si las palabras marcadas como sinónimas se definen de manera muy diferente, eso puede indicar inconsistencia en las definiciones. Por otro lado, si dos palabras no marcadas como sinónimas se definen de manera muy semejante, eso puede indicar que las definiciones son demasiado generales para reflejar el significado específico de esas palabras.

Como medida de semejanza se puede usar el número de palabras compartidas entre las dos definiciones, o variantes de este método, como se describió más arriba. Para obtener una medida más estricta, se puede considerar también el orden de las palabras

compartidas, es decir, alguna medida derivada de la distancia (Levenshtein, 1966).

Otra posible fuente de información sobre la sinonimia es un corpus grande de textos. Aquí, la hipótesis a verificar es que los sinónimos se usan en contextos iguales o muy parecidos. Sean las dos palabras en cuestión p_1 y p_2 y sea que aparecen en los dos contextos (digamos, oraciones) C_1 y C_2, respectivamente. ¿Cómo podemos saber que los textos C_1 y C_2 se parecen? No basta con identificar que ambas cadenas son iguales o muy parecidas y que sólo difieren en que en C_1 se usa p_1 y en C_2 se usa p_2 (en vez de p_1), lo difícil es saber si significan lo mismo; por ejemplo, aunque las palabras *vaca* y *cabra* pueden aparecer en contextos iguales —*la leche de vaca* (*cabra*) *es sabrosa y nutritiva*— eso no significa que son sinónimas ya que el significado de estos textos no es idéntico. Una de las formas en que podemos saber si el significado de dos textos, cortos pero diferentes, es idéntico, es con la comparación de diccionarios explicativos, sobre todo terminológicos, ya que en éstos se reduce la ambigüedad (Sierra y McNaught, 2003; Sierra y Alarcón, 2002). Por ejemplo, supongamos que tres diccionarios diferentes dan las siguientes definiciones:

- Diccionario 1: *velocímetro: dispositivo para <u>medir</u> la velocidad de movimiento;*
- Diccionario 2: *velocímetro: dispositivo para <u>determinar</u> la velocidad de movimiento;*
- Diccionario 3: *velocímetro: aparato que se usa para <u>determinar</u> la rapidez de moción de algo.*

Comparando la definición del diccionario 1 con la del diccionario 2, es fácil notar que la palabra *determinar* se usa en vez de *medir*; nótese que el hecho de que ambos textos definan la misma palabra, *velocímetro*, garantiza que el significado de los mismos es idéntico. En la práctica es más común el caso que se presenta en la comparación de las definiciones de los diccionarios 1 y 3: en

este caso no es tan simple detectar automáticamente la semejanza entre los dos textos, sin embargo, existen técnicas para hacerlo (Sierra y McNaught, 2000).

4.4 Herramienta ayudante de lexicógrafo

Las ideas presentadas en las secciones anteriores nos llevaron al desarrollo de una herramienta que permite al lexicógrafo investigar la estructura del diccionario con el fin de detectar y corregir varios tipos de defectos. La herramienta analiza el texto del diccionario y atrae la atención del lexicógrafo a los problemas encontrados, según lo expuesto en las secciones 4.2 y 4.3.

Además, la herramienta proporciona una interfaz interactiva para el desarrollo o la modificación del diccionario. Este software está diseñado para proporcionar al lexicógrafo la siguiente información:

- Visualiza el diccionario en una interfaz gráfica amigable, en un formato tabular, distinguiendo claramente diferentes elementos de cada definición, tales como la pronunciación, etimología, sentidos, subsentidos, ejemplos, relaciones con otras palabras, etcétera.
- Muestra varias características de la palabra elegida, tales como su frecuencia en las definiciones del diccionario, el tamaño de su propia definición, el largo mínimo del ciclo de definiciones en que está involucrada en el sistema (se refiere a las definiciones como *gallina es hembra del gallo* y *gallo es macho de la gallina*), etcétera.
- También proporciona la información sobre el uso de la palabra en un gran corpus de textos y en Internet[12], tales como la

[12] En caso de Internet, la frecuencia aproximada se calcula usando de las máquinas de búsqueda existentes, tales como Google, las cuales determinan el número de los documentos donde se encuentra la palabra.

frecuencia, los contextos del uso, los contextos agrupados, un árbol del agrupamiento de los contextos —desde la división *grosso modo* hasta los matices finos— que se usa para facilitar la división del artículo en sentidos, etc. Aquí, la herramienta permite al usuario elegir los sentidos para las ocurrencias de las palabras en el corpus (véase más abajo).
- Permite buscar las palabras por sus definiciones, por ejemplo: *¿cómo se llama un dispositivo para medir la velocidad de movimiento?* En esto se aplican los métodos de búsqueda inteligente usando sinonimia entre las palabras de la petición y el texto (Sierra y McNaught, 2003; Gelbukh *et al*., 2002c).
- Construye la lista de las palabras usadas en el corpus con una frecuencia considerable pero ausentes del vocabulario del diccionario. Para esto se emplea la normalización morfológica (lematización, cf. *stemming* en inglés) —para que el programa no reporte todas las formas morfológicas de las palabras (por ejemplo, *piensas*) como ausentes al vocabulario (que sólo contiene *pensar*).

En cuanto a los últimos puntos, la herramienta proporciona una interfaz gráfica para el estudio y marcaje del corpus (Ledo-Mezquita *et al*., 2003), permitiendo al usuario elegir los sentidos específicos, entre los que el diccionario proporciona, para cada ocurrencia de cada palabra significativa, con el fin de desarrollar un corpus marcado con sentidos.

Otro módulo de la herramienta ayuda al lexicógrafo a construir un mejor conjunto de las palabras primitivas, por ejemplo, el lexicógrafo debe considerar que el conjunto definidor no debe tener muchas palabras de frecuencia baja. Para esto la herramienta:

- Genera diferentes conjuntos definidores mínimos permitiéndole al usuario controlar varios parámetros del algoritmo de

su generación. Muestra los conjuntos generados junto con la información (tal como la frecuencia) sobre cada palabra incluida en el conjunto.
- Permite al lexicógrafo cambiar manualmente el conjunto definidor generado y verifica que el conjunto cambiado todavía es un conjunto definidor y que es mínimo.
- Permite al lexicógrafo cambiar las definiciones de las palabras e inmediatamente muestra el impacto en los conjuntos definidores que se generan.
- Dada una lista de las palabras que el lexicógrafo quiere que sean no primitivas, verifica si existe algún conjunto definidor que no las contiene. Éste existe siempre y cuando las palabras elegidas no formen círculos viciosos. Si es así, genera una o más variantes de tal conjunto. Si no es así, muestra los círculos, lo que ayuda a eliminar de la lista las palabras que los causan.
- Dada una lista de las palabras que el lexicógrafo quiere que sí sean definidoras, genera uno o varios conjuntos definidores que contengan estas palabras. Si el conjunto definidor no puede ser mínimo, sugiere eliminar ciertas palabras de la lista.

Dado un conjunto definidor mínimo, la herramienta puede:

- Para una palabra no primitiva, mostrar su definición expandida a las palabras definidoras, es decir, la que consiste sólo de las palabras definidoras.
- Para una palabra primitiva, mostrar los ciclos (más cortos o todos) que su definición actual causa en el diccionario.

En este momento no todos los módulos de la herramienta están completamente implementados, aunque disponemos de los algoritmos necesarios y planeamos incorporarlos. Los módulos de la herramienta más desarrollados hasta la fecha son los

del marcaje del corpus y la selección del vocabulario definidor (véase capítulo 14).

4.5 Conclusiones

Un diccionario explicativo es un sistema complejo con numerosas relaciones entre sus elementos y con diferentes restricciones (requerimientos) que las relaciones deben satisfacer para garantizar la integridad y consistencia del diccionario. La verificación de tales requerimientos involucra el análisis no local, es decir, la consideración de los elementos localizados en diferentes lugares del texto, lo que es casi imposible de hacer manualmente, pero que se facilita en gran medida con el uso de computadoras y la aplicación de los algoritmos correspondientes, de diferente grado de complejidad e inteligencia.

La verificación automática no sustituye al lexicógrafo, sino que atrae su atención hacia posibles problemas y le proporciona información necesaria para tomar una decisión informada y consciente, ya sea hacer modificaciones al texto del diccionario o no hacerlas. Más allá de la verificación, las herramientas computacionales permiten el desarrollo interactivo del diccionario, proporcionándole al lexicógrafo la información sobre las relaciones entre una palabra y las palabras relacionadas con ella («cercanas» a ella en la estructura lógica), aunque distantes en el texto plano del diccionario.

Aún más allá, las técnicas computacionales permiten la construcción puramente automática de muchos de los elementos del diccionario —desde el vocabulario y la información estadística, hasta la división de los artículos en sentidos con los ejemplos correspondientes, y la detección de sinonimia entre las palabras—, en la mayoría de los casos a partir del análisis de una gran cantidad de textos —un corpus. En este capítulo sólo hemos

considerado tales posibilidades con el fin de comparar los datos obtenidos automáticamente con los presentes en el diccionario. Otro uso de estos métodos —el cual no hemos discutido— es la construcción automática de un borrador del diccionario completo, para su perfección manual posterior.

Estas consideraciones llevaron al desarrollo —en el Laboratorio de Lenguaje Natural y Procesamiento de Texto del CIC-IPN— de una herramienta computacional que proporcione estos servicios al lexicógrafo, junto con las facilidades para el marcaje semiautomático de los sentidos de las palabras en el corpus.

Parte II

Aplicaciones del PLN con recursos léxicos grandes

Capítulo 5

ANÁLISIS MORFOLÓGICO AUTOMÁTICO BASADO EN UN DICCIONARIO DE RAÍCES*

En la agenda de la lingüística computacional está presente la necesidad de desarrollar varios recursos lingüísticos, entre los que se encuentran los corpus con diferentes tipos de marcas —fonéticas, prosódicas, morfológicas, sintácticas, semánticas, de sentidos de palabras, de anáfora, etc. Otra dirección de investigaciones es el desarrollo de los sistemas que realizan diferentes tipos de análisis lingüístico. Tanto para la preparación de los corpus como para el análisis, una etapa indispensable es la detección de las características morfológicas de las palabras, es decir, el análisis morfológico. Se puede tratar de omitir esta etapa o simularla utilizando diversas heurísticas, sin embargo, la calidad de los recursos y del análisis se mejora con un análisis morfológico exacto.

La morfología estudia la estructura de las palabras y su relación con las categorías gramaticales de la lengua. El objetivo del análisis morfológico automático es llevar a cabo la clasificación

* Con Francisco Velásquez

morfológica de una forma específica de palabra. Por ejemplo, el análisis de la forma *gatos* resulta en:

gato+Noun+Masc+Pl,

que nos indica que se trata de un sustantivo plural con género masculino y que su forma normalizada (lema) es *gato*.

Para el español existen varios sistemas de análisis morfológico, por ejemplo: MACO+ (Atserias *et al*, 1998), *FreeLing* (Carreras *et al*., 2004), FLANOM (Santana *et al*., 1999), o incluso el analizador morfológico integrado en MS Word. Sin embargo, hasta el momento no existe un sistema que realice un análisis exacto y de buena calidad con un diccionario razonablemente grande, que genere los lemas de las palabras y se pueda usar libremente con fines académicos como un módulo independiente. Los sistemas de análisis disponibles en línea por Internet no son de gran utilidad porque tardan mucho en el procesamiento y no pueden integrarse a los sistemas desarrollados por los investigadores.

Por ejemplo, el analizador MACO+ está disponible en forma de un servicio de análisis en línea (no tenemos ninguna información sobre su distribución libre como un módulo independiente fuera de línea para uso con fines académicos, lo que sería muy deseable). La herramienta *FreeLing* (Carreras *et al*., 2004) que sí está disponible libremente y contiene un analizador morfológico del español, sólo contiene un diccionario con 6,000 lemas (5,000 palabras comunes más las palabras de las categorías cerradas). La cobertura de dicho sistema reporta tan sólo 80% de las palabras en los textos, lo que quiere decir que cada quinta palabra no se analiza con su diccionario.*

Los lenguajes según sus características morfológicas —básicamente según la tendencia en la manera de combinar los morfemas— se clasifican en *aglutinativos* y *flexivos*.

* Cabe mencionar que la última versión de *FreeLing* ya contiene un diccionario mucho más grande.

Se dice que un lenguaje es *aglutinativo* si:

- Cada morfema expresa un sólo valor[13] de una categoría gramatical.
- No existen alternaciones de raíces o las alternaciones cumplen con las reglas morfonológicas que no dependen de la raíz específica, como, por ejemplo, armonía de vocales, etcétera.
- Los morfemas se concatenan sin alteraciones.
- La raíz existe como palabra sin concatenarse con morfemas adicionales algunos.

Ejemplos de lenguajes aglutinativos son las lenguas turcas (el turco, kazakh, kirguiz, etc.) y el húngaro.

Por otro lado, un lenguaje es *flexivo* si:

- Cada morfema puede expresar varios valores de las categorías gramaticales. Por ejemplo, el morfema *-mos* en español expresa cumulativamente los valores de las categorías *persona* (tercera) y *número* (plural).
- Las alternaciones de raíces no son previsibles —sin saber las propiedades de la raíz específica no se puede decir qué tipo de alternación se presentará.
- Los morfemas pueden concatenarse con ciertos procesos morfonológicos no estándares en la juntura de morfemas.
- La raíz no existe como palabra sin morfemas adicionales (por ejemplo, *escrib-* no existe como palabra sin *-ir*, *-iste*, *-ía*, etcétera).

Ejemplos de lenguajes flexivos son las lenguas eslavas (ruso, checo, ucraniano, etcétera) y las románicas (latín, portugués, español, etcétera).

[13] Por ejemplo, nominativo es un valor de la categoría gramatical caso.

Normalmente, esta clasificación de lenguajes refleja sólo las tendencias; es decir, muy raras veces un lenguaje es absolutamente aglutinativo o flexivo. Por ejemplo, el finlandés es un lenguaje básicamente aglutinativo, aunque con algunos rasgos de lenguaje flexivo —por ejemplo, varios valores de las categorías gramaticales pueden unirse en el mismo morfema.

En este capítulo sólo consideraremos los lenguajes flexivos y específicamente el español.

En teoría, ya que la morfología de cualquier lenguaje flexivo es finita, cualquier método de análisis basado en un diccionario da resultados igualmente correctos. Sin embargo, no todos los métodos de análisis automático son igualmente convenientes en el uso y fáciles de implementar.

Aquí presentamos una implementación para el español (Gelbukh *et al*., 2003b) de un modelo de análisis morfológico automático basado en la metodología de análisis a partir de generación (Gelbukh y Sidorov, 2003a; Sidorov, 1996). Esta metodología permitió llevar a cabo el desarrollo del sistema con un esfuerzo mínimo y aplicar el modelo gramatical del español transmitido en las gramáticas tradicionales —el más simple e intuitivo.

En este capítulo presentamos un sistema que realiza, para el español, un análisis morfológico de buena calidad con un diccionario razonablemente grande (de 26,000 lemas) y está disponible libremente para la comunidad de investigadores académicos como un módulo independiente bajo una licencia estándar, véanse los detalles en www.Gelbukh.com/agme. El nombre del sistema es *AGME* (Análisis y Generación Morfológica para el Español).

En lo que falta del capítulo se describen los modelos existentes de análisis morfológico automático, y especialmente el modelo de análisis a través de generación, después se presenta el proceso de generación y análisis que se usó en el sistema desarrollado para el español, se explica el procedimiento de preparación de los datos, se muestra brevemente la implementación del sistema y finalmente se dan las conclusiones.

5.1 Modelos de análisis morfológico automático

En la implementación de los analizadores morfológicos automáticos es importante distinguir:

- Modelo de análisis (el procedimiento de análisis).
- Modelo de gramática que se usa en el analizador (las clases gramaticales de palabras).
- Implementación computacional (el formalismo usado).

La razón de la diversidad de los modelos de análisis es que los lenguajes diferentes tienen una estructura morfológica diferente, entonces, los métodos apropiados para lenguajes, digamos, con morfología pobre (como el inglés) o para los lenguajes aglutinativos no son los mejores para los lenguajes flexivos como el español o, digamos, el ruso.

La complejidad del sistema morfológico de una lengua, para la tarea de análisis automático, no depende tanto del número de clases gramaticales ni de la homonimia de las flexiones, sino del número y tipo de las alternaciones en las raíces, que no se puede saber sin consultar el diccionario, por ejemplo: *m*o*ver — m*ue*vo* vs. *d*o*rmir — d*u*rmió* vs. *c*o*rrer — c*o*rro*. En este caso, el tipo de alternación es una característica de la raíz, y el único modo de obtener esta información es consultando un diccionario que contenga estos datos. Al contrario, para el caso de algún idioma aglutinativo como el finlandés, existen alternaciones de raíces, pero normalmente son predecibles sin el uso del diccionario.

Un ejemplo de clasificación de los métodos de análisis morfológico es la clasificación propuesta por Hausser (1999a, 1999b), en la que los métodos se clasifican en: los basados en formas, en morfemas y en alomorfos. Para distinguir entre los sistemas basados en alomorfos y los sistemas basados en morfemas también se usa el concepto de «procesamiento de raíces estático vs. dinámico». De hecho, en el método de alomorfos se guardan todos los alomorfos

de cada raíz en el diccionario, lo que es el procesamiento estático, mientras que en el método basado en morfemas es necesario generar los alomorfos de un morfema dinámicamente, durante el procesamiento.

Consideremos esos métodos con más detalle. Como un extremo, se pueden almacenar todas las formas gramaticales en un diccionario, junto con su lema y toda la información gramatical asociada a la forma. Éste método está basado en las formas de las palabras. En esta aproximación, un sistema morfológico es sólo una gran base de datos con una estructura simple. Este método se puede aplicar para los lenguajes flexivos, aunque no para los aglutinativos, donde se pueden concatenar los morfemas casi infinitamente. Las computadoras modernas tienen la posibilidad de almacenar bases de datos con toda la información gramatical para grandes diccionarios de lenguajes flexivos —de 20 a 50 MB para el español o el ruso.

Sin embargo, tales modelos tienen sus desventajas, por ejemplo, no permiten el procesamiento de palabras desconocidas. Otra desventaja es la dificultad de agregar palabras nuevas al diccionario —hay que agregar cada forma manualmente— que resulta muy costoso. Por ejemplo, los verbos españoles tienen por lo menos 60 formas diferentes (sin contar formas con enclíticos). Para evitar este tipo de trabajo manual se tienen que desarrollar los algoritmos de generación, lo que de hecho puede ser una parte significativa del desarrollo de los algoritmos de análisis, como se presenta en este capítulo.

Otra consideración a favor del desarrollo de los algoritmos, en lugar de usar una base de datos de las formas gramaticales, es el punto de vista según el cual los algoritmos de análisis son un método de compresión del diccionario. El método permite una compresión por lo menos 10 veces mayor. En nuestros experimentos efectuamos la compresión de los diccionarios del ruso y del español en forma de una base de datos con una utilidad de

compresión estándar (zip). El archivo del resultado para el ruso fue cerca de 30 veces más grande que el del diccionario de los sistemas de análisis; en el caso del español, la diferencia entre el tamaño de los archivos fue alrededor de 10 veces a favor del diccionario para el algoritmo.

Una razón más para el uso de los algoritmos de análisis es que es necesario para cualquier tarea que involucre el conocimiento morfológico, por ejemplo, para la tarea de dividir palabras con guiones. También, para la traducción automática o recuperación de información, a veces es mejor tener la información acerca de los morfemas que constituyen la palabra y no únicamente la información de la palabra completa.

Otro tipo de sistemas se basan en almacenamiento de morfemas (de algún alomorfo que se considera el básico) que representan las raíces en el diccionario. Es decir, el diccionario tiene un sólo alomorfo que representa cada morfema. Los demás alomorfos se construyen en el proceso de análisis. El modelo más conocido de este tipo es PC-KIMMO.

Muchos procesadores morfológicos están basados en el modelo de dos niveles de Koskenniemi (1983). Originalmente, el modelo fue desarrollado para el lenguaje finlandés, después se le hicieron algunas modificaciones para diferentes lenguas (inglés, árabe, etc.). Poco después de la publicación de la tesis de Kimmo Koskenniemi, donde se propuso el modelo, L. Karttunen y otras personas desarrollaron una implementación en LISP del modelo de dos niveles y lo llamaron PC-KIMMO.

La idea básica del modelo KIMMO es establecer la correspondencia entre el nivel profundo, donde se encuentran solamente los morfemas, y el nivel superficial, donde hay alomorfos. Eso explica porque en este modelo es indispensable construir los alomorfos dinámicamente.

Además, otra idea detrás del modelo PC-KIMMO es el enfoque hacia el formalismo —los autómatas finitos (transductores), y

de tal modo no pensar en la implementación de los algoritmos (Beesley y Karttunen, 2003). Es decir, los transductores por si mismos son una implementación del algoritmo. La complejidad del modelo gramatical no se toma en cuenta. No obstante, es necesario desarrollar las reglas para poder construir una raíz básica —el alomorfo que se encuentra en el diccionario— de cualquier otro alomorfo. Si en el idioma no existe una estructura muy compleja de alternación de raíces, entonces sí se puede usar el modelo PC-KIMMO (Karttunen, 2003). Para los idiomas donde hay muchas alternaciones de raíces no predecibles —como, por ejemplo, el ruso (Bider y Bolshakov, 1976)— es posible aplicar este modelo, pero el desarrollo de los algoritmos resulta mucho más difícil, aunque no imposible, porque todo el sistema es finito. Cabe mencionar que la complejidad de los algoritmos de este tipo según algunas estimaciones es NP-completa (Hausser, 1999b: 255).

Sin embargo, las ideas relacionadas con la implementación no deben considerarse como predominantes. Claro, si las demás condiciones son iguales, es preferible tener una implementación ya hecha (como, por ejemplo, un transductor), pero consideramos que la complejidad de los algoritmos representa un costo demasiado elevado. Nuestra experiencia muestra que cualquier otro modo de implementación —interpretador de las tablas gramaticales o programación directa de las reglas— es igualmente efectivo, tanto en el desarrollo como en el funcionamiento.

Hay otros modelos de análisis morfológico basados en morfemas. Para el español, Moreno y Goñi (1995) proponen un modelo para el tratamiento completo de la flexión de verbos, sustantivos y adjetivos. Este modelo —GRAMPAL— está basado en la unificación de características y depende de un léxico de alomorfos tanto para las raíces como para las flexiones. Las formas de las palabras son construidas por la concatenación de alomorfos, por medio de características contextuales especiales. Se hace uso de las gramáticas de cláusulas definidas (DCG), modeladas en la

mayoría de las implementaciones en Prolog. Sin embargo, según los autores, el modelo no es computacionalmente eficiente, es decir, el análisis es lento.

La desventaja común de los modelos basados en el procesamiento dinámico de las raíces es la necesidad de desarrollar los algoritmos de construcción de la raíz básica (la que se encuentra en el diccionario), y aplicarlos muchas veces durante el procesamiento, lo que afecta la velocidad del sistema. Por ejemplo, en el español, para cualquier *i* en la raíz de la palabra se tiene que probar cambiándola por *e*, por la posible alternación de raíz tipo *pedir* vs. *pido*. Esa regla puede aplicarse muchas veces durante el análisis para posibles variantes de la raíz (como, por ejemplo, *pido*, *pid-*, *pi-*, etcétera).

Los algoritmos de construcción de la raíz básica a partir de las otras raíces no son muy intuitivos, por ejemplo, normalmente no se encuentran en las gramáticas tradicionales de los idiomas correspondientes. Al contrario, las clasificaciones de alternaciones que están en las gramáticas usan la raíz básica y de esta raíz construyen las demás raíces. Además, los algoritmos de construcción de la raíz básica son bastante complejos: por ejemplo, para el ruso el número de las reglas se estima en alrededor de 1000 (Malkovsky, 1985).

Para evitar la construcción y aplicación de este tipo de algoritmos, se usa el enfoque estático (basado en alomorfos) del desarrollo de sistemas, cuando todos los alomorfos de raíz están en el diccionario con la información del tipo de raíz. Este tipo de sistemas son más simples para el desarrollo.

Para construir el diccionario de un sistema así, hay que aplicar el algoritmo de la generación de las raíces a partir de la raíz básica. Este procedimiento se hace una sola vez. Los algoritmos de la generación de las raíces son más claros intuitivamente a partir de la primera raíz, se hallan normalmente en las gramáticas de idiomas y se basan en el conocimiento exacto del tipo de raíz.

El único costo de almacenamiento de todos los alomorfos con la información correspondiente en el diccionario, es el aumento del tamaño del mismo, que además no es muy significativo. En el peor caso —si todas las raíces tuvieran alternaciones— el aumento correspondiente sería al doble (o triple si todas las raíces tienen 3 alomorfos, etc.). Sin embargo, las raíces con alternaciones usualmente representan, como máximo, 20% de todas las palabras, y además la mayoría de las raíces sólo tiene 2 alomorfos, por lo que el aumento del tamaño del diccionario es relativamente pequeño. Además, se puede aplicar algún método de compresión del diccionario reduciendo así los datos repetidos (Gel'bukh, 1992).

Otra consideración importante para los modelos de análisis basados en diccionarios de morfemas o alomorfos, es el tipo de modelos gramaticales que se usan.

La solución directa es crear una clase gramatical para cada tipo posible de alternación de raíz, junto con el conjunto de flexiones que caracterizan a la raíz. De tal modo, cada tipo de palabras tiene su propia clase gramatical. Sin embargo, el problema es que tales clases, orientadas al análisis, no tienen correspondencia alguna en la intuición de los hablantes y, además, su número es muy elevado. Por ejemplo, para el ruso son alrededor de 1000 clases (Gel'bukh, 1992) y para el checo son cerca de 1500 (Sedlacek y Smrz, 2001).

Otra posible solución es el uso de los modelos de las gramáticas tradicionales. Las gramáticas tradicionales están orientadas a la generación y clasifican las palabras según sus posibilidades de aceptar un conjunto determinado de flexiones (su paradigma). La clasificación según las posibles alternaciones de las raíces se da aparte, porque estas clasificaciones son independientes. De tal modo, las clases corresponden muy bien a la intuición de los hablantes y su número es el mínimo posible. Por ejemplo, en la clasificación según los paradigmas para el ruso, con su morfología bastante compleja, hay alrededor de 40 clases, para el español

se aplica el modelo estándar de las tres clases para los verbos (con las finales *-ar*, *-er*, *-ir*) y una para sustantivos y adjetivos; las diferencias en las flexiones dependen completamente de la forma fonética de la raíz, las peculiaridades adicionales, como por ejemplo *pluralia tantum*, se dan aparte. Esos modelos son intuitivamente claros y normalmente ya están disponibles —se encuentran en las gramáticas y diccionarios existentes.

Sin embargo, no es muy cómodo usar la clasificación orientada a la generación para el análisis directo. Para eso proponemos usar una metodología conocida como «análisis a través de generación». Nuestra idea es tratar de sustituir el procedimiento de análisis con el procedimiento de generación. Es bien sabido que el análisis es mucho más complejo que la generación; por ejemplo, podemos comparar los logros de la generación de voz con los del reconocimiento de voz.

5.2 Modelo de análisis a través de generación

Como hemos mencionado, un aspecto crucial en el desarrollo de un sistema de análisis morfológico automático es el tratamiento de las raíces alternas regulares (*deduc-ir — deduzc-o*). El procesamiento explícito de tales variantes en el algoritmo es posible, pero requiere del desarrollo de muchos modelos y algoritmos adicionales, que no son intuitivamente claros ni fáciles de desarrollar. Para la solución de esta problemática, el sistema que se describe a continuación implementa el modelo desarrollado en (Gel'bukh, 1992) y (Sidorov, 1996) y generalizado en (Gelbukh y Sidorov, 2002). Este modelo consiste en la preparación de las hipótesis durante el análisis y su verificación usando un conjunto de reglas de generación. Las ventajas del modelo de análisis a través de la generación son la simplicidad —no hay que desarrollar algoritmos de construcción de raíces, ni modelos gramaticales

especiales— y la facilidad de implementación. El sistema desarrollado para el español se llama AGME (Analizador y Generador de la Morfología del Español).

Como hemos mencionado, hay dos asuntos principales en el desarrollo del modelo para análisis morfológico automático:

1. Cómo tratar los alomorfos —estática o dinámicamente.
2. Qué tipo de modelos gramaticales usar.

La idea básica del modelo propuesto de análisis a través de generación, es guardar los alomorfos de cada morfema en el diccionario —procesamiento estático, que permite evitar el desarrollo de complejos algoritmos de transformaciones de raíces, inevitable en el procesamiento dinámico— y usar los modelos de las gramáticas tradicionales intuitivamente claros.

Proceso de generación

El proceso de generación se desarrolla de la siguiente manera. Tiene como entrada los valores gramaticales de la forma deseada y la cadena que identifica la palabra (cualquiera de las posibles raíces o el lema).

- Se extrae la información necesaria del diccionario.
- Se escoge el número de la raíz necesaria según las plantillas (véase Sección 4.2).
- Se busca la raíz necesaria en el diccionario.
- Se elige la flexión correcta según el algoritmo desarrollado. El algoritmo es bastante simple y obvio: por ejemplo, para el verbo de la clase 1 en primera persona, plural, indicativo, presente, la flexión es *-amos*, etcétera.
- La flexión se concatena con la raíz.

Proceso de análisis

El proceso general de análisis morfológico usado en nuestra aplicación es bastante simple: dependiendo de la forma de la palabra de entrada, se formula(n) alguna(s) hipótesis de acuerdo con la información del diccionario y la flexión posible, después se generan las formas correspondientes para tal(es) hipótesis, si el resultado de generación es igual a la forma de entrada, entonces la hipótesis es correcta. Por ejemplo, para la flexión *-amos* y la información del diccionario para la raíz que corresponde al verbo de la clase 1, se genera la hipótesis de primera persona, plural, indicativo presente (entre otras), etcétera.

Las formas generadas según las hipótesis se comparan con la original, en caso de coincidencia las hipótesis son correctas.

Más detalladamente, dada una cadena de letras (forma de palabra), se ejecutan los siguientes pasos para su análisis:

1. Quitar una a una sus últimas letras, formulando así la hipótesis sobre el posible punto de división entre la raíz y la flexión (también se verifica siempre la hipótesis de la flexión vacía «» = Ø).
2. Verificar si existe la flexión elegida. Si no existe la flexión, regresar al paso 1.
3. Si existe la flexión, entonces hallar en el diccionario la información sobre la raíz y llenar la estructura de datos correspondiente; si no existe la raíz, regresar al paso 1. En este momento no verificamos la compatibilidad de la raíz y la flexión —esto se hace en generación.
4. Formular la hipótesis.
5. Generar la forma gramatical correspondiente de acuerdo a la hipótesis y la información del diccionario.
6. Si el resultado obtenido coincide con la forma de entrada, entonces la hipótesis se acepta. Si no, el proceso se repite desde

el paso 3 con otra raíz homónima (si la hay) o desde el paso 1 con una hipótesis distinta sobre la flexión.

Nótese que es importante la generación porque de otro modo algunas formas incorrectas serían aceptadas por el sistema, por ejemplo, **acuerdamos* (en lugar de *acordamos*). En este caso existe la flexión *-amos* y la raíz *acuerd-*, pero son incompatibles, lo que se verifica a través de la generación.

En el caso del español, es necesario procesar los enclíticos. Se ejecuta un paso adicional antes de empezar el proceso de análisis —los enclíticos se especifican en el programa como una lista (*-me*, *-se*, *-selo*, *-melo*, etc.). Siempre se verifica la hipótesis de que pueda haber un enclítico al final de la cadena.

5.3 Modelos usados

En el español los procesos flexivos ocurren principalmente en los nombres (sustantivos y adjetivos) y verbos. Las demás categorías gramaticales (adverbios, conjunciones, preposiciones, etc.), presentan poca o nula alteración flexiva. El tratamiento de estas últimas se realiza mediante la consulta directa al diccionario.

Morfología nominal

La variedad de designaciones a que aluden los dos géneros y la arbitrariedad de la asignación del género (masculino o femenino) a los sustantivos impiden, en muchos casos, determinar con exactitud lo que significa realmente el género. Es preferible considerarlo como un rasgo que clasifica los sustantivos en dos categorías diferentes, sin que los términos *masculino* o *femenino* provoquen prejuicio en algún sentido concreto (Llorac, 2000).

No existe un modelo de reglas para la flexión de género en sustantivos. Por lo tanto, en nuestro programa se almacenan todas las formas de sustantivos singulares en el diccionario (por ejemplo, *gato* y *gata*). Los adjetivos siempre tienen ambos géneros, entonces sólo una raíz se almacena en el diccionario: por ejemplo, *bonit-* tanto para *bonito* como para *bonita*. Ahora bien, el tratamiento de la flexión del número puede ser modelado mediante un conjunto de reglas, así que es suficiente tener una sola clase gramatical, porque las reglas dependen de la forma fonética de la raíz.

Por ejemplo, las formas nominales que se terminan en una consonante que no sea *-s*, agregan *-es* en su pluralización (por ejemplo, *árbol — árboles*). Por otra parte, los nombres que se terminan en vocal *-á*, *-í*, *-ó*, *-ú* tienden a presentar un doble plural en *-s* y *-es* (*esquí — esquíes* y *esquís*); la información de doble plural se da con una marca en el diccionario. Algunos de ellos sólo admiten *-s* (*mamás*, *papás*, *dominós*, etc.). La información sobre el plural no estándar se representa a partir de las marcas, en el diccionario, para las raíces correspondientes.

Morfología verbal

Clasificamos a los verbos en regulares (no presentan variación de raíz, como *cantar*), semiirregulares (no más de cuatro alomorfos de raíces, como *buscar*) e irregulares (más de cuatro variantes de raíz, como *ser*, *estar*).

Afortunadamente, la mayoría de los verbos en español (85%) son regulares. Para éstos, usamos los tres modelos de conjugación tradicionales (representados, por ejemplo, por los verbos *cantar*, *correr* y *partir*).

Se usan doce modelos de conjugación verbal diferentes para los verbos semiirregulares, según las alternaciones de su raíz. Nótese que esta clasificación es independiente de los modelos

de paradigmas. Cada modelo tiene su tipo de alternación y su plantilla de raíces. Por ejemplo, en el modelo A1 se encuentra el verbo *buscar* (entre otros). Tiene dos raíces posibles, en este caso *busc-*, *busqu-*; la segunda raíz se usa para el presente de subjuntivo, primera persona del singular del pretérito indefinido de indicativo y en algunos casos del imperativo; la primera raíz se usa en todos los demás modos y personas.

Se usa una plantilla (cadena de números) para cada modelo de conjugación semiirregular. Cada posición representa una conjugación posible; por ejemplo, la primera posición representa la primera persona del singular del presente de indicativo, las últimas posiciones hacen referencia a las formas no personales. Los números usados en la plantilla son de 0 a 4, donde 0 indica que no existe la forma correspondiente; 1 indica el uso de la raíz original; 2, 3 y 4 son las demás raíces posibles. Por ejemplo, para el modelo A2 se usa la siguiente plantilla:

22211211111111111111111111111112221121111111111111111112122111

Eso quiere decir que para las formas 1, 2 y 3 se usa la raíz 2, y para las formas 4 y 5 la raíz 1, etc. Para el verbo *acertar*, que tiene la plantilla A2, las siguientes formas corresponden a la plantilla: *aciert-o, aciert-as, aciert-a, acert-amos, acert-aís,* etc. En total se usan 12 plantillas, correspondientes a 12 clases de verbos en relación con la similitud del número de la raíz que se utiliza para una forma gramatical dada.

Esta estructura nos facilita el proceso de generación de las formas verbales. Nótese que son 61 posibles formas, ya que no tomamos en cuenta las formas verbales compuestas (como, por ejemplo, *haber buscado*) porque cada una de sus partes se procesa por separado.

Al ser mínimo el número de los verbos completamente irregulares (como *ser*, *estar*, *haber*), su tratamiento consistió en almacenar

todas sus formas posibles en el diccionario. El proceso de análisis para estas palabras consiste en generar la hipótesis de un verbo irregular con la flexión vacía; esta hipótesis se verifica a través de la generación, la cual en este caso consiste en buscar la palabra en el diccionario, obtener todas sus variantes y desplegar el campo de información.

5.4 Preparación de los datos

La preparación preliminar de datos consiste en los siguientes pasos:

1. Describir y clasificar todas las palabras del lenguaje (español) en las clases gramaticales, y las marcas adicionales, como por ejemplo, *pluralia tantum*. Esta información se toma completamente de los diccionarios existentes.
2. Convertir la información léxica disponible en un diccionario de raíces. Sólo la primera raíz tiene que ser generada en este paso.
3. Aplicar los algoritmos de generación de raíces para generar todas las raíces, copiando la información de la primera raíz y asignando el número a la raíz generada.

La información para la preparación de este diccionario se tomó de los diccionarios existentes explicativos y bilingües.

5.5 Implementación

La base de datos (el diccionario) es una tabla en formato de Paradox donde se almacenan las raíces e información sobre ellas. El sistema se desarrolló en C++.

El analizador está disponible en www.Gelbukh.com/agme o bien en www.cic.ipn.mx/~sidorov/agme.

El analizador morfológico existe en dos versiones:

1. Ejecutable que toma un archivo de entrada, lo procesa y genera un archivo de salida —en la versión actual no se incluye algún etiquetador adicional para resolver la homonimia de partes de la oración, es decir, se generan todas las posibles variantes morfológicas; se puede usar algún etiquetador disponible para resolver la homonimia.
2. Un módulo DLL que se puede incorporar a los programas directamente; el API de este módulo permite llamar las funciones de análisis para una palabra y retorna valores morfológicos y lemas.

Actualmente el analizador tiene un diccionario de raíces para 26,000 lemas y procesa los textos a una velocidad promedio de 5 KB por segundo con un procesador Pentium IV. Por el momento, para almacenar el diccionario de raíces se usa una base de datos estándar, lo que permite aspirar a hacer el proceso de análisis más rápido en el futuro. Sin embargo, la velocidad existente es aceptable para todas aplicaciones prácticas.

El diccionario fue obtenido usando la conversión automática de un diccionario bilingüe disponible, con la conservación de las clases morfológicas de ese diccionario —gran ventaja de nuestro enfoque de desarrollo de los analizadores. Ese fue el único criterio para la selección de las palabras; no existe ningún obstáculo para hacer el diccionario más grande.

Un ejemplo de funcionamiento del sistema; para las palabras *lee* y *libro* se generan los siguientes resultados:

*lee leer (*VMRP2S0) leer (*VMIP3S0)*
*libro libro (*NCMS000) librar (*VMIP1S0)*

Primero el formato contiene la palabra, después el lema y las características morfológicas de la palabra. Se nota que las palabras originales tienen dos resultados homonímicos de análisis; en el primer caso del mismo lema, en el segundo caso de dos diferentes lemas.

El esquema de codificación es muy similar al estándar *de facto* para el español, PAROLE (Atserias *et al*., 1998), desarrollado dentro del proyecto EAGLES y usado, por ejemplo, en el corpus LEXESP y en el analizador *MACO*+. El primer símbolo corresponde a la parte de la oración; el segundo símbolo expresa algunas características léxicas; en el caso de los sustantivos el tercer símbolo es el género, en el caso de los verbos el modo (indicativo, imperativo, etc.); después se expresa el número o la persona, etcétera.

Los enclíticos se procesan tomando en cuenta las reglas de acentuación y la información correspondiente aparece en la salida.

*trabájaselo trabajar (*VMRP2S0 [se lo])*

Se procesan las palabras compuestas como, por ejemplo, *a_partir_de* o *sin_embargo*, etc. De hecho, ellas se contienen en un diccionario en formato de texto, y el usuario puede editar este diccionario.

5.6 Cómo se puede mejorar el analizador

Para mejorar el analizador consideramos que los comentarios de los usuarios serán muy valiosos.

Por otro lado, actualmente el diccionario del sistema no es muy grande —sólo contiene las raíces que corresponden a 26,000 lemas; sin embargo, es muy fácil para la mayoría de las palabras agregarlas en el diccionario. También esperamos contar con la ayuda de los usuarios en este proceso.

Actualmente, aunque el sistema contiene un algoritmo de generación —el cual es parte de nuestro enfoque de «análisis a través de la generación»—, no se dispone de una interfaz que permita al usuario usar el módulo de generación directamente, esto queda al trabajo futuro —lo que además, implicará la verificación directa de las formas generadas contra un corpus o Internet. Sin embargo, se puede decir que el módulo de generación se prueba indirectamente a través del análisis, es decir, si el resultado del módulo de generación coincide con las hipótesis de análisis, la generación es correcta. Los últimos detalles se verifican corroborando manualmente los resultados del análisis de los textos, con una posible ayuda de los usuarios. Es decir, si hubiera palabras que no se puedan analizar o que se analicen incorrectamente, eso indicaría que hay errores en el algoritmo de la generación o en el diccionario.

Como una dirección de progreso podemos mencionar el desarrollo e implementación de un algoritmo que permita agregar nuevas palabras contestando varias preguntas, basándose en sus formas gramaticales, es decir, sin necesidad de saber todo acerca de los modelos morfológicos del sistema, en especial de los verbos con alternación de raíces.

Otra posible mejora es el desarrollo de un módulo que permitiría corregir errores morfológicos sugiriendo las posibles variantes.

La última posibilidad de mejorar el analizador es agregar un algoritmo que permita procesar los derivados morfológicos.

5.7 Conclusiones

Un analizador morfológico es una pieza indispensable en los sistemas de procesamiento de lenguaje natural. Hasta hace poco

no existía un analizador morfológico para el español disponible libremente como un módulo independiente.

Se presentó aquí un sistema para el análisis morfológico, que implementa el modelo de comprobación de hipótesis a través de generación. Las ventajas de este modelo de análisis reflejadas en su implementación son su simplicidad y claridad, lo que resultó en un tiempo de implementación muy reducido: el desarrollo de los algoritmos principales sólo tomó unos días.

El diccionario actual tiene un tamaño considerable: 26,000 lemas.

Es importante mencionar que el sistema AGME no sobregenera ni sobreanaliza, es decir, sólo procesa las formas correctas.

Se está trabajando sobre la forma de resolver el problema de las palabras desconocidas (una aproximación inicial es la de formular una heurística del más parecido). Como trabajo futuro se sugiere considerar los procesos de derivación (*bella* ⇒ *belleza*) y composición (*agua* + *fiesta* ⇒ *aguafiestas*).

El sistema está disponible, sin costo alguno para el uso académico, como un archivo EXE o DLL de Windows.

Capítulo 6

ANÁLISIS SINTÁCTICO AUTOMÁTICO BASADO EN UN DICCIONARIO DE PATRONES DE MANEJO*

En este capítulo damos un ejemplo de un sistema, desarrollado el CIC-IPN (Gelbukh *et al*., 2002b), que permite hacer análisis sintáctico del español. Los programas que hacen análisis sintáctico se llaman *parsers*.

6.1 Análisis sintáctico automático

El *parser* es un programa que, en general, no depende del idioma de la frase de entrada. Lo que depende del idioma es la gramática —un conjunto de reglas que tienen una forma especial.

Los resultados del análisis sintáctico se muestran en la ilustración 6. El programa hace un análisis morfológico, y después trata de aplicar las reglas gramaticales para cubrir toda la frase. Si toda la frase no está cubierta, entonces, no se puede construir el árbol sintáctico.

---

* Con Sofía Galicia Haro e Igor A. Bolshakov.

Las reglas tienen la siguiente forma:

```
VP(nmb,pers,mean)
        -> VP_DOBJ(nmb,pers,mean)
        -> VP_OBJS(nmb,pers,mean)
```

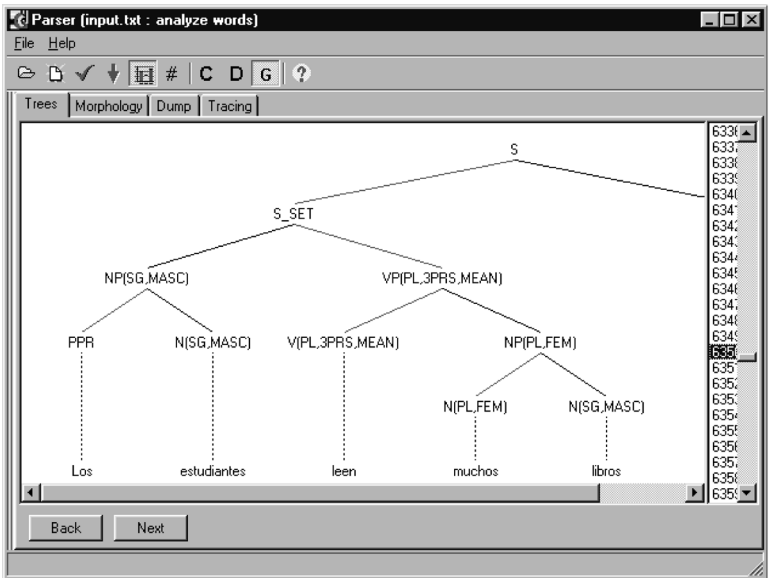


Ilustración 6. El árbol sintáctico.

Esta regla significa que la frase verbal puede ser frase verbal con objeto directo o indirecto.

```
VP_DOBJ(nmb,pers,mean)
        ->  @:VP_OBJS(nmb,pers,mean) dobj_suj:SUJ_DOBJ
            [dobj_suj:SUJ_DOBJ]
            # clavaban sus dardos
        ->  @:VP_DOBJ(nmb,pers,mean) obj:LIS_PP
            # trasladó su fábrica a la frontera
        ->  @:VP_DOBJ(nmb,pers,mean) &mod:VP_MODS
            # ordenó una fila moviendo las sillas
```

Es decir, esta regla determina qué formas gramaticales puede tener el objeto directo.

Las reglas están compiladas automáticamente en forma entendible para el *parser*.

El análisis sintáctico es el núcleo de los sistemas contemporáneos de análisis de texto. Existen diferentes algoritmos de análisis sintáctico, por ejemplo, el algoritmo de Early, el algoritmo de *chart*, etcétera. Normalmente, el programa y el algoritmo de análisis que se implementa no dependen del idioma de la frase de entrada; lo específico de los idiomas se introduce a través de una gramática formal: un conjunto de reglas dispuestas de forma especial.

Además de la existencia de diferentes algoritmos de parseo, hay varios formalismos que se usan para la representación de gramáticas formales, por ejemplo, gramáticas independientes de contexto (CFG, por sus siglas en inglés), gramáticas de estructura de frase generalizadas (GPS), gramáticas de estructura de frase dirigidas por el núcleo (HPSG), gramáticas de adjunción de árboles (TAG), entre otras. Los *parsers* normalmente implementan uno de los algoritmos de parseo y usan algún formalismo gramatical específico.

Un *parser* usa los resultados del análisis morfológico y aplica las reglas gramaticales tratando de cubrir toda la frase. Si toda la frase no está cubierta, como se dijo arriba, entonces no se puede construir el árbol sintáctico.

El *parser* que nosotros desarrollamos usa una gramática independiente de contexto (CFG) y el algoritmo de parseo de *chart*. Ambos componentes son los clásicos en el campo. También tuvimos que desarrollar la gramática para el español. Para evitar la necesidad de escribir todas las reglas de forma manual, se usó la idea de *unificación*, cuando las reglas tienen variables, éstas se sustituyen automáticamente por sus valores durante el parseo o durante la preparación final de la gramática. Actualmente, la gramática tiene alrededor de 150 reglas, a partir de las cuales, des-

pués de aplicar la unificación, se obtienen de manera automática aproximadamente diez mil reglas en la forma estándar de CFG.

Como sucede casi siempre en el análisis sintáctico, el *parser* genera varias, a veces miles, de posibles variantes de la estructura sintáctica; los seres humanos eligen una variante única usando su conocimiento del mundo y el sentido común. El *parser*, en cambio, usa varios métodos estadísticos de estimación de las variantes de la estructura sintáctica, tales como un diccionario semántico grande, o un diccionario de patrones de manejo sintáctico, etcétera.

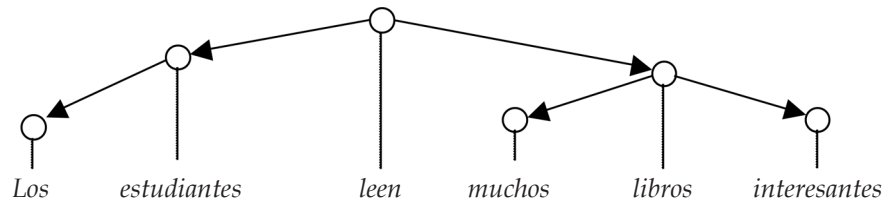
En el esquema se muestra una variante de la estructura sintáctica de una oración simple en español. Con los vértices rojos se marcan las palabras principales en cada combinación de palabras. En el esquema, el árbol se representa usando el formalismo de constituyentes, sin embargo, esa representación se puede convertir en una equivalente usando el formalismo de dependencias. Constituyentes y dependencias son dos modos de representación de la estructura sintáctica de las oraciones.

Los constituyentes se caracterizan por su forma de árbol binario, postulando los nodos que representan en el árbol cada grupo de palabras relacionadas. Cada palabra y cada constituyente de nivel bajo forman parte de un solo constituyente de nivel más alto. Por ejemplo, en el esquema, para la combinación de palabras *los estudiantes* se postula el nodo (constituyente) *NP* (*SG, MASC*); después este constituyente forma parte del constituyente *S_SET*, etc. Normalmente se empieza a construir el árbol de constituyentes desde abajo, juntando las palabras y los constituyentes relacionados. El *parser* usa precisamente este método porque solo así se pueden aplicar las gramáticas tipo CFG.

Las dependencias se representan en forma de flechas. No se postulan los nodos adicionales, sino que las palabras pueden tener varias dependientes. La construcción del árbol de dependencias se empieza desde arriba, buscando la palabra principal en la

oración y marcando sus palabras dependientes con las flechas; el proceso se repite para cada dependiente y así sucesivamente.

El siguiente árbol de dependencias corresponde al ejemplo del esquema:



Eso quiere decir que la palabra *leen* es la palabra principal en la oración y de ella dependen las palabras *estudiantes* y *libros*, de *libros* dependen *muchos* e *interesantes* y de *estudiantes* depende *los*.

A pesar de las diferencias superficiales obvias y de las diferencias en los métodos de parseo, los dos formalismos son equivalentes, es decir, representan la misma estructura, y existe un algoritmo para convertir un árbol de dependencias en un árbol de constituyentes (sin nombres de los nodos intermedios, ya que esos no existen en el árbol de dependencias) y viceversa.

La tendencia de la última década (o quizá de las últimas décadas) en el desarrollo de los sistemas computacionales es hacer los sistemas con un comportamiento menos «mecánico» y más «humano», más adaptado a la naturaleza humana, más inteligente —es decir, amigable con el usuario. Uno de los aspectos más importantes de este trato humano es la interfaz: el modo de comunicación entre el usuario y el sistema. Y el modo más humano de comunicación es el que estamos usando para dirigirnos a nuestro lector: el lenguaje natural.

Por otro lado, es lenguaje natural la forma en la que se almacena y se aumenta el conocimiento de la raza humana. Hoy en día este conocimiento —sobre todo los textos disponibles en Internet—

ya es accesible para el análisis y manejo automático por medio de las computadoras. La tarea de diversas ciencias relacionadas a las tecnologías de lenguaje es facilitar estas operaciones automáticas, con aplicaciones tan importantes como la recuperación de información (Gelbukh y Sidorov, 2000), extracción de información (Montes y Gómez *et al.*, 1999), minería de texto (Montes y Gómez *et al.*, 2001) y muchas otras.

Tradicionalmente, las técnicas y recursos para el análisis del lenguaje natural están orientados al idioma inglés, y el español ha quedado muy descuidado en cuanto a las herramientas computacionales para su procesamiento. Sin embargo, en los últimos años, en los países hispanohablantes se alcanzó un nivel muy alto de investigación en lingüística computacional y sus aplicaciones, sobre todo en España, con muchos grupos de excelencia que trabajan en esta dirección.

También se está consolidando la investigación en lingüística computacional en América Latina, y no es sorprendente que se haya iniciado la construcción de recursos léxicos y plataformas de desarrollo. A continuación presentamos el ambiente de desarrollo lingüístico computacional que hemos construido, y estamos usando, en el Laboratorio de lenguaje natural y procesamiento de texto del CIC-IPN, México.

6.2 Requerimientos en el análisis de lenguaje natural

Un rasgo de los sistemas de análisis de lenguaje natural es que, además del software que ejecuta el análisis, los sistemas de este tipo se apoyan fuertemente en grandes cantidades de datos que poseen estructuras complejas y diversas. Dependiendo de la teoría lingüística y los formalismos elegidos para la realización del sistema, los datos —los diccionarios y las gramáticas— que se usan pueden ser los siguientes:

- *Diccionario morfológico*. Es grande: contiene todas las palabras que se usan en una lengua junto con la información sobre su declinación o conjugación.
- *Gramática sintáctica*. Puede variar desde una muy sencilla —unas cuantas reglas— hasta una muy compleja —muchas miles de reglas fuertemente interrelacionadas.
- *Diccionarios de subcategorización, las combinaciones de palabras y la llamada atracción léxica*. Son muy grandes, ya que contienen la información sobre pares (y a veces conjuntos más grandes) de palabras del lenguaje (Bolshakov y Gelbukh, 2001).
- *Diccionarios semánticos*. Su tamaño y contenido varía mucho dependiendo de la teoría lingüística empleada y del propósito del sistema.
- *Información sobre el mundo real*: los hechos, la estructura física y lógica del mundo, las costumbres, geografía, historia, etc. (por ejemplo: partes del objeto están en el mismo lugar donde está el objeto, los objetos caen hacia abajo, México está en América, España es un reino cuando se trata de cierto periodo histórico, etcétera).
- *Información estadística*. Prácticamente todos los datos arriba mencionados se pueden (y usualmente se deben) acompañar con información estadística, que en muchos casos significa la naturaleza borrosa (*fuzzy*) de las descripciones correspondientes (por ejemplo, ¿en qué grado se puede concluir que la palabra «aviso» es verbo y no sustantivo? ¿en qué grado las palabras «prestar» y «ayuda» se combinan?, etcétera).

Normalmente estos datos, por su tamaño y complejidad, superan en mucho a la parte del software del sistema de procesamiento de lenguaje natural. Teniendo esto en cuenta, nos referimos a los sistemas de este tipo (y específicamente, a los datos lingüísticos) como *lingware*. Tales productos de *lingware*, como gramáticas o diccionarios, se pueden desarrollar e incluso comercializar independientemente del software que los emplea.

Obviamente, el desarrollo del *lingware* implica todo el ciclo de vida de los sistemas complejos, tales como el software. Sin embargo, este ciclo, y específicamente el manejo de los requerimientos, no es bien estudiado. Entre los requerimientos principales se pueden mencionar los siguientes:

- El tipo y el grado de detalles de la información de salida. Hay que mencionar que el grado de detalles y el tipo de información que se usa internamente depende de la calidad requerida más que de los requerimientos de salida, véase el siguiente punto.
- La calidad requerida de los resultados. Se refiere al balance entre la complejidad (el costo) del sistema y la frecuencia de errores. En muchas aplicaciones que emplean sólo estadísticas del análisis (Gelbukh *et al*., 1999), los errores son tolerables; en otras aplicaciones, donde se usan directamente las estructuras obtenidas (Montes y Gómez *et al*., 1999), se requiere mayor calidad de análisis. Los sistemas que permiten un gran número de errores se pueden desarrollar basándose en heurísticas simples.
- El balance entre precisión y especificidad (*recall*). Sin aumentar la complejidad y el costo del lingware, se le puede ajustar a que analice más oraciones, aunque algunas erróneamente, o a que rechace analizar algunas oraciones, mientras que las que acepte las analice correctamente. Por ejemplo, para el análisis correcto de las oraciones tipo «El lunes se enfermó» se puede introducir una regla sintáctica «frase nominal → frase circunstancial», pero con las oraciones tipo «El profesor se enfermó» esta regla produce una variante incorrecta del análisis, por lo cual puede resultar deseable deshabilitarla.
- El tipo y género de los textos a procesar. Todos los ajustes arriba mencionados dependen del tipo de textos a los cuales se aplica el análisis. Por ejemplo, un sistema configurado para los textos

médicos puede mostrar un comportamiento muy diferente cuando se aplica a artículos periodísticos. Esta circunstancia trae como consecuencia que un *lingware* no pueda ser perfecto y general: para cada nuevo tipo de textos se necesita desarrollar el *lingware* o, por lo menos, ajustarlo.

Para formular, procesar y comprobar estos requerimientos, necesitase falta una plataforma de desarrollo que proporcione tanto las herramientas necesarias (tales como el análisis morfológico del español), como la información sobre el comportamiento del analizador. En el Laboratorio de Lenguaje Natural del CIC, IPN se desarrolló tal ambiente, y a continuación se describe brevemente.

6.3 Ambiente de desarrollo

El programa denominado PARSER permite investigar la estructura sintáctica y morfológica de oraciones en español y proporciona la información detallada sobre el comportamiento interno de los componentes del analizador, de esta manera permite la depuración y desarrollo del *lingware* correspondiente. Con este programa se puede aprender el formalismo de gramáticas independientes del contexto. También es posible desarrollar y probar este tipo de gramáticas.

El núcleo del sistema es un analizador sintáctico que emplea una gramática extendida independiente del contexto, con elementos de unificación. Este programa incorpora los resultados de la investigación para compilar patrones de manejo de verbos, adjetivos y sustantivos del español. Los resultados incorporados permiten clasificar las variantes generadas por el analizador de una forma cuantitativa, mediante los pesos asignados a las variantes de acuerdo a los valores de las combinaciones de subcategorización. Los pesos de las combinaciones de subcategorización

son el resultado de un proceso de análisis sintáctico y de una extracción conforme a un modelo estadístico, para determinar los complementos de verbos, adjetivos y algunos sustantivos del español a partir de un corpus de textos.

El uso y la información que proporciona el ambiente

El texto a analizar se especifica con uno de los dos métodos siguientes:

- Se introduce una oración o un texto a analizar, tecleándolo en el programa o usando el comando «Pegar» de Windows.
- Se abre un archivo de texto con las oraciones a analizar.

Para procesar textos completos con varias oraciones, incluso corpus grandes, y obtener los resultados de todas las oraciones, se utiliza una función que permite procesar todas las oraciones a partir de la oración seleccionada, la cuál está marcada en el área «Texto»; esta oración se analiza automáticamente en el momento en que se selecciona y los resultados del análisis se muestran en el área mayor de la pantalla.

Los resultados del análisis de las oraciones se guardan en los archivos de salida y también se pueden analizar de modo interactivo en la pantalla. Específicamente, el programa ofrece la siguiente información (en todas las figuras, se muestran los resultados del análisis de la frase corta «*¡Llamaré a la policía!*»):

Vista de las variantes de la estructura sintáctica, véase la ilustración 7. Esta vista es primordial para el programa. Proporciona las variantes del análisis sintáctico de la frase. En la parte derecha de la pantalla se muestra una lista de las variantes del análisis, de la que se puede seleccionar una variante específica. El árbol sintáctico para esta variante se presenta en la parte central.

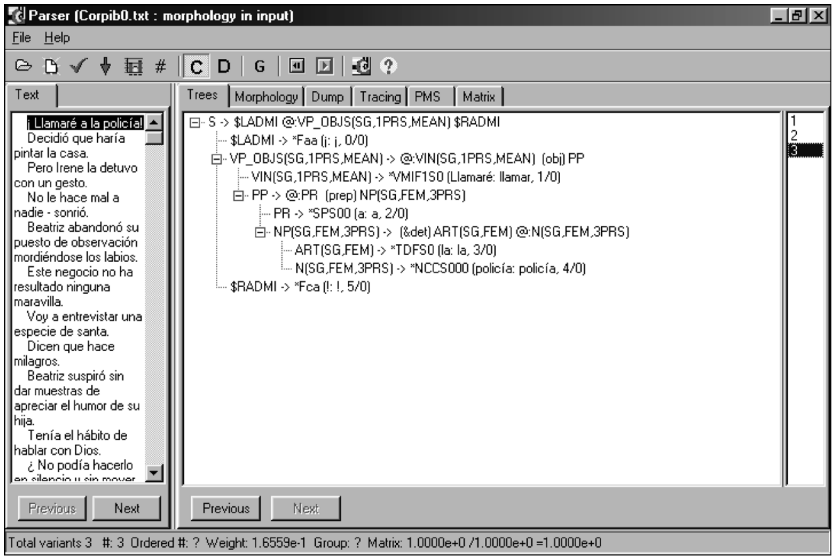


Ilustración 7. Vista del árbol sintáctico.

También se presenta toda la demás información para esta variante específica.

Por defecto, las variantes están ordenadas tal cual fueron generadas por el *parser*. Si se activa la tecnología de ponderación de variantes desarrollada en el Laboratorio (Galicia Haro *et al*., 2001; Gelbukh *et al*., 1998), se realiza el ordenamiento de variantes de acuerdo a la probabilidad de que se trate de la variante correcta. Por ejemplo, en lugar de la clasificación 1, 2, 3 se obtiene la clasificación: 3, 2, 1, lo que significa que la variante 3 es la más probable.

La primera columna muestra la salida del analizador no ordenada, la segunda columna muestra la clasificación correspondiente a los pesos de combinaciones de subcategorización. La tercera columna muestra el peso de la variante. La cuarta columna muestra el porcentaje promedio de clasificación y la quinta columna varios valores correspondientes a los enlaces de dependencias.

Vista de las marcas morfológicas de las palabras de la oración. El sistema adoptó el esquema de marcaje morfológico PAROLE que es un estándar de facto para las aplicaciones PLN para el español. Los números a la derecha representan las diferentes marcas morfológicas. En la palabra *No.* 3 se observan dos marcas: determinante y pronombre.

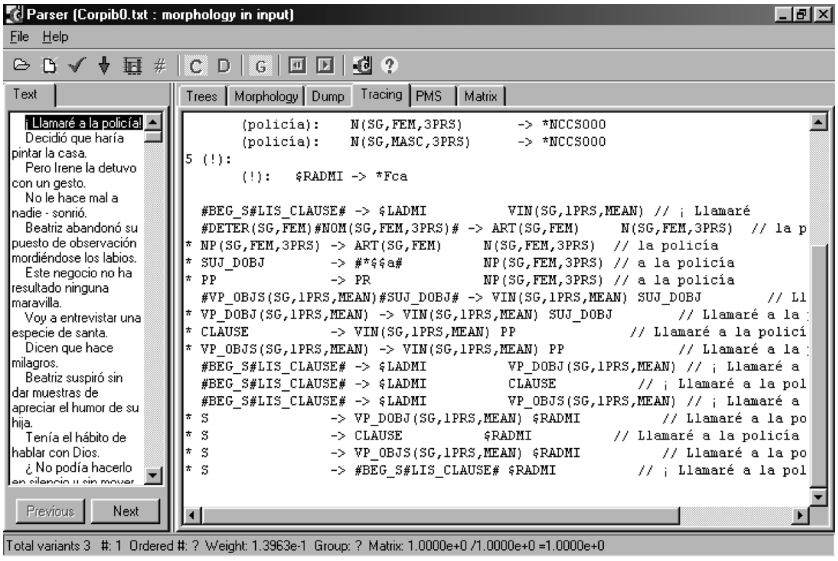


Ilustración 8. Bitácora del análisis.

Vistas de la información sobre el comportamiento del analizador. Estas vistas son muy importantes para el uso del programa, como una plataforma de desarrollo y depuración del *lingware*. La vista «Bitácora» muestra distintas derivaciones que va construyendo el analizador, lo que ayuda a entender cómo interfieren las reglas y por qué se generó el resultado que aparece como la salida (véase la ilustración 8). La vista «Todo» (volcado) muestra en una sola pantalla la lista de todas las variantes de la estructura, (de acuerdo a la selección actual de la opción de constituyentes o dependencias sintácticas), con alguna información adicional

sobre ellas, para facilitar la búsqueda de la variante necesaria y facilitar la comparación.

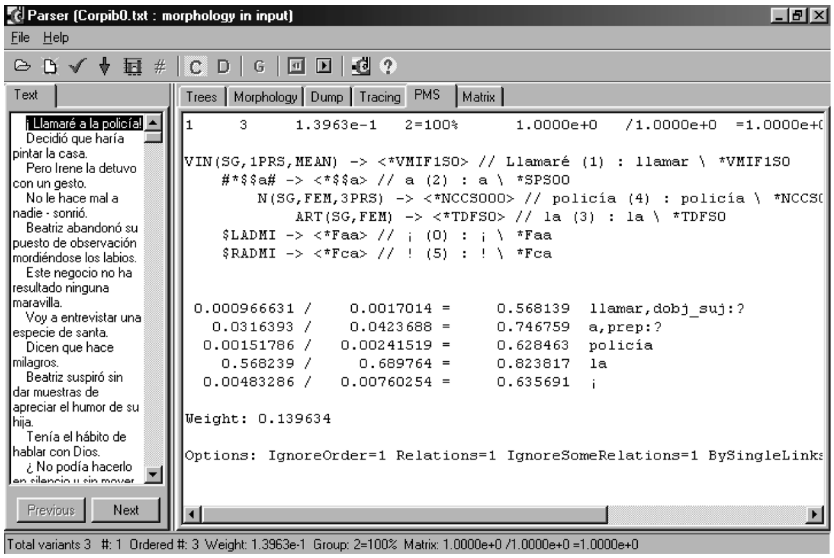


Ilustración 9. Información de subcategorización.

Vistas de la información sobre la compatibilidad de palabras y subcategorización. Las dos últimas pestañas «PMS» (patrones de manejo sintáctico, véase la ilustración 9) y «Matriz» muestran la información del diccionario de patrones de manejo, con los valores obtenidos para las combinaciones de subcategorización. La tecnología de ponderación de las variantes se desarrolló en el laboratorio con la ayuda de esta misma herramienta.

La vista «Matriz» muestra un cálculo de los valores obtenidos con base en una medida de conteo de dependencias correctas. Este conteo se emplea en una nueva tecnología de ponderación que se está desarrollando en el Laboratorio usando la presente herramienta. Se encuentra en etapa de pruebas, todavía sin resultados finales.

6.4 Conclusiones

El desarrollo de analizadores de lenguaje natural involucra la construcción de grandes recursos, con estructuras complejas, pero que no son propiamente códigos del programa. A estos recursos (diccionarios y gramáticas) se les conoce como *lingware*. Su ciclo de vida es parecido al del software. Específicamente, también involucra las etapas de pruebas, formulación, comprobación de requerimientos y depuración.

Se presentó brevemente un ambiente que facilita el desarrollo y la depuración de los analizadores de textos en español. El sistema contiene un analizador morfológico de esa lengua y un *parser* sintáctico que incorpora la tecnología de ponderación de las variantes sintácticas desarrollada en nuestro Laboratorio. El sistema se está usando activamente para la implementación del analizador sintáctico de alta calidad, que se apoya en los diccionarios de compatibilidad de las palabras en español.

Capítulo 7

RESOLUCIÓN DE CORREFERENCIA CON UN DICCIONARIO DE ESCENARIOS

La resolución de correferencia es una de las tareas más importantes en el procesamiento del lenguaje natural (PLN). Es necesaria en una amplia gama de tareas del PLN, de la comprensión de estadísticas del lenguaje, la traducción, y elaboración de resúmenes (Aone y McKee, 1993; Carretero, 1987; Cornish, 1996; Zorro, 1987; Fretheim y Gundel, 1996; Hahn *et al*., 1996; Hirst, 1981; Kameyama, 1997; Mitkov, 1997). A veces las relaciones de correferencia se llaman anafóricas.

Hay dos casos principales de relaciones de correferencia:

1) La correferencia directa, como en el discurso «*he visto una nueva casa ayer. Su cocina era excepcionalmente grande*» (*su* = de *la casa*).
2) La correferencia indirecta, como en el discurso «*he visto una nueva casa ayer. La cocina era excepcionalmente grande*» (*la cocina* = de la *casa*) (ejemplo de (Chafe, 1974)).

En el último caso, la relación de correferencia se sostiene conceptualmente entre las dos palabras diferentes, *cocina* y *casa*; nótese

que no hay ninguna correferencia explícita entre estas dos palabras. Como dijimos, la correferencia se sostiene entre la palabra *cocina* en el texto y la palabra *cocina* que está introducida implícitamente en el discurso por la palabra *casa.* Nos enfocamos principalmente a la resolución de correferencia indirecta, que es el caso más difícil, aunque aparece con menos frecuencia en los textos.

La resolución de correferencia indirecta y aún el descubrimiento de la presencia de correferencia indirecta es especialmente difícil (Erku y Gundel, 1987; Gundel *et al*., 1988; Indirect Anaphora, 1996; Sanford *et al.,* 1983). El marcador más frecuente de correferencia indirecta es la categoría de determinación, expresada en español por los sustantivos con artículos determinados (Ward y Birner, 1994). En el caso de correferencia directa se usan casi siempre pronombres personales, así que por lo menos para los ejemplos como 1) la presencia de correferencia es obvia. En los ejemplos como 2), sin embargo, el artículo determinado no sólo puede usarse en la función anafórica, sino en otras posibles funciones como deixis, contraposición, etcétera (ver más adelante).

Adicionalmente, el artículo determinado no es la única manera en que se expresa la correferencia indirecta. Un tipo particular de marcadores de la correferencia indirecta se encuentra en las expresiones con pronombres demostrativos, como en el ejemplo «*Vendí una casa. ¿Qué puedo hacer con este dinero?*». El ejemplo es aún más complejo ya que el pronombre demostrativo puede tener más sentidos que el artículo determinado.

Así, surgen dos problemas con respecto a la correferencia indirecta: (a) descubrir su presencia, y (b) resolver la ambigüedad de la relación de correferencia. Sin embargo, nos acercaremos al problema en el orden opuesto: nosotros intentaremos resolver la relación de correferencia, y si tenemos éxito consideraremos que el elemento de esta relación se descubre en el discurso. Nuestro trabajo expone una forma de resolución a través del manejo de un diccionario de correferencia indirecta (Gelbukh y Sidorov, 1999).

La estructura del trabajo es la siguiente. Primero, nosotros consideraremos algunos ejemplos útiles. Después formularemos las condiciones necesarias para que exista una relación de correferencia indirecta basados en un diccionario de escenarios. Finalmente, discutiremos un algoritmo que utilice estas condiciones para descubrir la relación en el texto.

7.1 Algunos ejemplos de correferencia indirecta

Consideremos los ejemplos siguientes de correferencia indirecta. Para una discusión extensa necesitamos también información sobre la posibilidad o imposibilidad de ocurrencia de pronombres demostrativos en varios contextos. Las variantes inaceptables se marcan con un asterisco.

1. *Compré una casa. La/*Esta cocina* (*paredes, techo*) *era sumamente grande.*
2. *Compré una casa. Las/*Estas dimensiones eran* 20 × 20.
3. *Compré una casa. El/*Este dueño anterior estaba contento.*
4. *Estaba comprando una casa. Contaba el/*este dinero cuidadosamente.*
5. *Vendí una casa. ¿Qué puedo hacer yo con el/este dinero?*
6. *Compré una casa. Me gustó el/este precio.*
7. *Juan estaba comiendo. La/* Esta mesa estaba sucia.*
8. *Juan estaba comiendo. Estaba oscuro en el/*este bosque.*
9. *Juan estaba comiendo. La/Esta comida estaba deliciosa.*
10. *Juan estaba comiendo. Las/Estas manzanas eran deliciosas.*
11. *Juan estaba cantando. El/Este ruido perturbó a Pedro.*
12. *Juan estaba cantando. Pedro detestó el/este ruido.*
13. *Juan estaba leyendo. Le gustó el/este autor.*
14. *Juan se murió. La/*Esta viuda estaba loca de dolor.*

En el ejemplo 1, la relación de la correferencia indirecta se sostiene entre *cocina* y *casa: la* cocina es la cocina de esa casa. En cada una de estas frases nosotros consideramos la relación del artículo determinado o del pronombre como puramente anafórica; por lo menos en estos ejemplos puede haber ese sentido. Las variantes marcadas con asterisco son imposibles en la interpretación de correferencia.

Sin embargo, en algunos casos el artículo determinado o el pronombre demostrativo no tienen una relación anafórica como la de los ejemplos anteriores. Entonces no son posibles en esta interpretación:

15. **Compró una casa. Las/Estas flores son bonitas.*
16. **Juan estaba comiendo. Estaba oscuro en el/este teatro.*
17. **Juan asistió a una ceremonia religiosa. El mulla y el rabino predicaron un sermón.*

Por otro lado, estos ejemplos son más o menos aceptables si no hay ninguna relación anafórica entre *flores* y *casa*, *teatro* y *comer*. En estos casos, las dos frases no pueden tener ninguna relación directa y la segunda frase puede referirse a un contexto más amplio, o su artículo determinado o pronombre pueden tener una función déictica; el portavoz puede apuntar simplemente con el dedo a las flores o puede estar en el teatro.

Es interesante en especial el caso de los pronombres demostrativos. Aunque la mayoría de los ejemplos anteriores no permite el uso de pronombres en la interpretación anafórica, parecen absolutamente justificados en otras interpretaciones:

18. *Compré una casa. Esta cocina* (*paredes, techo*) *era sumamente grande.*
19. *Estaba comprando una casa. Contaba este dinero cuidadosamente.*

Una de las posibles interpretaciones no-anafóricas del primer ejemplo es la contraposición: «*esta cocina* es grande mientras las otras cocinas no lo son;» en este caso se usa una tensión de entonación especial que no se refleja en el texto escrito. Otra posible interpretación no-anafórica es de nuevo la función déictica: el portavoz está físicamente en *esta cocina* o está mostrándole *este dinero* al oyente.

Otro ejemplo que no permite la relación anafórica es:

20.**Pedro detesta que Juan estuviera comiendo aquí. La/esta mesa estaba sucia.*

Surge una pregunta: ¿Cómo distinguir en el discurso los casos de posible relación de correferencia expresada por el artículo determinado o por el pronombre demostrativo? ¿En qué casos semejantes la relación es posible? En las próximas secciones nosotros enumeramos algunas condiciones necesarias para esta relación.

7.2 Correferencia indirecta como referencia a un elemento del escenario

La correferencia indirecta puede pensarse como correferencia entre una palabra y una entidad que está introducida implícitamente en el texto. Llamamos a las entidades implícitas que se relacionan con una palabra, un *escenario prototípico* de esa palabra. Así, la relación de correferencia se produce entre una palabra y un elemento del escenario prototípico de otra palabra en el texto; el elemento correspondiente no tiene una representación concreta en el texto. La idea de escenarios explícitos fue desarrollada, por ejemplo, en (Shank *et al*., 1980).

Hay tres posibles tipos de correferencia indirecta que dependen de las relaciones entre el antecedente y el anafor: 1) el anafor es una palabra en el texto mientras que el antecedente es un elemento de un escenario de otra palabra; éste es el caso más común; 2) viceversa, un concepto implícito hace referencia a una palabra en el texto (un caso bastante raro); y 3) la referencia se hace entre los conceptos implícitos (un caso aún más raro). Consideremos los ejemplos siguientes:

21. *Juan estaba comiendo. La mesa estaba sucia.*
22. *Juan se murió. La viuda estaba loca de dolor.*
23. *Juan fue enterrado. La viuda estaba loca de dolor.*

Aquí los artículos determinados se usan con la *mesa* y la *viuda.* Sin embargo, estas palabras (y los conceptos correspondientes) no aparecen literalmente en el discurso que los antecede. ¿Cuál es la razón para su determinación? Puede explicarse por la existencia de la relación de correferencia indirecta: *comer* – *mesa*, *morir* – *viuda*, *enterrar* – *viuda.* El antecedente *comer* contiene en su escenario prototípico una ranura para un *lugar* con un posible valor de *mesa* en el primer ejemplo. En el segundo ejemplo el verbo *morirse* se incluye en el campo semántico de la palabra *viuda.* En el tercer ejemplo, el concepto *morirse* es común con los significados léxicos de *viuda* y *enterrar.*

Así, nosotros podemos formular la siguiente versión preliminar de una condición necesaria para la posibilidad de correferencia indirecta:

Condición 1 *(preliminar).* La correferencia indirecta sólo es posible si cualquiera de las siguientes situaciones se sostiene:

- El anafor pertenece al escenario del antecedente.
- El antecedente pertenece al escenario del anafor.
- Los escenarios se intersectan.

Los tres elementos de la Condición 1 corresponden con los tres tipos de correferencia indirecta y no son iguales. Así que la decisión que se tomó en base a la primera parte de la Condición 1 tiene más probabilidades de ser correcta que una basada en la segunda. La tercera parte de Condición 1 es un caso aún más raro y el menos probable. Así, en el algoritmo cuantitativo tienen probabilidades diferentes.

Notemos que la correferencia indirecta puede mezclarse con algunos fenómenos que involucran substitución de una palabra por otra, como el uso de sinónimos, términos más generales o más específicos (véanse los ejemplos 12 y 10, correspondientemente), metáfora (ejemplo 13), o cambiando la parte concreta de la oración en el discurso (derivación). Tales fenómenos son transparentes para la correferencia indirecta, aunque disminuyen la adecuación de correferencia indirecta en el texto y así reducen la fiabilidad del resultado de su descubrimiento. Probablemente a mayor distancia entre las nociones correspondientes, corresponda una menor expresión de correferencia indirecta.

Pensemos en las palabras que pueden sustituirse por un anafor o antecedente *compatible* con él. Ellas son equivalentes a la palabra básica de nuestros algoritmos. Como palabra compatible es posible usar, por ejemplo, un sinónimo, un término más específico o más generalizado, o una metáfora de la palabra básica, aunque no cualquiera de éstos, dependiendo de un contexto específico. Las reglas para determinar compatibilidad están más allá del alcance de este libro. Por ejemplo, la relación de compatibilidad no es simétrica: una metáfora puede aparecer apenas como un elemento del escenario, mientras su aparición como elemento de relación anafórica de superficie es posible. Análogamente, un término más generalizado puede usarse apenas como un anafor de la superficie mientras que el antecedente oculto es su término más específico. Probablemente, las razones para esto están en el mecanismo de la correferencia indirecta cuando la presencia del concepto po-

tencialmente introducido y su representación de superficie (y, así, el tipo de situación) deben clarificarse en el contexto explícito, véanse los ejemplos 8, 10. En el ejemplo 24:

24. *Juan asistió a una ceremonia religiosa. Los mullas predicaron un sermón.*

Sólo la segunda frase clarifica que la ceremonia era musulmana. Estos hechos necesitan investigación extensa. Aquí enfatizamos en que ellos se tomarán en cuenta al aplicar las condiciones de correferencia indirecta entre la compatibilidad del anafor y el antecedente formuladas a continuación.

Ahora podemos formular una versión mejorada de la Condición 1:

Condición 1. La correferencia indirecta sólo es posible si cualquiera de las situaciones siguientes se sostiene:

- El anafor es compatible con un elemento del escenario del antecedente.
- El antecedente es compatible con un elemento del escenario del anafor.
- Sus escenarios tienen intersección (en el sentido de compatibilidad).

7.3 Condiciones sintácticas

En este apartado consideramos únicamente las relaciones de correferencia entre las palabras de frases diferentes (o partes diferentes de una frase compuesta). Una interesante discusión acerca de la posibilidad de las relaciones anafóricas dentro de una frase simple será el tema de un trabajo futuro. Aquí discutimos sólo una complicación más relacionada a las frases incluidas.

Como el ejemplo 20 muestra, la Condición 1 no es la única condición necesaria para la posibilidad de la relación anafórica. Nuestro análisis extenso se relacionó con los problemas inducidos por este ejemplo. A primera vista, la condición siguiente es adecuada:

Condición 2 *(preliminar).* La correferencia indirecta sólo es posible para el nivel semántico más alto de la situación.

En el ejemplo 20, el nivel más alto de la situación es «*Pedro detestó*» y la correferencia indirecta a la situación subordinada no es posible. El nivel semántico más alto corresponde, por supuesto sintácticamente, a la parte principal de la frase compleja. Sin embargo, esta condición no es verdadera para el ejemplo siguiente:

25. *Juan se desanimó al encontrar que su automóvil no funcionaba. La/*Esta batería estaba muerta.*

Puesto que ambos ejemplos, 20 y 25, consisten en dos frases, la diferencia entre ellos no puede depender de las relaciones entre la primera parte y la segunda. Nosotros creemos que la diferencia está en el nivel de coherencia entre las dos frases. En el ejemplo 25 las dos frases son coherentes: una nueva frase puede construirse relacionándolas con la conjunción subordinada *porque*. En el ejemplo 20 ninguna conjunción subordinada es aplicable. Así que nosotros podemos introducir una noción de conexión sintáctica en el sentido descrito anteriormente. Si dos frases pueden conectarse por una conjunción subordinada, entonces están sintácticamente conectadas. Esta noción obviamente tiene una naturaleza discursiva y se relaciona con la coherencia del texto (Downing y Noonan 1995; Fraurud 1992, 1996; Partee y Sgall 1996; Tomlin 1987). Consideremos más ejemplos para esta noción:

26. *Juan detestó que yo comprara una casa. La cocina (paredes, techo) era sumamente grande.*

27. *Juan detestó que yo comprara una casa. El dueño anterior estaba contento.*
28. **Juan estaba satisfecho de que yo comprara una casa. Yo detesté el precio.*
29. **Detesté que Juan estuviera comiendo allí. La comida estaba deliciosa.*
30. **Estaba desanimado de que Juan estaba leyendo allí. Le gustó el autor.*
31. **Yo estaba muy trastornado por la muerte de Juan. La viuda estaba loca de dolor.*

Los ejemplos 26 y 27 sólo son aceptables si nosotros asumimos que hay una relación causal entre la primera y la segunda frases, es decir, si a Juan no le gustan las cocinas grandes (ejemplo 26) y por una razón desconocida odia al dueño anterior (ejemplo 27). Si no hay ninguna relación, entonces los ejemplos son inaceptables. Así, nosotros podemos modificar la Condición 2:

Condición 2. Si no se conectan las partes que contienen el anafor y el antecedente sintácticamente, entonces la correferencia indirecta sólo es posible para el nivel semántico más alto de la situación.

Desgraciadamente, hasta ahora no tenemos ningún algoritmo que descubra la conexión sintáctica. Entretanto, la Condición 2 podría justificarse por una evaluación estadística: ¿que es lo más frecuente en los textos, la presencia o ausencia de la conexión sintáctica? Esto requiere investigaciones extensas, aunque nuestra evaluación muestra que las relaciones subordinadas normalmente tienen una representación de superficie. Ahora usamos la Condición 2 en nuestro algoritmo, aunque los casos en los que la Condición 2 se aplicaría son bastante raros en textos reales.

Enfatizamos que las Condiciones 1 y 2 muestran sólo la posibilidad de la correferencia indirecta pero no su presencia obligatoria, siendo así condiciones necesarias pero no suficientes.

7.4 El algoritmo y el diccionario

El algoritmo, para detectar la correferencia, trabaja de la siguiente forma. Considera cada palabra. Si una palabra se introduce con un artículo determinado o un pronombre demostrativo, entonces es un anafor potencial, y el algoritmo intenta encontrar un antecedente creíble para él. Busca los posibles candidatos para antecedentes basándose en la distancia lineal y en la estructura del anafor potencial. En el caso más simple, es suficiente probar las palabras precedentes, a la izquierda, cuyo peso baja conforme la distancia crece; el algoritmo se detiene cuando encuentra un antecedente o cuando las probabilidades son demasiado bajas. Como hemos mencionado, el algoritmo actual no prueba las palabras dentro de la misma frase simple.

Para cada antecedente potencial se prueban las condiciones descritas antes. Como hemos expuesto, en algunos casos (por ejemplo, midiendo compatibilidad) el grado de satisfacción de la condición puede determinarse como una probabilidad, en lugar de una respuesta sí-no. En ese caso, se combinan (multiplican) las probabilidades para las condiciones y la distancia, y se usa un umbral para decidir cual par de palabras pasa la prueba. Si el candidato satisface todas las condiciones aplicables, la relación anafórica se encuentra.

Para verificar la posibilidad de una relación de correferencia indirecta entre dos palabras se usa un diccionario que enlista los miembros del escenario prototípico para cada palabra. En nuestro caso, usamos un diccionario compilado de varias fuentes, como el diccionario Clasitex, FACTOTUM —que es el diccionario de SemNet derivado del tesauro de Roget— y algunos otros.

Nuestro diccionario de escenarios prototípicos tiene la estructura descrita en (Gelbukh *et al.*, 1999). En ese diccionario cada palabra se relaciona a las palabras que pueden significarse como participantes potenciales de la situación expresados por

la palabra de entrada para el caso más simple. No se especifican los tipos de relaciones entre las palabras, lo que significa que la relación no debería ser una de las relaciones estándar predeterminadas (parte, actuante, etc.). Este tipo de conocimiento no se usa en el algoritmo. Por ejemplo, la entrada del diccionario para la palabra *iglesia* incluye las palabras relacionadas a ésta en uno de los diccionarios mencionados arriba: *sacerdote, vela, icono, oración,* etcétera.

Para verificar la compatibilidad de palabras (sinonimia, generalización, especificación, metáfora) usamos un tesauro compilado con base en el diccionario FACTOTUM de SemNet, *WordNet* y algunas otras fuentes.

7.5 Conclusiones y trabajo futuro

Hemos expuesto aquí un algoritmo basado en un diccionario de filtración de posibles candidatos para los antecedentes de la correferencia indirecta expresados con un artículo determinado o, como caso especial, con un pronombre demostrativo. El algoritmo verifica dos condiciones: 1) la intersección entre los escenarios, y 2) la posibilidad de la relación sintáctica. En la práctica, sugerimos usar este algoritmo sobre todo para el descubrimiento de la presencia misma de la correferencia indirecta, y no sólo para el descubrimiento del antecedente cuando la presencia de la relación anafórica se conoce.

Planeamos extender este trabajo de información al diccionario. Primero, el diccionario debe incluir una clasificación tipo referida a «los pesos» de los elementos del escenario. Los elementos obligatorios tienen el peso más alto; sin embargo, los «optativos» pueden relacionarse más estrechamente a la palabra de entrada o pueden estar más bien lejos de ella. Por ejemplo, la palabra *mesa* en el ejemplo 7 no es obligatoria, pero es un participante

muy probable de la situación de *comer.* Por otro lado, la palabra *bosque* en el ejemplo 8 es un posible, pero no probable, participante de esta situación. En el ejemplo 16 la palabra *teatro* parece ser imposible como un lugar de *comer.* Pueden obtenerse tales pesos de algunos diccionarios semánticos, como el número de relaciones entre las palabras, y también de un corpus grande.

La segunda extensión al diccionario será la especificación de las alternativas. La fuente más importante de alternativas es el grupo de palabras que ocupan la misma ranura en el escenario, o el mismo rol semántico, como *asunto, lugar*, etc. En el ejemplo 17, el escenario para la *ceremonia religiosa* incluiría como posibles participantes *mulla* y *rabino*, y la presencia de cualquiera de ellos es obligatoria, sin embargo, ellos no pueden aparecer, los dos, en el escenario prototípico. Así, el escenario debe especificar un rol que enliste *cura*, *padre*, *papa*, *mulla*, y *rabino.* El rol es marcado como obligatorio, pero la noción *ceremonia religiosa* puede ser el antecedente sólo para una palabra de esta lista. De forma semejante, el escenario de *lugar* para el verbo *comer* incluiría *mesa, bosque*, etc. Así, planeamos agrupar las palabras en los escenarios según su exclusividad mutua en la situación.

Capítulo 8

DESAMBIGUACIÓN DE SENTIDOS DE PALABRAS UTILIZANDO RECURSOS LÉXICOS*

8.1 Descripción del problema

Desde el inicio de los servicios digitales de información, la recuperación de la información ha sido punto de atención para los investigadores en gestión del conocimiento, inteligencia artificial, sistemas computacionales y lingüística computacional. Su importancia radica en que la eficiencia del algoritmo de búsqueda garantice el acceso rápido y pertinente de la información solicitada, dentro de un entorno en que se procesan grandes volúmenes de texto. Este algoritmo de búsqueda es uno de los problemas importantes a resolver dentro de la gestión del conocimiento y el procesamiento del lenguaje natural.

En los últimos años, con el desarrollo de los servicios digitales de información de los medios de almacenamiento masivo y de las redes de telecomunicaciones, se ha difundido el uso de las bibliotecas digitales con búsquedas en línea, dejando atrás a los

* Con Yoel Ledo Mezquita.

algoritmos de búsqueda tradicionales e imponiéndose el desarrollo de algoritmos de búsqueda inteligentes que garanticen mejores resultados en las búsquedas temáticas.

Sin embargo, el desarrollo de métodos que contribuyan a la solución de esta problemática, no ha sido muy investigado para la recuperación de información y en la lingüística computacional. Producir conocimiento sobre las búsquedas inteligentes para los servicios en línea, las condiciones que la determinan y sus mecanismos de procesamiento apoyado en el PLN, son las metas inmediatas a lograr.

Cualquier sistema de procesamiento del lenguaje necesita utilizar abundante conocimiento sobre las estructuras del lenguaje, las cuales son de tipo morfológico, sintáctico, semántico y pragmático. El conocimiento morfológico nos proporciona información de cómo se construyen las palabras; el sintáctico de cómo combinar las palabras para formar oraciones; el semántico sobre lo que significan las palabras y cómo éste contribuye su en el significado completo de la oración, y por último, el pragmático sobre cómo el contexto afecta a la interpretación de las oraciones.

Como ya se mencionó anteriormente en el libro, todas las formas anteriores de conocimiento lingüístico tienen un problema asociado: la ambigüedad. Por lo tanto, la resolución de este tipo de problema es uno de los objetivos principales de cualquier sistema de PLN. Se distinguen diversos tipos de ambigüedades: estructural, léxica, de ámbito de cuantificación, de función contextual y referencial.

En el presente capítulo nos centramos en la resolución de la ambigüedad léxica, la cual aparece cuando las palabras presentan una misma grafía con diferentes significados. A esta tarea se le conoce como desambiguación del sentido de las palabras (DSP; en inglés *WSD*, *word sense disambiguation*).

La resolución de la ambigüedad de los sentidos de las palabras, es un mecanismo lingüístico para definir el sentido más adecuado

de una palabra, según el contexto donde se emplee, que se define en función de los posibles sentidos de las palabras.

Por ejemplo, un mecánico de autos busca «¿dónde comprar un gato?» y obtiene respuestas sobre los «gatos siameses», «gatos monteses» y otros. Un comerciante de frutas busca «producción de lima» y obtiene respuestas sobre la «ciudad de Lima en Perú», «fruta lima», «herramientas para limar metales». Estas imprecisiones son debidas a los distintos sentidos que tienen las palabras. Otro ejemplo, un economista busca «historia del banco» y obtiene respuestas sobre los «bancos de arena», «bancos de madera» y las «instituciones financieras». Un músico busca «formato de letra» y obtiene respuestas sobre el «documento comercial de pago», «letras del alfabeto» y «letras musicales». Estas imprecisiones se deben a los distintos sentidos que tienen las palabras.

La recuperación de información consiste en la tarea de ordenar los documentos, tanto de texto como de multimedia, que pertenecen a una colección dada de acuerdo a la probabilidad estimada de relevancia para las necesidades de información del usuario. Estas necesidades de información son expresadas generalmente por el usuario en función de las respuestas obtenidas a un requerimiento de un lenguaje no formalizado (por ejemplo, sentencias) o un conjunto de términos en un lenguaje natural.

El esfuerzo requerido para la recuperación de la información es notoriamente complejo, debido a que la relación de «relevancia» entre documentos y las necesidades de información, son dependientes de las preferencias e interpretaciones subjetivas del usuario. Además, esta relación es inherentemente no formalizable (Saracevic, 1995).

La enorme disponibilidad actual de documentos almacenados electrónicamente, especialmente en plataformas distribuidas, ha transformado a la recuperación de la información en una disciplina importante. El Internet contiene grandes cantidades de información (en julio de 2000 se sumaba a unas dos mil millones

de páginas que abarcan unos 38 terabytes de datos y que crece 7 millones de páginas diariamente; también contiene alrededor de 450 millones de imágenes (Pimienta, 2000; Lawrence, 2000)), potencialmente interesante y accesible para muchos usuarios: 615 millones en el 2002, de los que 48 millones hablan español; 1030 millones en el 2005, de los que 80 millones hablan español (Global Reach, 2002).

La DSP es considerada como uno de los más importantes problemas de investigación en el procesamiento del lenguaje natural (Wilks and Stevenson, 1996). Es esencial para las aplicaciones que requieren la comprensión del lenguaje y de mensajes, comunicación hombremáquina, la recuperación de información y otros. Es requerido en aplicaciones de:

- Traducción automática: se refiere más que nada a la traducción correcta de información de un lenguaje a otro, según lo que se quiere expresar en cada oración, y no sólo palabra por palabra. Una aproximación a este tipo de traductores en Internet es el Babylon. Muy buen traductor tiene Google.
- Extracción de información y generación de resúmenes. Los nuevos programas deben tener la capacidad de crear el resumen de un documento sobre la base de los datos proporcionados, con un análisis detallado del contenido, sin limitarse nada más a sacar las primeras líneas de los párrafos.
- Reconocimiento de voz. Esta es una de las aplicaciones del PLN que más éxito ha tenido en la actualidad, ya que es común que las computadoras de hoy tengan esta facilidad. El reconocimiento de voz puede tener dos usos posibles: para identificar al usuario o para procesar lo que el usuario dicte. Y existen ya programas comerciales accesibles por los usuarios por ejemplo: *Dragon Naturally Speaking*.
- Recuperación de información. Un claro ejemplo de esta aplicación sería el siguiente: una persona llega a la computadora y

le dice en lenguaje natural (oral o escrito) qué es lo que busca; ésta busca y le dice qué es lo que tiene referente al tema.

En los últimos diez años se han multiplicado las investigaciones para desambiguar palabras automáticamente y crear métodos para identificar los problemas y aplicar las soluciones encontradas. Sin embargo, los sistemas actuales de recuperación de información en línea carecen de un método inteligente que permita mejorar su eficiencia. Por lo tanto, este capítulo se concentra en presentar un método de desambiguación de los sentidos de las palabras usando grandes recursos léxicos para ser aplicado en la recuperación de la información y en la navegación en hipertexto.

El disponer de servicios de recuperación inteligente de información eficientes permitiría mejorar la respuesta a los usuarios que buscan información. En base a esta suposición, el objetivo de este capítulo es presentar el diseño de un nuevo método de desambiguación de los sentidos de las palabras usando grandes recursos léxicos que mejore la pertinencia de la información recuperada.

8.2 Antecedentes

En cierto sentido el trabajo de DSP ha vuelto recientemente a los métodos empíricos y a los análisis basados en corpus que caracterizan algunos de los esfuerzos iniciales por resolver el problema. Con mayores recursos y métodos estadísticos reforzados a su disposición, los investigadores están mejorando los resultados de los pioneros, pero parece que están acercándose al límite de lo que puede lograrse en el marco actual con técnicas y estructuras de representación que impiden distinguir entre lexicones, bases de conocimiento y modelos estadísticos de corpus de texto para el PLN (Dolan *et al*., 2000).

Por supuesto, la DSP es en parte problemática debido a la dificultad inherente de determinar o incluso definir el sentido de la palabra, y esto no parece ser fácilmente resuelto en el futuro cercano (Ravin y Leacock, 2000). No obstante, parece claro que la investigación actual en la DSP podría beneficiarse considerando las teorías del significado, el trabajo en el área léxico semántica y el uso de grandes fuentes de conocimientos.

De los enfoques de análisis para la resolución de la ambigüedad del sentido de las palabras, dos son los que más influencia han tenido en el área:

- Mediante métodos estadísticos.
- Mediante fuentes adicionales de conocimiento.

8.3 Enfoque basado en métodos estadísticos

En este enfoque no se usan algunas fuentes adicionales de conocimiento para la resolución de la ambigüedad (Manning and Schütze, 1999), lo que tradicionalmente se apoya en los corpus.

Un corpus es una muestra amplia de la lengua escrita o hablada, que proporciona las bases para:

- Analizar la lengua y determinar sus características;
- Entrenar a las máquinas, por lo general para adaptar su comportamiento a circunstancias específicas;
- Verificar empíricamente una teoría lingüística;
- Ensayar una técnica o aplicación de ingeniería lingüística a fin de determinar su buen funcionamiento en la práctica.

Existen corpus nacionales que contienen cientos de millones de palabras, pero se trata de corpus construidos para fines concretos. Por ejemplo, un corpus puede contener grabaciones de conduc-

tores de automóvil para simular un sistema de control capaz de reconocer órdenes verbales, y con ello determinar las necesidades de los usuarios con vistas a su producción comercial.

Como ejemplo de métodos usando enfoques estadísticos tenemos: Algoritmos que se basan en los clasificadores bayesianos, las redes neuronales, las máquinas de apoyo vectoriales u otras técnicas de la estadística pura. A continuación se presentan los tres modelos de recuperación más relevantes de información basados en las redes bayecianas.

El primero, denominado Modelo de las redes de interferencias (*Inference Network Model*), fue desarrollado por Croft y Turtle (1990). Está constituido como una red bayesiana en la que se distinguen a su vez dos subredes: la red de documentos, que es fija para una colección dada, y con dos tipos de nodos: término y documento (de los nodos documento salen arcos hacia los nodos término por los que han sido indizados), y la red de la consulta, que se crea cuando el usuario propone una consulta al sistema de recuperación de información y, contiene nodos consulta y nodos término (los arcos van de los nodos término a los nodo consulta).

Ambas subredes se conectan por medio de los nodos término que existen en ambas, desde los nodos de la red de documentos a la de consultas. Una vez que se han estimado las probabilidades, la inferencia se hace a instancias de cada documento sucesivamente y calculando la probabilidad de que la consulta quede satisfecha dado el documento que ha sido observado. Una vez que todas las propagaciones hayan finalizado, se genera el correspondiente ordenamiento de documentos.

Estrechamente relacionado con este trabajo, está el conocido como método Ghazfan (1996) que presenta un modelo básicamente igual al anterior, pero con la diferencia de que cambia la orientación de los arcos. Formalmente, para una consulta, los documentos se ordenan según las probabilidades de pertenecer

a la respuesta. Para ello, se instancian los nodos de la consulta, propagando sólo una vez y calculando así la probabilidad de que cada documento sea relevante para la consulta dada.

Por último, Ribeiro (1996) presenta el modelo llamado Modelo de las redes de creencia (*Belief Network Model*), donde se consideran únicamente dos tipos de nodos, documentos y términos, enlazándolos por arcos de los segundos a los primeros. En este modelo, la consulta se considera como un tipo especial de documento, se propaga también una única vez (como Ghazfan) y se obtiene el ordenamiento según sus probabilidades.

Los tres modelos mencionados hacen suposiciones de independencia entre términos, y por tanto, no establecen arcos directos entre nodos término. Los modelos *Inference Network* y *Belief Network* no aplican ningún algoritmo de propagación como tal, sino que debido a la topología que tienen sus grafos pueden evaluar la probabilidad de manera directa, con resultados análogos a los de la propagación (Campos, 2001).

8.4 Enfoque basado en las fuentes adicionales de conocimiento

En este enfoque se hace uso de de los grandes recursos lingüísticos para la resolución de la ambigüedad tales como los tesauros, los diccionarios de sinónimos, los diferentes tipos de diccionarios morfológicos, etcétera.

Durante los últimos años se han realizado algunas investigaciones sobre DSP basada en el conocimiento. Lesk (1986), propone un método para descifrar el sentido de una palabra en un contexto, según el número de coincidencias que aparecen entre las palabras del contexto y sus definiciones del diccionario explicativo.

Cowie *et al*. (1992) describen un método para resolver la ambigüedad léxica de textos basado en la definición dada en *Longman*

Dictionary of Contemporary English (Diccionario de inglés contemporáneo de Longman) con el que obtiene 47% en cuanto a distinguir los sentidos y 72% para las palabras homógrafas.

Yarowsky (1992) deriva clases de palabras a partir de palabras en categorías comunes del *Roget's International Thesaurus*. Wilks *et al*. (1993) utilizan las co-ocurrencias de datos, extraídos del LDOCE, para construir vectores de contexto y de sentidos asociados a las palabras. Voorhees (1993) define la construcción denominada *hood*, que utiliza los hipónimos para nombres incorporados en *WordNet*.

Sussna (1993) define una métrica basada en la distancia semántica entre los términos de un texto, que consistía en asignar pesos a los enlaces de *WordNet* según los tipos de relación (sinónimos, hiperónimos, etc.) en el conteo del número de arcos del mismo tipo que salen del nodo y en la profundidad total del arco.

En Resnik (1995) se define una métrica basaba en la similitud semántica para las palabras en la jerarquía *WordNet*. Aguirre y Rigau (1996) combinan un conjunto de algoritmos no supervisados para desambiguar nombres del corpus *Semcor*. Rigau *et al*. (1997) combinan un conjunto de algoritmos no supervisados para desambiguar el sentido de las palabras en un corpus no etiquetado.

McHale (1997) presenta los resultados obtenidos de la combinación de *Roget's International Thesaurus* y la taxonomía de *WordNet* con la similitud semántica como medida. Stetina *et al*. (1998) introduce un método para la DSP, basado en un corpus de entrenamiento etiquetado sintáctica y semánticamente. Este método explota la información del contexto de la oración y sus relaciones semánticas.

Resnik (1999) presenta una medida para la semejanza semántica presente en una taxonomía IS-A, y la aplica en un algoritmo para resolver las ambigüedades sintácticas y semánticas. Mihalcea y Moldovan (1999) exponen un método para desambiguar

nombres, verbos, adverbios y adjetivos de un texto, sobre la base de la referencia del sentido proporcionado por *WordNet*. Montoyo (2001) presenta un método que resuelve la ambigüedad léxica de nombres en textos escritos en inglés basado en la taxonomía de nombres que utiliza *WordNet*.

Al valorar los dos enfoques analizados, el enfoque mediante métodos estadísticos y el enfoque mediante fuentes de conocimiento, decidimos usar el segundo enfoque.

Las ventajas del enfoque basado en fuentes adicionales de conocimiento son:

- Su claridad: se puede verificar el algoritmo paso por paso.
- Su decisión es totalmente explícita.
- En teoría este enfoque puede alcanzar el 100% de eficiencia.
- No depende de procesos de aprendizaje y de entrenamiento.

8.5 Método propuesto

La idea del algoritmo de Lesk original (Lesk, 1986) se basa en la búsqueda de intersección de las definiciones de un diccionario explicativo con las definiciones de palabras del contexto en el mismo diccionario. La idea del algoritmo de Lesk simplificado consiste en la búsqueda de las intersecciones de definición de la palabra en cuestión con las palabras del contexto.

Estamos proponiendo un método de desambiguación de los sentidos de las palabras que es una combinación de ambos métodos y utiliza adicionalmente la información léxica obtenida de diferentes diccionarios. Además usamos la posición relativa de las palabras del contexto para ponderar sus pesos.

El algoritmo realiza varios pasos.

Primero, hacemos la normalización morfológica de las palabras del documento resolviendo la homonimia morfológica

usando un conjunto de las heurísticas sintácticas. Cada palabra se sustituye por su forma normalizada (lema), por ejemplo, *trabajo, trabaja, trabajé, trabajaron* se sustituye por el infinitivo *trabajar*.

Después, se eliminan las palabras auxiliares del documento; tales como preposiciones, artículos, verbos auxiliares, pronombres. En este caso, después de este procedimiento el documento contiene solamente verbos, adjetivos, sustantivos y adverbios.

Realizamos desambiguación de los sentidos de las palabras en un dominio local limitado a cierto número de palabras.

Hicimos los experimentos con diferentes tamaños de la ventana. Nuestros experimentos demostraron que los mejores resultados se obtienen con la ventana de tamaño igual a 17 palabras. Entonces, usamos este tamaño de la ventana. El método pondera la influencia de la palabra de contexto dependiendo de la distancia de la palabra en cuestión: Más cercana es la palabra, mayor influencia tiene. De manera empírica llegamos a la conclusión que los coeficientes exponenciales son mejores para ponderar dicha influencia dentro de la ventana mencionada.

En el siguiente paso del algoritmo, para cada palabra con varios sentidos se calculan valores utilizando ambos algoritmos basados en el método de Lesk. Después aplicamos el mismo procedimiento para los sinónimos obtenidos de un diccionario de sinónimos y para las palabras que forman otras relaciones léxicas (hiperónimos, hipónimos, etc.) obtenidas de *WordNet*. Cada valor se pondera según la distancia de la palabra en cuestión. Sumamos los valores obtenidos para cada sentido aplicando diferentes pesos a los valores retornados: para los algoritmos de Lesk utilizamos el peso 1, para los sinónimos 0.9, y para otras relaciones léxicas 0.8. Los valores óptimos de estos pesos se obtuvieron haciendo las series de experimentos.

8.6 Evaluación del método

En total hicimos experimentos para 4,287 sentidos de 872 palabras. Usamos 80 contextos que contenían en total 1,293 palabras. Hicimos tres tipos de evaluaciones usando el algoritmo de Lesk original, el algoritmo de Lesk simplificado, y el método propuesto.

Ejemplos de los datos de entrada se presentan en la tabla 1.

Tabla 1. Ejemplos de datos de entrada.

N	Contexto	Palabras a desambiguar	Sentidos analizados
1.	*Dejar a un gato normalmente sociable en una pensión o en la clínica.*	6	31
2.	*Detienen a militar implicado en el asesinato de policías*	5	14
3.	*No se pierda las razones para suscribirse al mejor diario digital.*	6	50
4.	*Participa en nuestro concurso de fotografía digital*	4	12

Los resultados obtenidos con el método propuesto son mejores que en caso de los otros métodos. En la siguiente tabla se presentan los resultados globales para varias frases.

Tabla 2. Comparación del método propuesto con los métodos de base.

			Lesk Original		Lesk Simplificado		Método propuesto	
N	**Palabras**	**Sentidos**	**E**	**P (%)**	**E**	**P (%)**	**E**	**P (%)**
1	6	31	2	33	2	33	5	83
2	5	14	3	50	2	33	4	67
3	6	50	3	33	2	22	3	33
4	5	12	2	40	1	20	2	40
5	9	33	7	78	4	44	5	56
6	9	34	6	67	3	33	4	44
7	8	59	4	50	4	50	5	63
8	8	21	4	50	4	50	6	75
9	9	37	6	67	5	56	7	78
10	11	66	5	45	4	36	7	64
11	10	32	7	70	4	40	5	50

En la tabla, N significa el número secuencial, E significa éxito, y P significa precisión.

El método propuesto obtuvo en promedio 63% de precisión, que es 13% mejor que el método original de Lesk que obtuvo 50% y 22% mejor que el método de Lesk simplificado que obtuvo 41%.

8.7 Conclusiones

Nuestro método realiza la desambiguación de los sentidos de las palabras del contexto basado en la comparación de los sentidos de la palabra analizada en relación al contexto y a los sentidos de

las palabras que conforman el contexto. Teniendo en cuenta la influencia de cada palabra del contexto según la distancia a la que se encuentra de la palabra analizada y la influencia de la semejanza en función de varios recursos léxicos.

Los resultados obtenidos demuestran que nuestro método obtuvo mejor precisión que los otros dos métodos anteriores.

Nuestro método propone una nueva forma de determinar la semejanza de contextos usando diferentes recursos léxicos tales como el diccionario explicativo con definiciones normalizadas, sinónimos, antónimos, merónimos (parte de), holónimo (contiene a), hipónimos, hiperónimos en el cual cada recurso aporta a la determinación de la semejanza obteniéndose mejores resultados.

Nuestro método hace una atenuación de la influencia en el peso de las palabras según la distancia a la que se encuentren de la palabra analizada, de forma tal que palabras más cercanas tienen mayor influencia y que palabras más lejanas tienen menor influencia expresándose de forma discreta bajo una curva exponencial.

La disponibilidad de un método de desambiguación de sentidos de palabras para la recuperación inteligente de información en buscadores es de tanta necesidad en la actualidad por el gran volumen de información que existe y se consume diariamente por las personas.

Capítulo 9

RECUPERACIÓN DE DOCUMENTOS CON COMPARACIÓN SEMÁNTICA SUAVE

Los buscadores de documentos tradicionales carecen de la inteligencia suficiente para garantizar la pertinencia de los resultados de la búsqueda respecto a los intereses de los usuarios (Baeza-Yates y Ribeiro-Neto, 1999; Frakes y Baeza-Yates, 1992; Kowalski, 1997). Los problemas básicos de la búsqueda intelectual se relacionan con los documentos que contienen las formas declinadas o conjugadas de las palabras de la petición, palabras relacionadas por el sentido, etc. Por ejemplo, si el usuario necesita encontrar documentos sobre religión (y formula su petición como *religión*), un sistema con comparación «rígida» (exacta) entre petición y texto no encontrará documentos que mencionen las palabras *religiones*, *cristianismo*, *sacerdote*(*s*), etc. —documentos que un sistema inteligente debe encontrar (Alexandrov *et al*., 1999; Gelbukh *et al*., 1999).

Un sistema de búsqueda inteligente también debe proporcionar varias opciones de ordenamiento de los documentos y la presentación mejorada de los resultados de la búsqueda.

En este capítulo se describe un método que implementa este tipo de inteligencia usando diccionarios lingüísticos grandes con estructura simple (Gelbukh y Sidorov, 2002a). El método fue diseñado y elaborado en un sistema de software inteligente que permite a los usuarios ejecutar, a través de Internet, búsquedas temáticas en la base de datos del Senado de la República Mexicana, la cual contiene tanto la información legislativa (las leyes y normas), como el diario de debates del Senado.

Primero se describe el método propuesto y las opciones de la búsqueda que le dan inteligencia al sistema, es decir, se expone qué tipos de diccionarios se usan y qué resultados se pueden obtener aplicándolos en recuperación de textos dentro del sistema. Después se enlistan las opciones de búsqueda que proporciona el sistema. Finalmente, se muestra la forma de presentación de los resultados y se dan las conclusiones.

9.1 El método

La idea principal del método es permitir la búsqueda por palabras parecidas (relacionadas) en lo que respecta a su sentido (Gelbukh *et al*., 1999; Fellbaum, 1998) y también tomando en cuenta todas las formas gramaticales de las palabras (Gelbukh, 2000a; Hausser, 1999; Koskenniemi, 1983).

Para realizar las búsquedas se realiza el enriquecimiento de la petición (Gelbukh, 2000a; Gusfield, 1997) usando los diccionarios lingüísticos grandes con estructura simple. Para cada palabra el diccionario contiene la lista de palabras relacionadas, marcándose el grado de relación (las formas gramaticales, los sinónimos, los antónimos, etc.). El enriquecimiento de la petición consiste en que se agregan a la petición estas palabras para cada palabra originalmente incluida por el usuario (véase la sección 3).

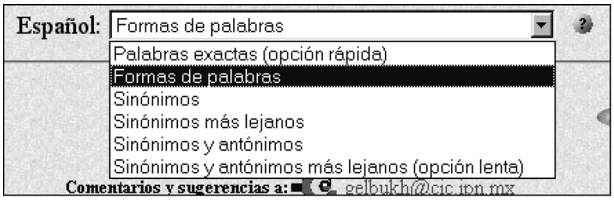


Ilustración 10. Opciones de búsqueda con diccionarios.

El usuario tiene la posibilidad de elegir el «grado de inteligencia» deseado (escogiendo en el menú una de las opciones de búsqueda), es decir, controla el grado de inexactitud de la comparación. En el proceso de enriquecimiento sólo participan las palabras relacionadas con el grado de relación seleccionado. En el sistema antes mencionado, este control se realiza mediante el campo que se presenta en la ilustración 10 y ofrece las siguientes opciones:

- *Palabras exactas*: no se efectuará ningún tipo de enriquecimiento de la petición (se efectuará comparación exacta entre la petición y el texto del documento). Por ejemplo, si se busca por la petición *pequeño*, sólo se encontrarán los documentos que contienen exactamente la forma *pequeño*.
- *Formas de palabras:* se efectuará el enriquecimiento de la petición con las formas morfológicas de las palabras. Por ejemplo, si se busca por la petición *pequeño*, se encontrarán los documentos que contienen las formas *pequeño, pequeña, pequeños* y *pequeñas*.
- *Sinónimos*: se efectuará el enriquecimiento de la petición con los sinónimos cercanos de las palabras de la petición. Por ejemplo, si se busca por la petición *pequeño*, se encontrarán también los documentos que contienen la palabra *chico*. Esta opción implica también la comparación de las formas morfológicas de las palabras: *pequeñas*, *chicas*.

– *Sinónimos más lejanos*: se efectuará también el enriquecimiento de la petición con los sinónimos lejanos de las palabras de la petición. Por ejemplo, si se busca por la petición *pequeño*, se encontrarán los documentos que contienen la palabra *reducido* (*reducida*, etc.), además de las formas *pequeño*(*s*), *chico*(*s*).
– *Sinónimos y antónimos*: se efectuará también el enriquecimiento de la petición con los antónimos exactos de las palabras de la petición (además de los sinónimos cercanos y lejanos y las formas morfológicas). Por ejemplo, si se busca por la petición *pequeño*, se encontrarán los documentos que contienen las formas *pequeño*(*s*), *chico*(*s*), *reducido*(*s*), *grande*(*s*).
– *Sinónimos y antónimos más lejanos*: se efectuará también el enriquecimiento de la petición con los antónimos lejanos de las palabras de la petición. Por ejemplo, si se busca por la petición *pequeño*, se encontrarán los documentos que contienen las formas *pequeño*(*s*), *chico*(*s*), *reducido*(*s*), *grande*(*s*), *amplio*(*s*), etc. Esta opción da el máximo grado de suavidad de la comparación que fue implementado en el sistema.

Estas opciones son compatibles con las demás opciones de búsqueda, tales como la búsqueda por frase. Por ejemplo, con la opción de sinónimos y la frase, si se busca por la petición *manual chico*, se encontrará el documento que contiene la frase *libro pequeño*.

El sistema puede ser fácilmente extendido a las búsquedas con generalización y por palabras relacionadas: por la petición *religión* se encontrará *cristianismo* y *sacerdote*; por la petición *mesa* se encontrará *mueble* (Gelbukh *et al*., 1999). Técnicamente, sólo es necesario agregarle un diccionario (un grado de relación más lejana que la sinonimia) con la misma estructura: para cada palabra encabezada (por ejemplo, *religión*) se enlistan las palabras relacionadas (*cristianismo*, *sacerdote*, etc.). En el sistema actual no se implementó este diccionario por consideraciones de espacio.

Usualmente los usuarios indican en la petición la primera forma de la palabra (singular para los sustantivos, masculino singular para los adjetivos, infinitivo para los verbos) cuando se trata de la palabra «en general», es decir, independientemente de su forma morfológica. Si el usuario especificó una forma concreta de la palabra (por ejemplo, *encontraron*), lo más probable es que sabe que esa es la forma en que se encuentra la palabra en el documento que necesita. Entonces, en la implementación actual, la opción de morfología se limitará a las primeras formas de las palabras en la petición. Es decir, por la petición *pequeño* se encontrará tanto *edificio pequeño* como *casa pequeña*, pero por la petición *pequeña* no se encontrará *edificio pequeño*.

9.2 Diccionarios

El método usado para la comparación suave en el sistema se basa en los diccionarios lingüísticos. En esta sección, se describen los diccionarios incorporados en el sistema. Cada tipo de búsqueda por palabras, con la generalización de sentido o por formas gramaticales, corresponde a un diccionario conveniente. Ya que el tamaño de los diccionarios es muy grande, se hizo un análisis estadístico para determinar las palabras más útiles y sólo se incluyeron éstas. En particular, como palabras encabezado, los verbos se dan sólo en infinitivo, que es la forma en la que los verbos se usan típicamente en las peticiones de los usuarios.

En el sistema actual no se usa ningún tipo de razonamiento ni de generación ni de análisis dinámico de las formas en el momento de ejecución de la búsqueda. Gracias a esto, en su implementación práctica se pudieron usar las herramientas y métodos estándares del manejador de bases de datos Informix.

Diccionario morfológico

El diccionario morfológico permite encontrar documentos que contienen formas de la palabra diferentes de las que se teclearon por el usuario en el campo de búsqueda: por ejemplo, para la petición *pensar*, se encontrará *piensa*.

El diccionario consiste en conjuntos de palabras. La primera palabra de cada conjunto es la palabra encabezado. Si esta palabra aparece en la petición del usuario, las demás palabras de su conjunto se buscarán también en los documentos.

El diccionario implementado en el sistema contiene 61,490 artículos (conjuntos) que contienen, en total, 786,366 formas de palabras.

A continuación se da una muestra del diccionario:

{*ababol* ababoles}
{*ababoles* ababol}
{*abacorar* abacorábamos abacoráis abacoráramos abacoráremos abacorásemos abacoré abacoréis abacoró abacora abacoraba abacorabais abacoraban abacorabas abacorad abacorada abacoradas abacorado abacorados abacoramos abacoran abacorando abacorará abacorarán abacorarás abacoraré abacoraréis abacoraría abacoraríais abacoraríamos abacorarían abacorarías abacorara abacorarais abacoraran abacoraras abacorare abacorareis abacoraremos abacoraren abacorares abacoraron abacoras abacorase abacoraseis abacorasen abacorases abacoraste abacorasteis abacore abacoremos abacoren abacores abacoro}
{*abadía* abadías}
{*abadías* abadía}

Sinónimos más cercanos

El diccionario de sinónimos más cercanos permite encontrar los documentos que contienen los sinónimos de las palabras que se teclearon por el usuario en el campo de búsqueda: por ejemplo, para la petición *senado*, se encontrará *asamblea*.

En nuestra implementación, el concepto de sinonimia comprende también, técnicamente, el concepto de las formas de las palabras (se tratan como sinónimos de la palabra encabezado). Por esta razón, en el diccionario se incluyeron las formas morfológicas de las palabras, además de sus sinónimos.

El diccionario contiene 65,378 artículos (conjuntos), 1,822,164 palabras en total. A continuación se da una muestra del diccionario:

{*abacá* cáñamo cabuya fibra fibras}
{*abacería* almacén almacenes bodega bodegas colmada colmado colmados comercio comercios comestibles ultramarinos}
{*abacero* comerciante comerciantes negociante negociantes proveedor proveedora proveedores suministrador suministradora suministradoras tendero tenderos vendedor vendedora vendedoras vendedores}
{*abaco* anotador anotadora anotadores capitel capiteles columna columnas contador contadores coronamiento remate remates tabla tablas tanteador}

Sinónimos más lejanos

El diccionario de sinónimos más lejanos permite encontrar los documentos que contienen los sinónimos más lejanos de las

palabras que se teclearon por el usuario en el campo de búsqueda: por ejemplo, para la petición *condenar*, se encontrará no sólo *acusar*, sino también *criticar*.

El diccionario contiene 65,378 artículos (conjuntos), 1,927,976 palabras en total.

Todos los sinónimos y antónimos más cercanos

El diccionario de todos los sinónimos y antónimos más cercanos permite encontrar los documentos que contienen los sinónimos y antónimos más cercanos de las palabras que se teclearon por el usuario en el campo de búsqueda. Por ejemplo, para la petición *condenar*, se encontrará no sólo *acusar* y *criticar*, sino también *(no) liberar*.

El diccionario contiene 65,379 entradas, 2,310,735 palabras en total.

Todos los sinónimos y antónimos más lejanos

El diccionario de todos los sinónimos y antónimos más lejanos permite encontrar los documentos que contienen los sinónimos y antónimos más lejanos de las palabras que se teclearon por el usuario en el campo de búsqueda. Por ejemplo, para la petición *condenar*, se encontrará no sólo *acusar*, *criticar* y *(no) liberar*, sino también *(no) aprobar*.

El diccionario contiene 65,379 entradas, 2,341,701 palabras en total.

Interfaz del usuario

El buscador del Senado de la República Mexicana es un sistema computacional que ejecuta búsquedas temáticas en la Base de datos del Senado (principalmente, en los Diarios de debates del Senado).

Por diseño, el buscador está basado en el manejador de bases de datos que se usaba anteriormente en el Senado, Informix. Por eso, parte de la funcionalidad del sistema se fundamenta en las posibilidades internas de búsqueda de Informix y su extensión para búsquedas en textos; esta extensión se proporciona por el módulo *Excalibur Text Search*. Para el desarrollo de la interfaz de Internet se usó el módulo *Web DataBlade*, también de Informix.

Opciones de búsqueda

Como cualquier programa de búsqueda, el buscador desarrollado permite al usuario especificar varias opciones de búsqueda y de ordenamiento de resultados, ejecutar las búsquedas en diferentes modos y en diferentes colecciones de documentos (véase la ilustración 11).

La opción *Ley* permite al usuario escoger entre la colección de documentos para la búsqueda: constitución, Diario de debates o ambas colecciones; se puede agregar más colecciones. La opción *Ordenar por* proporciona al usuario las posibilidades de diferentes ordenamientos de los documentos encontrados: por fecha, por similitud a la petición (relevancia) o por número de palabras encontradas en el documento.

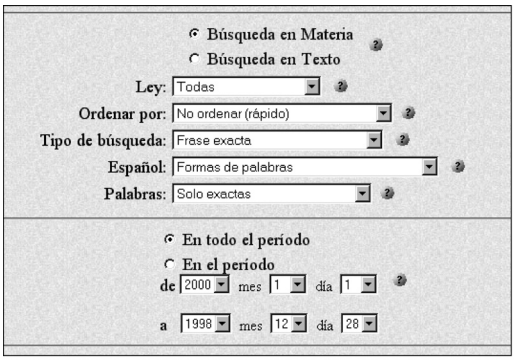


Ilustración 11 Opciones de búsqueda.

Dependiendo de la selección de *Tipo de búsqueda*, la petición puede ser una sola palabra (por ejemplo, *senado*), varias palabras (por ejemplo, *ley senado eléctrico*), una expresión lógica (por ejemplo, *senado y (eléctrico o energético) y no industria*), o una frase (por ejemplo, *industria eléctrica*).

La opción *Palabras* permite usar la función incorporada en el manejador de bases de datos para la búsqueda de palabras escritas con errores (digamos, sin acentos). Los valores posibles de esta opción son: *exactas*, *con diferencia en 1 letra* y *parecidas*. La última variante significa que la diferencia puede ser de más de una letra. La última opción funciona de una manera mucho más lenta.

La opción *Buscar sólo en materia o en texto completo* ofrece diferentes variantes. Si se elige la opción *Búsqueda en Materia*, el sistema localizará las palabras sólo en los títulos («materias») de los documentos. Si se elige la opción *Búsqueda en Texto*, el sistema buscará y localizará la palabra en el texto completo del documento, incluyendo el título. Con esta opción se proporciona también la información de cuántas veces se encontraron en el texto las palabras de la petición. De esta manera, se mide la relevancia del documento: a mayor número de apariciones de las palabras, mayor es su relevancia.

La opción *Búsqueda en Texto* implica una localización más exhaustiva que *Búsqueda en Materia*, mientras que ésta es aproximadamente 10 veces más rápida.

El campo *Período* se aplica sólo a los diarios de debates y permite agregar un filtro adicional para la búsqueda: la fecha de publicación del diario.

Resultados de búsqueda

Al ejecutar la búsqueda, aparece la pantalla de los resultados. Por ejemplo, al ejecutar la búsqueda por la palabra *derecho* en

la Constitución Política Mexicana con la opción de sinónimos, aparece la pantalla mostrada en la ilustración 12.

Cada elemento de la lista corresponde a un documento encontrado. En la lista se muestra el título del documento (su materia) —que en el caso de la Constitución es el artículo. También se proporcionan los datos formales del documento; en el caso de la Constitución nos referimos al número del artículo y a la sección en donde se encuentra; para los Diarios de debates, son la legislatura y la fecha de publicación del documento, etcétera.

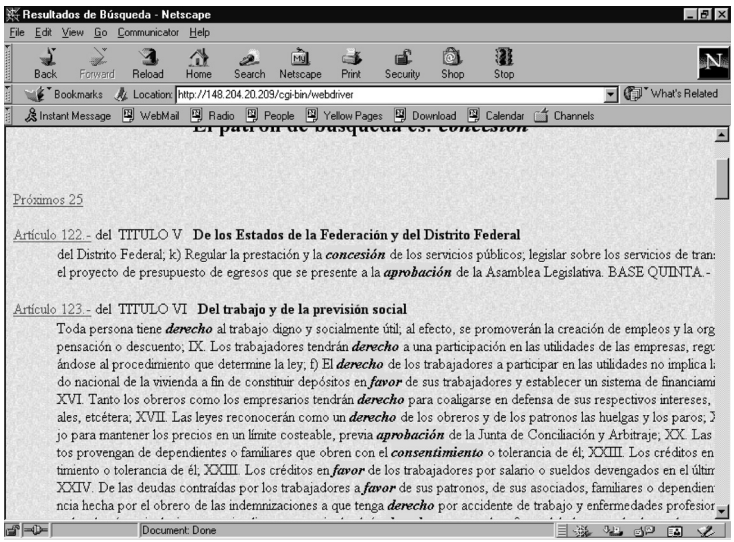


Ilustración 12. Resultados de búsqueda.

Las demás líneas de cada elemento muestran todas las ocurrencias de las palabras encontradas en el documento, en el contexto en el cual aparecen, con las palabras de búsqueda enfatizadas en negrillas (y también subrayadas en color amarillo). Esta parte de la información constituye un tipo de «resumen» del documento (relativo a la petición específica) que permite al usuario decidir rápidamente si el documento es relevante o si parece serlo, en todo caso, el usuario puede ver el texto completo.

En la ilustración 12 se puede apreciar que por la petición *derecho* se encontró no sólo la palabra exacta *derecho* sino también sus sinónimos *concesión*, *favor*, *consentimiento*, *aprobación*, que se hallan enfatizadas en negrillas en la línea respectiva del contexto.

Entonces, como ya mencionamos, si se elige la opción *Formas de palabras: s*e efectuará el enriquecimiento de la petición con las formas morfológicas de las palabras. Por ejemplo, si se busca por la petición «*pequeño*», se encontrarán los documentos que contienen las formas *pequeño, pequeña, pequeños, pequeñas*.

Si se elige la opción *Sinónimos***:** se efectuará el enriquecimiento de la petición con los sinónimos de las palabras de la petición. Por ejemplo, si se busca por la petición «*pequeño*», se encontrarán los documentos que contienen las formas *pequeño*, *chico*.

Si se elige la opción *Sinónimos más lejanos***:** se efectuará el enriquecimiento de la petición con los sinónimos cercanos y lejanos de las palabras de la petición. Por ejemplo, si se busca por la petición «*pequeño*», se encontrarán los documentos que contienen las formas *pequeño*, *chico, reducido*.

Si se elige la opción *Sinónimos y antónimos***:** se efectuará el enriquecimiento de la petición con los sinónimos cercanos y lejanos de las palabras de la petición, así como sus antónimos. Por ejemplo, si se busca por la petición «*pequeño*», se encontrarán los documentos que contienen las formas *pequeño*, *chico, reducido, grande*.

Si se elige la opción *Sinónimos y antónimos más lejanos***:** se efectuará el enriquecimiento de la petición con los sinónimos cercanos y lejanos de las palabras de la petición, así como sus antónimos cercanos y lejanos. Por ejemplo, si se busca por la petición «*pequeño*», se encontrarán los documentos que contienen las formas *pequeño*, *chico, reducido, grande, amplio* y sus formas morfológicas.

Veamos otro ejemplo de búsqueda. El modo de sistema es «búsqueda con sinónimos». La Ilustración 13 muestra los resultados de la búsqueda para la palabra «*coche*». También fue encontrada la palabra «*automóvil*».

9.4 Conclusiones

Se describió un método que permite mejorar las búsquedas en las bases de datos documentales: la búsqueda con comparación suave (no exacta) entre la petición y el texto del documento. Por ejemplo, para la petición *pensar* se encuentran todas sus formas gramaticales (*pienso*, *piensas*, etc.); para la petición *derecho* todos sus sinónimos (*concesión*, *favor*, *consentimiento*, *aprobación*, etc.). El método se basa en los diccionarios lingüísticos grandes con una estructura simple y permite al usuario ajustar las opciones al grado deseado de exactitud (o suavidad) de comparación.

Este método fue utilizado en el sistema desarrollado para la búsqueda documental en los textos políticos en la Base de datos del Senado de la República Mexicana.

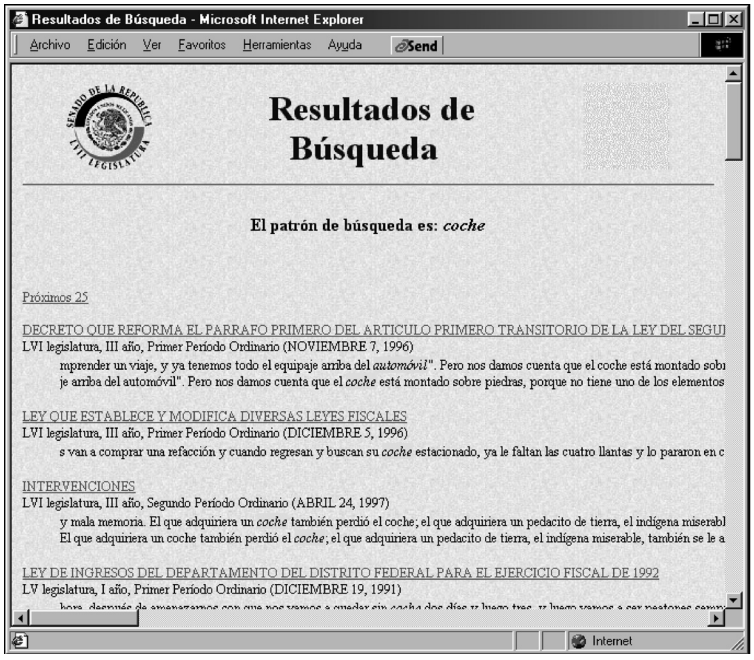


Ilustración 13. Resultados de búsqueda

Capítulo 10

COMPARACIÓN DE LOS COEFICIENTES DE LAS LEYES DE ZIPF Y HEAPS EN DIFERENTES IDIOMAS

Dos de las leyes estadísticas empíricas que rigen el comportamiento de las palabras en el texto, son las leyes de Zipf y Heaps (Manning y Schütze, 1999; Zipf, 1949). Se aplican normalmente a textos relativamente grandes, son especialmente notables en los corpus.

La ley de Zipf consiste en lo siguiente: se cuentan las frecuencias de las palabras (lemas[14] o formas de palabras) en el texto. A estas palabras se asocian sus rangos de acuerdo con sus frecuencias, empezando con los valores más altos. Estos rangos se ordenan desde los más grandes hasta los más pequeños. La ley de Zipf dice que en cualquier texto suficientemente grande, los rangos están en razón inversa con las frecuencias, es decir[15]:

---

[14] La forma de la palabra es una forma flexiva de la misma, tal cual se usa en el texto; lema es la forma normalizada de la palabra que usualmente aparece como la palabra encabezada en los diccionarios. Por ejemplo, para las formas de palabras: *pensará, pienso, pensándolo,* el lema es *pensar*.

[15] Ignoramos las correcciónes de Mandelbrot para la ley de Zipf (Manning and Shutze, 1999), pues sólo afectan los valores extremos de la distribución y no afectan los fenómenos que aquí discutimos.

$$f_r \approx C / r^2 \tag{1}$$

o bien, en la forma logarítmica:

$$\log f_r \approx C - z \log r \tag{2}$$

donde f_r es la frecuencia de la unidad (lema o forma de palabra) en el texto, con el rango r, z es el coeficiente exponencial (cerca de 1) y C es una constante. En la escala logarítmica, la gráfica de esta distribución es, aproximadamente, una línea recta que forma un ángulo de –45° con el eje de abscisas.

Otra ley estadística empírica que describe el comportamiento de las palabras en el texto es la de Heaps. Es mucho menos conocida que la de Zipf, pero no es menos importante.

La ley de Heaps dice que la cantidad de palabras (formas de palabra o lemas) diferentes en el texto está en razón directa con el exponente de su tamaño:

$$n_i \approx D \cdot i^h \tag{3}$$

o bien, en la forma logarítmica:

$$\log n_i \approx D + h \log \tag{4}$$

donde n_i es el número de palabras (formas de palabras o lemas) diferentes que ocurrieron antes de la palabra número i, h es el coeficiente exponencial (entre 0 y 1) y D es una constante. En la escala logarítmica la gráfica de esta distribución es aproximadamente una línea recta que forma un ángulo de 45° en relación con el eje de abscisas.

La naturaleza de las leyes de Zipf y Heaps no es clara. Es curioso que prácticamente cualquier texto obedezca semejantes leyes

empíricas. Desde el punto de vista lingüístico es interesante que las distribuciones de los lemas y de las formas de palabras sean muy parecidas, incluso para lenguajes con una morfología tan desarrollada como el ruso. En cuanto a la ley de Zipf, se sabe de otros fenómenos distintos, relacionados con la vida cotidiana, que obedecen esta ley, por ejemplo, el número de habitantes en las ciudades.

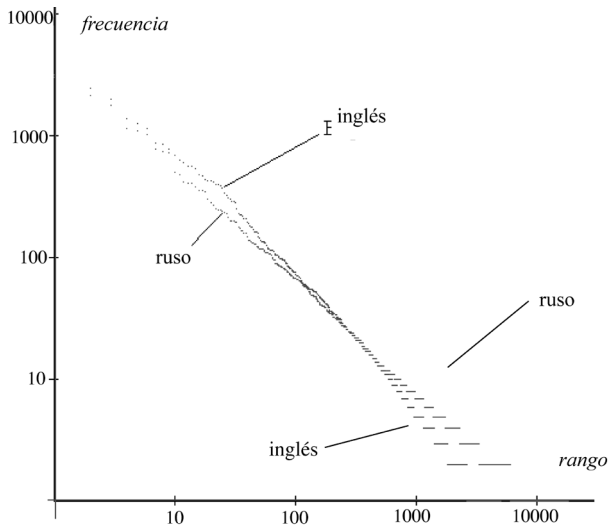


Ilustración 14. Las gráficas de la distribución de la ley de Zipf para el texto inglés *No.* 10 y el texto ruso *No.* 13 (escala logarítmica).

Puesto que estas leyes son atributos del lenguaje natural, se considera que se pueden usar para detectar la naturaleza lingüística de las señales desconocidas (Elliott, 2000).

En la práctica, es importante saber los valores de los coeficientes de estas leyes en diferentes idiomas (Gelbukh y Sidorov, 2001); por ejemplo, para el desarrollo de las bases de datos documentales que contienen textos completos de los documentos, pues esta información permite determinar el tamaño necesario del archivo de índice en función del tamaño de la base de textos.

En el presente capítulo se demuestra que el coeficiente *z* de la ley de Zipf y el coeficiente *h* de la de Heaps, dependen significativamente del lenguaje. En concreto se consideran los idiomas inglés y ruso. Los experimentos se realizaron usando un volumen de texto lo suficientemente grande y los textos que se utilizaron fueron de diferentes géneros de las bellas letras.

10.1 Resultados experimentales

Se procesaron 39 obras de las bellas letras en cada uno de los dos idiomas —inglés y ruso—. Los textos se seleccionaron de manera aleatoria con respecto a las obras disponibles de diferentes géneros. Una restricción adicional residió en la condición de que el tamaño del texto fuera no menor de 100 KB (no menos que 10,000 palabras); en total, no menos de 2.5 millones de palabras (24.8 MB) para el inglés y 2.0 millones de palabras (20.2 MB) para el ruso.

Hicimos nuestros experimentos tanto con los lemas como con las formas de las palabras. En los experimentos, las palabras se normalizaron automáticamente. No se usó ningún método de resolución de homonimia y todos los lemas se agregaron al archivo que contenía los resultados de normalización. Al final, el resultado mostró que los coeficientes de ambas leyes, tanto para lemas como para formas de palabras, son diferentes para diferentes idiomas.

Desarrollamos una herramienta que mostraba las gráficas de distribución según la ley de Zipf y la ley de Heaps. Para la ley de Zipf se usan los puntos:

$$x_r = \log r,\ y_r = \log f_r \tag{5}$$

y para la ley de Heaps los puntos:

$$x_i = \log i,\ y_i = \log n_i \tag{6}$$

Los ejemplos de las gráficas se presentan en la ilustración 15. En el eje *X* se dan los rangos y en el eje *Y* las frecuencias. Viendo las gráficas para los textos en diferentes idiomas son notables las diferencias, lo que se confirma con los datos calculados en los experimentos.

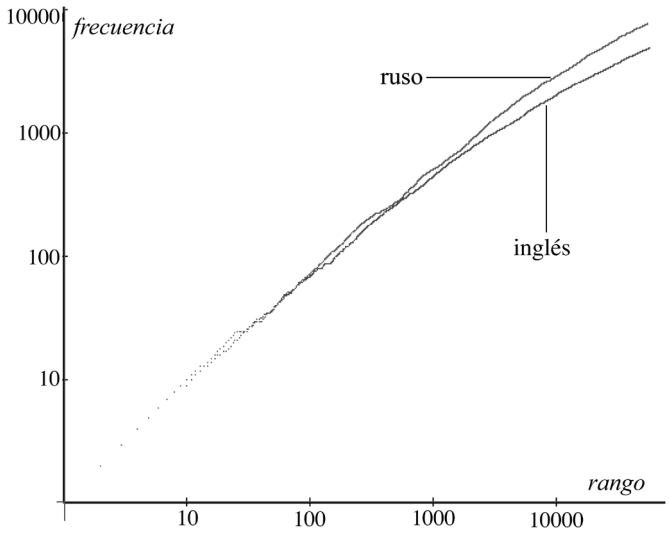


Ilustración 15. Ls gráficas de la distribución según la ley de Heaps para el texto en inglés *No*. 10, y el texto en ruso *No*. 20 (escala logarítmica).

Para los cálculos exactos se usó el método de regresión lineal. La gráfica se aproximaba a una línea recta: $y = ax + b$, donde a y b corresponden a los coeficientes z y C en la ley de Zipf y a h y D en la ley de Heaps. Debido al hecho de que la densidad de los puntos (x_i, y_i) se aumenta exponencialmente con el aumento de x_i estamos multiplicando todos los valores por c^{-x_i}, esto permite tomar en cuenta todas las partes de la gráfica para el cálculo de la distancia, donde c es la base de logaritmo, que se usa para calcular (x_i, y_i); en nuestro caso c es igual a 10. Para calcular a y b usamos las siguientes fórmulas:

$$b = \frac{\sum_i \frac{x_i}{c^{x^i}} \sum_i \frac{x_i y_i}{c^{x_i}} - \sum_i \frac{x_i^2}{c^{x_i}} \sum_i \frac{y_i}{c^{x_i}}}{\left(\sum_i \frac{x_i}{c^{x_i}}\right)^2 - \sum_i \frac{x_i^2}{c^{x_i}} \sum_i \frac{1}{c^{x_i}}}, \quad a = \frac{\sum_i \frac{y_i}{c^{x_i}} - b \sum_i \frac{1}{c^{x_i}}}{\sum_i \frac{x_i}{c^{x_i}}}, \qquad (7)$$

Las fórmulas modificadas dan mejores resultados de aproximación que la regresión lineal estándar. Los resultados de los cálculos se dan en las tablas correspondientes. Sólo proporcionamos los valores de z y *h*, porque los valores de C y *D* no tienen mayor importancia.

Para evaluar nuestros resultados usamos la regla común de 3σ, donde σ es desviación estándar.

Para el inglés $z = 0.97 \pm 0.06$ calculado en las formas de las palabras y $z = 0.98 \pm 0.07$ calculado en los lemas. Para el ruso $z = 0.89 \pm 0.07$ calculado en las formas de las palabras y $z = 0.91 \pm 0.09$ calculado en los lemas. La diferencia entre el inglés y el ruso para la ley de Zipf es 8.3% en formas de palabras y 9.9% en las de los lemas.

Para el inglés $h = 0.79 \pm 0.05$ calculado en los lemas y $h = 0.80 \pm 0.05$ calculado en las formas de las palabras. Para el ruso $h = 0.84 \pm 0.06$ calculado en los lemas y $h = 0.89 \pm 0.05$ calculado en las formas de las palabras. La diferencia entre el inglés y el ruso para la ley de Heaps es 5.6% en formas de las palabras y 5.9% en los lemas.

Tanto para la ley de Zipf, como para la ley de Heaps la diferencia es significativa, es decir, mayor que 1% comúnmente aceptado.

10.2 La posible explicación de la diferencia

La primera idea que viene a la mente es tratar de buscar la posible explicación del fenómeno de incompatibilidad entre los

coeficientes en las diferencias gramaticales entre los idiomas. El ruso tiene un sistema morfológico mucho más complejo que el inglés —es un idioma sintético mientras que el inglés es analítico. Otra consideración es que un texto en ruso debe tener mucho más formas gramaticales que un texto en inglés. Sin embargo, esta idea parece incorrecta porque los coeficientes para lemas y para formas de palabras en ruso son muy cercanos.

Una posible explicación, que nos parece probable, es tratar de relacionar esas diferencias con el concepto de la «riqueza léxica» de los idiomas. Es bien conocido que a veces el idioma «obliga» al hablante a expresar algún valor semántico que es irrelevante en una determinada situación solamente porque tiene palabras demasiado «especializadas». Por ejemplo, en inglés se dice *The table was near the wall*, literalmente *La mesa está cerca de la ventana.* Mientras que en ruso se necesita decir literalmente *La mesa está parada cerca de la ventana*, es decir, el ruso obliga a hacer la elección de varias palabras con diferentes modos de presencia (*yacer, estar sentado, estar parado*, etc.). Es decir, en dado caso, el número de palabras que se puede elegir es mayor en ruso. Sin embargo, esta hipótesis necesita más investigaciones.

10.3 Conclusiones

Hemos demostrado que los coeficientes exponenciales de las leyes de Zipf y Heaps dependen sustancialmente de los idiomas. Llegamos a esta conclusión a partir de comparar los coeficientes calculados para los 39 diferentes textos de tamaño considerable y de diferentes géneros de ficción en ruso y en inglés. El tamaño de los textos es comparable para los diferentes idiomas. Para calcular los coeficientes se usó el método de regresión lineal con una normalización adicional.

En el futuro planeamos seguir investigando el concepto de la «riqueza léxica», por ejemplo, haciendo comparaciones entre el

original y su traducción. También parece interesante calcular los coeficientes tomando en cuenta las partes de la oración. Además, nos gustaría calcular los coeficientes para otros idiomas, sin embargo, es difícil, debido a la ausencia de grandes conjuntos de textos disponibles.

10.4 Apéndice 1: valores de los coeficientes de las leyes de Zipf y Heaps

En las tablas 1 y 2 se presentan los valores de los coeficientes de las leyes de Zipf y Heaps para los idiomas inglés y ruso. El número del texto en las tablas corresponde al número del texto en las tablas 3 y 4, donde se presenta la información sobre el texto. Los datos de las tablas están ordenados por el valor de coeficiente de Zipf para las formas de las palabras.

Tabla 1. Los valores para el inglés.

Texto	Género	Zipf (formas)	Zipf (lemas)	Heaps (lemas)	Heaps (formas)
1	novela policíaca	1.037639	1.034344	0.759330	0.773328
2	aventuras	1.004620	0.998473	0.788285	0.802263
3	novela	0.999033	0.991512	0.794793	0.808854
4	novela	0.996945	0.987663	0.777628	0.790161
5	novela policíaca	0.991697	0.973819	0.793684	0.802822
6	novela policíaca	0.991656	0.986506	0.784293	0.794182
7	aventuras	0.991037	0.979161	0.795032	0.805405
8	novela	0.988051	0.988768	0.801261	0.811563
9	ciencia ficción	0.984583	0.979749	0.790036	0.803747
10	ciencia ficción	0.984467	0.972981	0.798092	0.807740
11	ciencia novela	0.983066	0.994065	0.800523	0.812804

12	ciencia ficción	0.982076	0.983231	0.810374	0.821457
13	novela policíaca	0.982069	0.954409	0.804559	0.812377
14	novela policíaca	0.981934	0.972231	0.806420	0.816998
15	novela	0.978492	0.968451	0.815062	0.825980
16	novela	0.978363	0.966682	0.798223	0.807001
17	novela policíaca	0.978101	0.967885	0.809228	0.819173
18	infantil	0.976800	1.012762	0.742432	0.756829
19	ciencia ficción	0.976773	0.966636	0.784674	0.796484
20	aventuras	0.971846	0.961520	0.823809	0.831446
21	novela	0.971531	0.958468	0.806512	0.815702
22	aventuras	0.971082	0.989721	0.792677	0.802851
23	novela	0.970900	0.962113	0.794577	0.804060
24	novela	0.968299	0.993697	0.803362	0.815941
25	infantil	0.968028	0.959380	0.777983	0.793339
26	novela	0.967511	0.974234	0.754915	0.767074
27	novela	0.966305	1.001287	0.778061	0.790588
28	ciencia ficción	0.965116	0.950745	0.794937	0.804610
29	ciencia ficción	0.961867	0.949584	0.813870	0.825393
30	novela	0.961286	0.952750	0.799193	0.809003
31	ciencia ficción	0.955980	0.945660	0.803026	0.810366
32	ciencia ficción	0.955516	0.940502	0.809863	0.820718
33	novela	0.954731	1.026885	0.741586	0.753864
34	novela	0.952700	0.991605	0.795840	0.811328
35	ciencia ficción	0.952088	0.941467	0.780060	0.788162
36	infantil	0.950748	0.972238	0.771153	0.781493
37	novela policíaca	0.948861	0.967911	0.792331	0.801062
38	ciencia ficción	0.948237	0.945391	0.801813	0.814089
39	novela	0.930612	0.972905	0.816378	0.824606
Promedio:		0.973863	0.975318	0.792458	0.803458
$3 \times \sigma$:		0.057036	0.065021	0.055954	0.053281

σ es desviación estándar

Tabla 2. Los valores para el ruso.

Texto	Género	Zipf (formas)	Zipf (lemas)	Heaps (lemas)	Heaps (formas)
1	infantil	0.936576	0.964813	0.787141	0.841100
2	novela	0.935878	0.964046	0.825040	0.871886
3	novela	0.929603	0.955567	0.839364	0.889200
4	novela policíaca	0.928132	0.939130	0.839518	0.886388
5	novela policíaca	0.924204	0.944139	0.858930	0.894042
6	novela policíaca	0.917411	0.942821	0.822190	0.873935
7	aventuras	0.916674	0.960386	0.793264	0.855948
8	novela	0.912970	0.931723	0.842878	0.885869
9	novela	0.912406	0.940216	0.822597	0.871927
10	novela policíaca	0.909435	0.927857	0.839980	0.889580
11	novela	0.908496	0.963706	0.814065	0.864963
12	novela	0.906881	0.922668	0.838711	0.886990
13	ciencia ficción	0.903534	0.919563	0.816362	0.868314
14	novela	0.902698	0.927154	0.846717	0.894226
15	ciencia ficción	0.902272	0.915499	0.842399	0.885195
16	infantil	0.901783	0.916074	0.844565	0.886987
17	ciencia ficción	0.899720	0.911501	0.821493	0.871524
18	ciencia ficción	0.892304	0.907987	0.853072	0.896268
19	novela	0.890569	0.946387	0.846493	0.891929
20	novela	0.890088	0.902435	0.859763	0.900825
21	novela policíaca	0.887773	0.909617	0.838548	0.889677
22	novela	0.886602	0.898627	0.856025	0.897606
23	novela	0.884160	0.963282	0.818838	0.864900
24	novela	0.883826	0.896010	0.832264	0.885477
25	novela policíaca	0.883621	0.880983	0.872263	0.910767
26	infantil	0.883044	0.885564	0.856513	0.895081
27	ciencia ficción	0.881713	0.889017	0.848118	0.891209

28	aventuras	0.880597	0.899939	0.834420	0.882924
29	novela	0.879422	0.887770	0.873361	0.905620
30	ciencia ficción	0.876683	0.885460	0.858251	0.899792
31	novela	0.874849	0.888930	0.852379	0.897232
32	novela policíaca	0.873471	0.907970	0.830596	0.882299
33	novela policíaca	0.870795	0.863837	0.876895	0.915232
34	novela	0.867954	0.885425	0.871117	0.907745
35	ciencia ficción	0.867008	0.870758	0.870979	0.903001
36	ciencia ficción	0.863004	0.879573	0.841957	0.884644
37	aventuras	0.859045	0.894258	0.834773	0.885242
38	novela policíaca	0.857402	0.871889	0.850555	0.896164
39	ciencia ficción	0.839270	0.840562	0.881458	0.912924
Promedio:		0.892869	0.912901	0.842406	0.887555
$3 \times \sigma$:		0.068292	0.094028	0.063054	0.046417
σ es desviación estándar					

10.5 Apéndice 2: listas de textos utilizados en los experimentos

En nuestros experimentos utilizamos los textos enlistados en las tablas 3 y 4. El número del texto en las tablas 1 y 2 corresponde al número en las tablas que se presentan a continuación. Los conjuntos de textos para los lenguajes ruso e inglés son aproximadamente equivalentes en cuanto al estilo, género, tamaño, etc. Puesto que la comprensión del significado de los títulos de los textos no es importante para el lector y, además, en la mayoría de los casos no se tiene la traducción canónica del título específico, presentamos en la tabla los títulos en sus respectivos idiomas sin traducción.

Tabla 3. Textos en inglés

Texto	Autor	Título	Género
1	A. Conan Doyle	*Novels and Stories*	novela policíaca
2	W. Scott	*Ivanhoe*	aventuras
3	H. Melville	*Moby Dick*	novela
4	H. Beecher Stowe	*Uncle Tom's Cabin*	novela
5	A. Conan Doyle	*The Case Book of Sherlock Holmes*	novela policíaca
6	A. Conan Doyle	*The Memoirs of Sherlock Holmes*	novela policíaca
7	E. R. Burroughs	*Tarzan of The Apes*	aventuras
8	T. Hardy	*Far from the Madding Crowd*	novela
9	W. Schwartau	*Terminal Compromise*	ciencia ficción
10	A. Hope	*The Prisoner of Zenda*	ciencia ficción
11	M. Twain	*Life on the Mississippi*	ciencia novela
12	J. Verne	*From the Earth to the Moon*	ciencia ficción
13	A. Conan Doyle	*His Last Bow*	novela policíaca
14	G. K. Chesterton	*The Innocence of Father Brown*	novela policíaca
15	N. Hawthorne	*The Scarlet Letter*	novela
16	M. Twain	*The Adventures of Tom Sawyer*	novela
17	G. K. Chesterton	*The Wisdom of Father Brown*	novela policíaca
18	Gene Stratton-Porter	*Laddie. A True Blue Story*	infantil
19	R. J. Denissen	*The Europa Affair*	ciencia ficción
20	A. Bierce	*Can Such Things Be*	aventuras
21	J. Verne	*Around the World in Eighty Days*	novela

22	E. R. Burroughs	*The Mucker*	aventuras
23	A. Conan Doyle	*Valley of Fear*	novela
24	W. Scott	*Chronicles of the Canongate*	novela
25	R. Kipling	*The Jungle Book*	infantil
26	J. Austin	*Pride and Prejudice*	novela
27	D. H. Lawrence	*Sons and Lovers*	novela
28	D. K. Bell	*Jason the Rescuer*	ciencia ficción
29	W. Gibson	*Neuromancer*	ciencia ficción
30	Baroness Orczy	*The Scarlet Pimpernel*	novela
31	D. Adams	*The Restaurant at the End of the Universe;*	ciencia ficción
32	D. K. Bell	*Van Gogh in Space*	ciencia ficción
33	M. Twain	*The Adventures of Huckleberry Finn*	novela
34	H. D. Thoreau	*Walden & on The Duty of Civil Disobedience*	novela
35	L. Dworin	*Revolt of the Cyberslaves*	ciencia ficción
36	L. Maud Montgomery	*Anne of Green Gables*	infantil
37	A. Conan Doyle	*Hound of Baskervilles*	novela policíaca
38	B. Sterling	*The Hacker Crackdown*	ciencia ficción
39	N. Hawthorne	*The House of the Seven Gables.*	novela

Tabla 4. Textos en ruso*.

Texto	Autor	Título	Género
1	Н. Носов	Приключения Незнайки	infantil
2	В. Аксенов	Сборник	novela
3	А.Солженицын	Архипелаг ГУЛаг	novela
4	А. Степанов	День гнева	novela policíaca
5	В. Федоров и др.	Бенефис двойников	novela policíaca
6	Ю. Семенов	Семнадцать мгновений весны	novela policíaca
7	Г. Хаггард	Дочь Монтесумы	aventuras
8	Вл. Кунин	Повести	novela
9	А. Покровский	"...Расстрелять"	novela
10	М. Наумова	Констрикторы	novela policíaca
11	Ф. Достоевский	Неточка Незванова	novela
12		Азюль	novela
13	В. Пелевин	Сборник рассказов и повестей	ciencia ficción
14	М. Горький	Автобиографические рассказы	novela
15	С. Михайлов	Шестое чувство	ciencia ficción
16	Л. Лагин	Старик Хоттабыч	infantil
17	Д. Громов	Сборник рассказов и повестей	ciencia ficción
18	В. Рыбаков	Рассказы	ciencia ficción
19	Е. Козловский	Киносценарии и повести	novela
20	А. Мелихов	Во имя четыреста первого, или Исповедь еврея	novela
21	А. Курков	Выстрел	novela policíaca
22	В. Иванов	Голубые пески	novela
23	М. Мишин	Почувствуйте разницу	novela
24	А. Платонов	Котлован	novela

25	В. Черняк	Выездной!	novela policíaca
26	А. Некрасов	Приключения капитана Врунгеля	infantil
27	И. Федоров	Рассказы	ciencia ficción
28	У. Комм	Фрегаты идут на абордаж	aventuras
28	Н. Галкина	Ночные любимцы	novela
30	Б. Иванов, Ю. Щербатых	Случай контрабанды	ciencia ficción
31	В. Набоков	Рассказы	novela
32	В. Суворов	Аквариум	novela policíaca
33	В. Черняк	Жулье	novela policíaca
34	С. Дышев	До встречи в раю	novela
35	Н. Перумов	Рассказы, Русский меч	ciencia ficción
36	А. Первушин	Рассказы	ciencia ficción
37	Т. Майн Рид	Американские партизаны	aventuras
38	М. Болтунов	“Альфа” - сверхсекретный отряд...	novela policíaca
39	В. Бабенко	Игоряша «Золотая рыбка».	ciencia ficción

* Tanto los autores como los títulos de sus trabajos, en su mayoría, no son conocidos fuera de Rusia. La información tiene un carácter ilustrativo y no afecta los resultados de la investigación; por eso se dan sin traducción.

Parte III

Construcción de recursos léxicos para el PLN

Capítulo 11

COMPILACIÓN AUTOMÁTICA DEL CORPUS LÉXICA Y MORFOLÓGICAMENTE REPRESENTATIVO

Una de las aplicaciones principales de los corpus es proporcionar un conocimiento léxico de diferentes tipos y niveles (Biber *et al*., 1998; McEnery y Wilson, 1996), usualmente, de las probabilidades empíricas de diferentes propiedades o situaciones en el texto. En este sentido, hay dos tipos de probabilidades:

- Probabilidades absolutas: ¿cuál es la probabilidad de la situación *X* en el texto? Por ejemplo, ¿cuál es la frecuencia relativa de la palabra *ecuación* en el texto?
- Probabilidades condicionales: ¿cuál es la probabilidad de la situación *X* en la presencia de la situación *Y*? Es decir, la investigación estadística se aplica a un subcorpus que consiste en los contextos que contienen la situación *Y*. Por ejemplo, ¿cuál es la probabilidad de la palabra *ecuación* asumiendo que la siguiente palabra es *diferencial*?

Aunque la primera tarea parece interesante, su importancia práctica se limita a los casos de análisis de las estructuras del texto consideradas completamente fuera del contexto, lo que prácticamente no sucede.

A diferencia de la primera, la segunda tarea tiene un sinnúmero de aplicaciones. Por ejemplo:

- Aprendizaje automático de los marcos de subcategorización: ¿qué preposiciones se usan con la palabra *empezar*? (Galicia-Haro *et al*., 2001).
- Aprendizaje de las combinaciones de palabras: con la palabra *atención*, se usan las palabras *prestar*, *atraer*, *perder* (Benson *et al*., 1986; Bolshakov y Gelbukh, 2000; Bolshakov 1994).
- Aprendizaje de las propiedades sintácticas: ¿la palabra *gran* se usa antes o después de la palabra que modifica?
- Detección de los llamados malapropismos (las palabras mal escritas que se parecen a palabras existentes de la misma categoría gramatical): en la presencia de las palabras *novia*, *iglesia* y *ceremonia*, ¿es más probable que aparezca *casar* o *cazar*?[16]

Sin embargo, los corpus tradicionales —grandes colecciones de textos seleccionados aleatoriamente de entre textos de cierto género, tema, etc., véase (Biber, 1993)— son más apropiados para la primera tarea, más específicos para determinar si la frecuencia de una cierta palabra (construcción, etc.) es alta o baja.

En cualquier corpus de este tipo, unas pocas palabras ocurren muchísimas veces, ocupando la mayor parte de su volumen y tomando la mayor parte del tiempo del procesamiento. Por otro lado, la inmensa mayoría de las palabras del lenguaje tiene muy poca o ninguna representación en el corpus. Este fenómeno se conoce como la ley de Zipf: la palabra con el rango estadístico n tiene aproximadamente una frecuencia C / n, donde C es una constante (véase el capítulo 10).

Esto no presenta mayor problema para la tarea de aprendizaje de las probabilidades absolutas (primera tarea). Sin embargo, la

[16] En todos los dialectos del español, salvo en el castellano, no hay diferencia en la pronunciación de [s] y [z].

utilidad de corpus de tamaño razonable para el aprendizaje de las probabilidades condicionales (segunda tarea) es limitada, debido a que para casi todas las palabras del lenguaje el número de ocurrencias es estadísticamente insuficiente para hallar resultados confiables. Por otro lado, la inmensa mayoría de ocurrencias de las palabras en el corpus son repeticiones de las mismas palabras, redundantes desde el punto de vista estadístico: bastaría un número mucho menor de ocurrencias de cualquiera de ellas.

Una solución a este problema es el uso del corpus más grande que se ha creado por la humanidad: la Internet. En un corpus tan enorme, hay información suficiente para aprender las propiedades de un gran número de palabras.

Los sistemas que utilizan Internet como corpus se conocen como corpus virtuales (Kilgariff, 2001) y se utilizan de la siguiente manera: el usuario presenta su petición especificando la palabra en cuestión (*Y* en los términos de la segunda tarea descrita al inicio de esta sección), el programa busca en Internet un cierto número de contextos relevantes (no todos, ya que pueden ser demasiados) y presenta al usuario, para su investigación estadística, un subcorpus construido de esta manera. Un ejemplo de esta herramienta es, digamos, WebCorp[17].

A pesar de la indudable utilidad de los corpus virtuales, los corpus «reales» —en forma de archivo— tienen ventajas importantes sobre los corpus virtuales:

- Respuesta rápida que no depende del tráfico de la red.
- Uso local de los recursos. No sobrecarga la red.
- Posibilidad de revisión, control de calidad, limpieza manual, marcaje manual (digamos, marcaje sintáctico o morfológico) y toda clase de preparativos que distinguen un corpus de una colección de textos no preparados.

[17] www.webcorp.org.uk

– Resultados estables y reproducibles en tiempo y espacio. A la misma petición hecha por diferentes usuarios en diferentes días o años se presenta la misma respuesta. Esto facilita la utilización del corpus para la comparación de diferentes métodos y sistemas, para el trabajo paralelo de diferentes grupos de desarrollo o pruebas, etc. Internet, al contrario, cambia constantemente: cada segundo aparecen nuevos sitios y páginas, se encienden o se apagan los servidores, las máquinas de búsqueda indexan nuevas páginas y quitan otras, etc.

Para combinar las ventajas de los corpus virtuales y los almacenados localmente, nosotros proponemos la construcción, a través de Internet, de un gran diccionario de contextos de palabras.

Los diccionarios de contextos se conocen como las concordancias de tipo KWIC (*key words in context*). En nuestro caso, el diccionario en este formato se compila a través de la Internet, lo que permite hacerlo muy grande (dar información para un gran número de palabras) y emplearlo para las tareas para las que normalmente se emplean los corpus.

Específicamente, se trata de ejecutar peticiones a Internet sobre cada palabra, guardando la respuesta. La recuperación posterior de estas respuestas simula el resultado de la petición a un corpus virtual con toda su riqueza léxica, pero sin necesidad de conectarse a la red y con todas las ventajas descritas arriba (la posibilidad de control y marcaje manual, resultados estables, etcétera).

Es decir, nuestro sistema es un corpus virtual (ya que simula su comportamiento) sin serlo físicamente (sin necesidad de conectarse a la red). Gracias al número limitado de los contextos que se obtienen y se guardan para cada palabra, no es necesario guardar toda la Internet en el disco local, sino un archivo relativamente pequeño que cabe en un CD-ROM.

Sin embargo, lo llamamos diccionario porque es estructurado: proporciona los datos para cada palabra encabezado.

Obviamente, el costo de estas comodidades es la pérdida de la flexibilidad de los corpus virtuales reales. Los parámetros del corpus (tales como el umbral del número de contextos para cada palabra) se deben determinar de antemano. El tipo de petición (contextos de una palabra versus contextos que contienen dos palabras específicas, etc.) también se fija en el tiempo de compilación. Para cambiar los parámetros sólo se tiene que repetir el proceso automático de compilación del corpus.

En el resto del capítulo se presenta con mayor detalle el diccionario compilado, se describe el método que hemos empleado para su compilación y los parámetros variables, los resultados experimentales y el trabajo futuro.

11.1 El diccionario de contextos

El diccionario presenta, para cada palabra encabezado, un cierto número (N) de contextos de uso. Los contextos deben ser lingüísticamente válidos, es decir, ser oraciones completas (o unas ventanas de texto) en español que son representativas del uso de la palabra en el lenguaje.

El diccionario de este tipo también se puede llamar un corpus representativo (Gelbukh *et al*., 2002a), ya que cada palabra para la que hacemos una búsqueda tiene en él representación estadísticamente significativa.

El número N se determina con las fórmulas correspondientes de la estadística matemática, basándose en el porcentaje requerido de confiabilidad de los resultados de la investigación estadística. O bien, de manera sencilla, se puede determinar como suficiente basándose en la experiencia del investigador. En nuestros experimentos usamos $N = 50$.

Para algunas palabras no se puede obtener de la Internet el número requerido N de contextos válidos. Estas palabras, en

el sentido estricto, no pertenecen al vocabulario del diccionario (porque no cumplen los requisitos) pero sus contextos se guardan y se presentan al usuario con la advertencia de que la información puede ser estadísticamente menos confiable.

Las ventajas de nuestro diccionario (un corpus virtual, virtual) en comparación con el corpus tradicional (un texto muy largo) son:

- Este corpus ocupa mucho menos espacio y es más fácil de manejar porque no contiene un número inmenso de palabras frecuentes.
- Además, el corpus contiene contextos de las palabras más raras, de las que el corpus tradicional contiene pocas o ninguna ocurrencias.

Por lo tanto, con este diccionario se pueden obtener las estadísticas de uso de las palabras de baja frecuencia, las cuales, de hecho, son la mayoría de las palabras del lenguaje.

Las ventajas de nuestro diccionario, en comparación con un corpus virtual, se han descrito en la sección anterior.

Los corpus virtuales también tienen sus problemas. Uno de éstos es la baja calidad de los contextos encontrados:

- La palabra puede usarse como nombre propio de una persona, empresa, producto, etc. En algunos casos los contextos no son lingüísticos, sino son rótulos de imágenes, direcciones de Internet, nombres de campos de formularios, archivos, tablas, etcétera.
- En otros casos el contexto no es de uso de la palabra, sino, por ejemplo, un artículo del diccionario que la explica o traduce.
- Finalmente, el lenguaje usado por los autores en Internet puede ser poco culto o gramáticamente mal formado.

En cuanto al último problema no se puede hacer mucho. Además, no resulta claro si es un problema del corpus o una propiedad que refleja el uso real (a diferencia del uso prescrito) del lenguaje.

En cuanto a los demás problemas, los hemos solucionado hasta cierto grado empleando filtros que detectan esos defectos en un contexto dado y previenen que el contexto entre en el diccionario.

Otra complicación es el manejo de las formas gramaticales de las palabras. Obviamente, la búsqueda del lexema se efectúa en los textos y se debe representar en los contextos del diccionario como una forma específica de la palabra (*pienso*, *pensaríamos* versus *pensar*). Hay tres estrategias posibles para calcular el porcentaje deseado de las formas correspondientes en el diccionario. Se puede representar las formas así:

- Uniformemente: si la palabra pensar tiene $K = 65$ formas, buscar N / K contextos para *pensar*, N / K para *pienso*, N / K para *pensaríamos*, etc. Con esta estrategia, cada lexema se representa con N contextos.
- Independientemente: N contextos para *pensar*, N para *pienso*, N para *pensaríamos*, etc. Con esta estrategia, cada lexema se representa con $K \times N$ contextos, es decir, los lexemas de verbos serán mejor representados que los de sustantivos.
- Proporcionalmente en relación con su uso en los textos: los N contextos para el lexema se dividen de tal manera que a las formas más usadas les corresponde un mayor número de contextos.

Las diferentes estrategias corresponden a diferentes tipos de aplicación del diccionario. Ya que nuestra motivación fue el estudio de las propiedades combinatorias de las palabras que dependen poco de las formas gramaticales, en nuestros experimentos empleamos la última estrategia.

11.2 Compilación del diccionario a través de la Internet

Para la compilación del diccionario se ejecutaron los siguientes pasos (todos salvo el primero son completamente automáticos):

1. Se compiló la lista de las palabras que debieron ser representadas en el diccionario.
2. Se generaron todas las formas gramáticas (morfológicas) de estas palabras.
3. Para cada palabra se calculó la proporción (porcentaje) de sus diferentes formas gramáticas, en la que deben ser representadas en el diccionario. Esta proporción corresponde al uso de las formas en Internet. Por ejemplo, si la forma singular de la palabra ocurre en 400 mil documentos en Internet y la forma plural en 100 mil, entonces en sus $N = 50$ contextos en el diccionario va a ocurrir 40 veces en singular y 10 en plural. Para los verbos, algunas formas (menos usadas para el verbo específico) no obtuvieron ninguna representación en el diccionario.
4. Para cada forma morfológica de la palabra (digamos, singular y plural), se buscaron en Internet los documentos que la contienen, haciendo una petición a un buscador de Internet (se experimentó con *Google* y *Altavista*) y analizando su respuesta con el fin de obtener el URL del documento.
5. Para cada URL obtenido, se bajó el documento, se analizó su estructura HTML y se extrajeron los contextos de la palabra en cuestión.
6. Para cada contexto, se aplicaron los filtros heurísticos con el fin de rechazar los contextos no válidos (más cortos que $n = 8$ palabras sin traspasar las marcas HTML, o donde la palabra resulta un nombre propio, una dirección de Internet, etcétera).

El paso 5 se repitió hasta encontrar, de ser posible, el número requerido (determinado en el paso 3) de los contextos válidos de la forma morfológica dada de la palabra.

Para el paso 2 se empleó un sistema de análisis y generación morfológico descrito en el capítulo 5. Si para el paso 1 se usara un corpus, las palabras se lematizarían empleando el mismo sistema. Sin embargo, nosotros utilizamos la lista de palabras que ya estuvieron en la primera forma gramatical (singular, infinitivo, etc.) extraída de un diccionario.

En el paso 3, para cada forma morfológica de la palabra se ejecutaba la misma petición a las máquinas de búsqueda que en el paso 4, pero sólo se analizaba el número total de documentos que contenían esa palabra (los buscadores proporcionan este número).

Empezando con el paso 2, el programa funciona de manera totalmente automática, sin intervención humana alguna.

Con la configuración del buscador usado (los buscadores de Internet usualmente permiten especificar el lenguaje de los documentos a buscar), se asegura que los textos serán en español.

En el paso 6 se requirió un tamaño definido de contexto porque los contextos cortos no contienen suficiente información lingüística. Además, frecuentemente no son expresiones de lenguaje natural sino otro tipo de datos (rótulos de imágenes, nombres de campos, botones o archivos, etcétera).

Los experimentos también han mostrado que una palabra puede tener muy alta frecuencia pero como apellido y no como palabra normal. Por ejemplo, muchos contextos encontrados para la palabra *abad* referían al apellido. Por eso no se consideraron válidos aquellos contextos en que la palabra en cuestión empieza con mayúscula en medio de la oración.

Nótese que con rechazar algún contexto no se pierde mucho, ya que usualmente el número de contextos disponibles es muy grande, a reserva de que no se rechacen de modo tan sistemático que afecten sus propiedades estadísticas.

Obviamente, uno de los filtros verifica que los contextos no se repitan literalmente (lo cual significaría que se trata de copias múltiples del mismo documento).

11.3 Resultados experimentales

El programa que aplica el algoritmo descrito se realizó de tal manera que cumplió con los siguientes parámetros:

- Los umbrales numéricos tales como N y n.
- El conjunto de los filtros a aplicar.
- El nombre del archivo que contiene la lista de palabras encabezado.
- El lenguaje para el cual se compiló el diccionario.

Todos se especifican a través de la interfaz del usuario. Una vez especificados los parámetros, el programa funciona en el modo automático hasta que se genera el diccionario indispensable.

En la lista de las palabras encabezado usamos el vocabulario del diccionario *Anaya* de la lengua española, que contiene 33 mil de las palabras más comunes del español. Como estas palabras ya están en la forma morfológica principal, no usamos el lematizador en este paso. Obviamente, la lista de palabras se puede ampliar sin necesidad de repetir el proceso para las palabras actuales.

Para estas 33 mil palabras la ejecución automática del programa tomó alrededor de tres semanas, llevado a cabo en una computadora PC Pentium IV conectada a la red de área local de velocidad mediana y variable según el tráfico. Durante estas tres semanas, para el total de 100 mil formas morfológicas de las palabras de la lista, se ejecutaron alrededor de 200 mil peticiones (pasos 3 y 4 del algoritmo) al buscador de Internet y se

descargaron alrededor de 33 × N = 1650 mil documentos. Para la compilación del diccionario se usó el buscador *Google*, ya que su tiempo de respuesta es muy bueno.

El archivo de resultado tiene un tamaño de 221 MB, es decir, una tercera parte de un disco compacto.

Para la mayoría de las palabras de la lista inicial, el diccionario contiene los 50 contextos esperados. Sin embargo, para 10% de las palabras el número de contextos válidos encontrados es menor. En el momento actual no es claro si esto se debe a la baja frecuencia de su uso en Internet, a algún problema técnico del programa, o a la implementación de los filtros.

11.4 Conclusiones

El diccionario propuesto tiene las ventajas de los corpus virtuales:

- Menor número de palabras redundantes.
- El número suficiente de contextos de la mayoría de las palabras para un aprendizaje estadísticamente confiable.

así como las de los corpus tradicionales:

- Tamaño razonablemente pequeño.
- Manejo local de los recursos con respuesta rápida y sin sobrecarga de la red.
- Posibilidad de clasificación, limpieza y marcaje manual.
- Estabilidad y reproducibilidad de los resultados en el tiempo y espacio.

También hereda algunas desventajas de los corpus virtuales:

- Calidad inferior de los textos debido al lenguaje cotidiano de Internet (depende mucho de la calidad de los filtros empleados).

– Imposibilidad de elegir el género, tópico, autor, etcétera.
– Imposibilidad (sin aplicar las herramientas correspondientes) de resolver la homonimia de algunas partes de la oración (¿*trabajo* es verbo o sustantivo?), lo que aumenta el ruido en el corpus obtenido; y de distinguir los sentidos de palabras diferentes, lo que hace que un sentido frecuente sea representado de forma mucho más amplia que los demás.

así como las de los corpus tradicionales:

– Menor flexibilidad en las peticiones.
– Parámetros del corpus fijados en el tiempo de compilación.

El primer grupo de restricciones es inherente a los corpus basados en Internet y puede limitar su uso. El último grupo no presenta mayor problema, ya que con el programa que hemos desarrollado, la obtención de un corpus con parámetros distintos es cuestión de unos clics (y varias semanas de trabajo automático de la computadora).

Basados en esta idea realizamos un programa para compilar automáticamente tales diccionarios. Con él recopilamos un diccionario que, para unas 33 mil palabras en español, da alrededor de 50 contextos por palabra, divididos entre 100 mil formas gramaticales ponderadas según sus frecuencias de uso total. Lo que resultó en un archivo de tan sólo 221 MB —lo que cabe a un CD.

El diccionario obtenido se usará para el aprendizaje automático de los diccionarios estadísticos de diferentes tipos para el español, como el diccionario de marcos de subcategorización y el diccionario de combinaciones de palabras (atracción léxica o colocaciones). Este último será la base para la compilación manual del diccionario de las funciones léxicas del español.

Se pueden mencionar las siguientes direcciones del trabajo futuro:

- Mejorar los filtros heurísticos para los contextos.
- Experimentar con diferentes valores de *N* y *n*.
- Experimentar con diferentes maneras de ponderación de las formas gramaticales.
- Probar otros tipos de peticiones. Por ejemplo, compilar un diccionario que para cada par de palabras de cierta lista (Bolshakov y Gelbukh, 2000) dé ejemplos de sus coocurrencias. También, un diccionario donde la unidad de la petición no sea una palabra, sino una cierta combinación de características gramaticales (por ejemplo, pretérito plural de segunda persona): encontrar *N* contextos para cada combinación de ciertas características. El algoritmo correspondiente es obvio y la modificación al programa existente será mínima.
- Tratar de desarrollar filtros temáticos o de género, con el fin de compilar un diccionario para un cierto género o tema específico. El algoritmo estará basado en un tesauro (Gelbukh *et al*., 1999) y en enriquecimiento de la petición (Gelbukh, 2000).
- Realizar el esquema de autoenriquecimiento del corpus: usar los archivos descargados de Internet para descubrir nuevas palabras que no están en la lista inicial e incluirlas en el diccionario (buscar *N* contextos para ellas).

Con el fin de alcanzar dicho autoenriquecimiento resultará útil generar las formas gramaticales de las palabras descubiertas. Para ello se usará un analizador y generador morfológico heurístico (Gelbukh y Sidorov, 2003a): así se podrán comprobar las hipótesis sobre la categoría gramatical (sustantivo, verbo, etc.) y el tipo de declinación o conjugación de la nueva palabra. El analizador también usará la Internet. Pongamos un ejemplo, para la palabra *internetizados* se generarán las formas hipotéticas *internetizar*, *internetizarán*, etc.; su presencia en Internet comprobará que la palabra inicial es el verbo del tipo de conjugación predicho.

Capítulo 12

CONSTRUCCIÓN AUTOMÁTICA DEL DICCIONARIO DE COLOCACIONES BASÁNDOSE EN UN ANÁLISIS SINTÁCTICO AUTOMÁTICO

Existe una demanda creciente de diversos recursos en la lingüística moderna y especialmente en el procesamiento de lenguaje natural (PLN). Un ejemplo importante de recurso lingüístico es una base de datos suficientemente grande de combinaciones de palabras. Esa base de datos, de hecho, es también un diccionario de combinaciones de palabras (Gelbukh *et al*., 2004). La relación entre los conceptos de colocación y de combinación de palabras se presenta en la siguiente sección.

El problema de cómo representar la información acerca de la compatibilidad de las palabras tiene una larga historia —los primeros artículos aparecen en los años 50; básicamente, se puede decir que el centro de atención de los investigadores fue el concepto de colocación. La directriz principal de los investigadores consistió en integrar este concepto a la práctica lexicográfica y a los métodos de enseñanza de las lenguas extranjeras, por ejemplo, cuántos modelos hay que poner en los diccionarios o libros de texto, si los ejemplos de colocaciones son solamente ejemplos de uso o son parte esencial del conocimiento de un lenguaje, etcétera.

Después de una discusión amplia, el punto de vista más común es que es muy difícil encontrar una definición concisa y formal de colocación. Sin embargo, la mayoría de los investigadores están de acuerdo en que las colocaciones son una parte importante del conocimiento de un lenguaje y también son útiles para diferentes tareas del procesamiento automático de lenguaje natural, como la traducción automática, la generación de texto, la recuperación inteligente de información, etc. Todo esto implica la necesidad de compilación de diccionarios especializados en colocaciones e incluso en combinaciones libres de palabras. En la siguiente sección se presenta una exposición más detallada del concepto de colocación.

Los diccionarios de combinaciones de palabras se pueden aplicar en la mayoría de las tareas de procesamiento automático de lenguaje natural. Eso se debe a que la información *a priori,* el hecho de que una palabra pueda tener cierta relación con alguna otra palabra, permite reducir drásticamente la ambigüedad, uno de los problemas principales del PLN. Veamos un ejemplo. En el caso clásico de la ambigüedad referencial *Juan tomó la torta de la mesa y la comió* el pronombre *la* se refiere a *la torta*, y no a *la mesa*, sin embargo, si sustituimos la palabra *comió* por *limpió*, el pronombre se referirá a la palabra *mesa*. Este tipo de problemas de referencia podrían ser resueltos si tuviéramos en nuestra base de datos las siguientes combinaciones de palabras: *comer* + *torta* y *limpiar* + *mesa*. Como una opción, se pueden tener las combinaciones *comer* + *comida* y *limpiar* + *mueble*, y después aplicar la inferencia basados en el conocimiento de que *torta* es *comida* y *mesa* es una especie de *mueble*. Obviamente pueden existir casos en los que la información de la base de datos no es suficiente, pero en la mayoría de las ocasiones esta información puede ayudar a resolver la ambigüedad.

Otro ejemplo de la importancia del conocimiento de las combinaciones más probables de palabras es la traducción automática.

Posiblemente en este caso sería mejor tener las bases de datos de ambos idiomas y la correspondencia entre ellas, pero aun con una sola base de datos se puede sacar suficiente provecho. Por ejemplo, es muy útil para detectar y/o traducir las funciones léxicas como *prestar atención* (*pay attention*, en inglés), cuando *prestar* no debe traducirse como *borrow*, sino como *pay* (véase la explicación en la siguiente sección).

También es útil la información de las combinaciones de palabras para la resolución de las ambigüedades sintácticas, por ejemplo, *barco de madera de roble*. La ambigüedad consiste en saber qué palabra gobierna a *de roble* —*madera* o *barco*. En el primer caso, si tuviéramos la combinación *madera de roble* en la base datos, entonces, la ambigüedad se resolvería. Nótese que si tuviéramos también la combinación *barco de madera*, la situación no crearía problemas, porque las dependencias sintácticas serían complementarias. Por otro lado, si tuviéramos también la combinación *barco de roble*, entonces, para poder resolver este tipo de ambigüedad, tendríamos la necesidad de agregar la información de los tipos de relaciones a nuestra base de datos de combinaciones de palabras.

Existen muchos métodos de extracción de colocaciones que están basados en el análisis de los grandes corpus (Baddorf y Evans, 1998; Basili *et al*., 1993; Dagan *et al*., 1999; Kim *et al*., 2001; Kita *et al*., 1994; Yu *et al*., 2003). La mayoría de estos métodos están orientados a la búsqueda de combinaciones de palabras sustentándose en la medida de su mutua información. Sin embargo, esos métodos no garantizan el descubrimiento de las colocaciones que no tienen una frecuencia suficientemente alta. Desafortunadamente, por más grande que sea el corpus, la gran mayoría de las combinaciones de palabras no tienen frecuencia alta. Además, parece que el tamaño del corpus para tal búsqueda debería ser notablemente mayor que los corpus existentes.

Uno de los trabajos clásicos sobre la extracción de colocaciones es (Smadja, 1993). En él se presenta el sistema *Xtract*, que permite

encontrar coocurrencias de palabras apoyándose en la información mutua. El trabajo sugiere tres fases de procesamiento, y en la tercera fase se aplica el análisis sintáctico parcial para rechazar los pares de palabras en los que no existe una relación sintáctica. Sin embargo, este procedimiento se aplica a los pares de palabras obtenidos con un umbral alto de frecuencias. El objetivo de este método es solamente determinar las colocaciones y no las combinaciones libres de palabras (véase la discusión en la siguiente sección). Se reportan la precisión de 80% y la especificidad (*recall*) de 94%, considerablemente altas del sistema *Xtract*. Sin embargo, la evaluación se realizó comparando los resultados del sistema con la opinión de un lexicógrafo. Tampoco se explica lo que es el término *colocación* en relación con el sistema —que obviamente implica algo con cierta frecuencia, aunque no esté claro cuál es la frecuencia deseada y qué pasa con las combinaciones menos frecuentes. De igual manera, no está claro si se procesan las combinaciones libres de palabras en caso de que tengan alta frecuencia.

Además, en el campo del PLN existen varios intentos por aplicar los resultados del análisis sintáctico automático (*parsing*) a diferentes tareas (Basili *et al*., 1993; Church *et al*., 1991). Uno de los ejemplos más reciente del uso del análisis sintáctico automático para las tareas relacionadas con combinaciones de palabras es el trabajo de (Strzalkowski *et al*., 1999). Se extraen las combinaciones de palabras relacionadas sintácticamente y se usan en recuperación de información, es decir, la consulta sobre este par de palabras asigna mucho más peso a los documentos donde existe la conexión sintáctica. Sin embargo, solamente participan los sustantivos con sus modificadores y no se discute el tratamiento de preposiciones ni conjunciones. Además, el área de aplicación de los resultados es la recuperación de información, que es muy distinta de nuestro trabajo; recalcamos que el propósito de trabajo influye mucho en la estructura de datos y en los procedimientos.

Existen algunos recursos disponibles que contienen información sobre las combinaciones de palabras. Uno de lo más grandes diccionarios de colocaciones y combinaciones de palabras libres es el sistema CrossLexica (Bolshakov, 1994; Bolshakov y Gelbukh, 2000; 2002). El sistema contiene alrededor de 1,000,000 de combinaciones de palabras para el ruso, junto con las relaciones semánticas entre ellas. Existe la posibilidad de inferencia semántica. Para el lenguaje inglés hay varios recursos de este tipo, aunque no tan grandes, por ejemplo, el diccionario Oxford (OCD, 2003) que contiene 170,000 combinaciones de palabras o el diccionario Collins (Bank of English) con 140,000 combinaciones de palabras. Dichos diccionarios no contienen las relaciones semánticas entre las palabras.

En este capítulo discutiremos primero el concepto de colocación y su relación con las combinaciones libres de palabras. Después, se discutirán los requerimientos generales para el análisis del lenguaje natural y se presentará el ambiente para el desarrollo de las gramáticas libres de contexto. Luego se describirá el método de construcción automática de una base de datos de combinaciones de palabras con base en el análisis sintáctico automático. Finalmente evaluaremos el desempeño del método propuesto en comparación con el método que usa bigramas.

12.1 Combinaciones idiomáticas, colocaciones y combinaciones libres de palabras

Ahora vamos a discutir el concepto de colocación con más detalle. Intuitivamente, la colocación es una combinación de palabras que tienen una tendencia clara a utilizarse juntas. Sin embargo, el grado de esta tendencia es diferente para diferentes combinaciones. Así, las colocaciones pueden ser vistas como una escala con grados diferentes de intensidad de relación entre las

palabras, que van desde las combinaciones idiomáticas hasta las combinaciones libres.

En un extremo de la escala hay combinaciones idiomáticas completas como, por ejemplo, *estirar la pata*, donde ni la palabra *estirar*, ni *la pata* pueden ser reemplazadas sin destruir el significado de la combinación. En este caso —y en todos los casos de combinaciones idiomáticas— el significado de toda combinación no está relacionado con los significados de sus componentes. Este tipo de combinaciones también pueden llamarse frasemas, como propone Mel'čuk (Mel'čuk, 1996).

En el otro extremo de la escala hay combinaciones totalmente libres de palabras, como, por ejemplo, *ver un libro,* donde cualquier palabra de la combinación puede ser sustituida por un conjunto grande de distintas palabras y el significado de toda la combinación es la suma de los significados de sus palabras constituyentes.

Más o menos en la mitad de la escala existe lo que se llaman funciones léxicas (Mel'čuk, 1996) como, por ejemplo, *prestar atención*. En este caso, el significado de la combinación está directamente relacionado solamente con una palabra —en el ejemplo, con la palabra *atención*— mientras que la otra palabra expresa cierta relación semántica estándar entre los actuantes de la situación. La misma relación se encuentra, por ejemplo en combinaciones como *estar en huelga*, *dar un grito*, etc. Usualmente para una relación semántica y para que una palabra conserve su significado (como argumento de la función), existe sólo una manera de elegirla palabra que expresa la relación predeterminada (valor de la función) en el lenguaje. Por ejemplo, en español es *prestar atención*, mientras que en inglés es *to pay attention* (literalmente, *pagar atención*), en ruso es *obratit' vnimanije* (literalmente, *mover atención hacía algo*), etcétera.

En lo que se refiere a las combinaciones libres de palabras, se puede ver que algunas de ellas son «menos libres» que otras,

aunque siguen siendo combinaciones libres de palabras en el sentido en que el significado de la combinación es la suma de los significados de las palabras constituyentes. El grado de libertad depende de cuántas palabras pueden usarse como sustitutos de cada palabra de la combinación. Mientras más pequeño sea el número de los sustitutos, más idiomática será la combinación de palabras; aunque estas combinaciones nunca alcanzarían la idiomaticidad de las combinaciones idiomáticas y de las funciones léxicas, en cuyos casos los significados no se suman.

Las restricciones en combinaciones libres de palabras tienen una naturaleza semántica, por ejemplo, *ver un libro* es menos idiomática que *leer un libro*, debido a que existen muchas más palabras que pueden sustituir *libro* combinándose con el verbo *ver* que con el verbo *leer*. Es decir, prácticamente cualquier objeto físico puede *verse*, mientras que solamente los objetos que contienen alguna información escrita (o la extensión metafórica de la información escrita, como, digamos, «*leer los signos de enojo en su cara*») pueden *leerse*.

Otro punto importante es que algunas combinaciones libres de palabras pueden tener relaciones asociativas entre sus miembros, es decir, *un conejo* puede *saltar* y *una pulga* también, porque es su modo usual de moverse. Sin embargo, *un lobo* usualmente no *salta* aunque potencialmente puede moverse de esa manera. Esto hace algunas combinaciones de palabras más idiomáticas, debido a que la relación entre las palabra se fortalece por asociación.

En sentido estricto, solamente las funciones léxicas son colocaciones, pero el manejo común de este concepto también se expande a las combinaciones libres «más idiomáticas». Debido a que no existe una división muy clara entre las combinaciones más idiomáticas y menos idiomáticas, el concepto de colocación finalmente puede cubrir también muchas combinaciones libres de palabras.

Como se demuestra en la discusión anterior, las dificultades de definición del concepto de colocación están relacionadas con

la imposibilidad de distinguir de manera clara el grado de idiomaticidad en las combinaciones de palabras.

Recalcamos que la solución obvia de usar el término colocación solamente para las funciones léxicas contradice a la práctica común. Parece que la mejor solución es conformarse con esta forma de usar el término, ya aceptada por la comunidad científica. La alternativa sería inventar algún otro término para las combinaciones libres con cierto grado de idiomaticidad.

12.2 Enriquecimiento automático del diccionario de colocaciones

Tradicionalmente, las combinaciones libres de palabras se han considerado de poco interés para la lingüística. Sin embargo, como ya mencionamos, cualquier combinación libre de palabras no es totalmente libre —es hasta cierto grado idiomática, debido a que la mayoría de las combinaciones tienen restricciones semánticas para la compatibilidad de sus constituyentes. Entonces, si hay restricciones, cualquier combinación de palabras que se considere posible contiene la información de que las palabras que la integran son compatibles. Aquí nos gustaría recordar la bien conocida idea de Firth —«conocerás la palabra por la compañía que ésta mantiene».

Nótese que la compilación manual de una base de datos de combinaciones de palabras es una tarea que consume demasiado tiempo. Por ejemplo, el sistema CrossLexica (Bolshakov y Gelbukh, 2000) ha sido compilado durante más de 14 años y todavía está muy lejos de su terminación.

La alternativa para el método manual es usar algún método automático. En este caso tenemos dos posibilidades: usar el método directo que toma las palabras vecinas (método de bigramas) o tratar de aplicar los resultados del análisis sintáctico.

De antemano, el segundo método tiene la ventaja de aplicar más conocimiento lingüístico, por lo tanto, se esperan de él resultados mejores. Más adelante vamos a describir los dos métodos y a hacer la comparación entre ellos para un texto en español.

Usamos el análisis sintáctico automático basándonos en el *parser* y la gramática para el español descritos en (Galicia Haro *et al*., 2001). Los resultados del análisis sintáctico se representan utilizando el formalismo de dependencias (Mel'čuk, 1988). La idea del formalismo de dependencias es que las relaciones sintácticas entre las palabras se representan directamente usando, por ejemplo, flechas. Las relaciones están asociadas directamente con los pares de palabras, así no es necesario pasar por el árbol de constituyentes para obtener cada relación; una palabra siempre es la cabeza de una relación y la otra palabra es su dependiente; una palabra principal puede tener varias dependientes. Es bien conocido que el poder expresivo de este formalismo es equivalente al formalismo de constituyentes. En nuestro caso preferimos las dependencias porque este formalismo contiene la representación inmediata de las combinaciones de palabras.

Es obvio que el método para la construcción de la base de datos no depende esencialmente del idioma y es fácilmente aplicable para cualquier lenguaje si se dispone de una gramática y un *parser*. También hay que mencionar que no existen muy buenas gramáticas formales para el español, por lo tanto, el método necesita posverificación del análisis sintáctico; sin embargo, aun así es mucho más eficiente que el procedimiento manual.

El método enfrenta algunos problemas adicionales que es necesario resolver:

- Determinar qué información es necesario guardar en la base de datos.
- Tratamiento de las conjunciones coordinativas.
- Tratamiento de las preposiciones.

– Filtrado de algunos tipos de relaciones y algunos tipos de nodos (pronombres, artículos, etcétera).

Almacenamos las combinaciones obtenidas en una base de datos; tanto la palabra principal como la dependiente están normalizadas, pero también se guarda alguna información acerca de la forma gramatical de la dependiente. En nuestro caso, para los sustantivos almacenamos la información del número —singular o plural— es decir, *leer libro Sg* es un registro y *leer libro Pl* (equivale a *leer libros*) es otro en la base de datos. Para los verbos guardamos la información si es gerundio, participio o infinitivo —recordemos que el verbo finito no puede ser dependiente. También se guarda la frecuencia de la combinación. Claro que en el momento de la consulta se puede ignorar la información gramatical si es irrelevante.

Las conjunciones coordinativas son cabezas en la relación coordinativa, pero las combinaciones de palabras que deben ser agregadas a la base de datos son las combinaciones entre la cabeza y sus dependientes. Por ejemplo, *leo un libro y una carta*, las combinaciones que deben extraerse son *leo un libro* y *leo una carta*; así, el método detecta esta situación y genera dos combinaciones virtuales que se agregan a la base de datos.

El tratamiento de la relación preposicional es diferente de otras relaciones. Dado que las preposiciones usualmente expresan relaciones gramaticales entre palabras —en otros lenguajes estas relaciones pueden expresarse por los casos gramaticales, por ejemplo— la relación importante no es la relación con la preposición, sino la relación entre los dos lexemas conectados a través de la preposición. Sin embargo, la preposición por sí misma también es de interés lingüístico, porque contiene información acerca de la preposición que se usa. Por eso, almacenamos en la base de datos la combinación de palabras que contiene los tres miembros: la cabeza de la preposición, la preposición, y su

dependiente, es decir, *Él juega con el niño* nos da la combinación *jugar* + *con* + *niño.*

Se aplican dos tipos de filtros —filtros de los nodos basados en las categorías gramaticales y filtros de los tipos de relaciones.

El filtrado basado en las categorías gramaticales es fácil, dado que el *parser* se basa en el análisis morfológico, y, por lo tanto, la información morfológica de cada palabra está disponible. Esto permite filtrar, por ejemplo, las combinaciones sin el contenido léxico importante, tales como pronombres (personales, demostrativos, etc.), artículos, conjunciones subordinativas, negación y los números. Las palabras de estas categorías no tienen restricciones semánticas en compatibilidad y no son de interés para la base de datos que estamos construyendo. El usuario puede crear sus propios filtros para las otras categorías gramaticales.

El filtro de las relaciones se aplica para diferentes tipos de ellas; este filtro depende de la gramática que se utiliza y de que están contemplados los diferentes tipos de relaciones sintácticas. En la gramática formal que usamos existen las siguientes relaciones para la palabra dependiente: *dobj* (objeto directo), *subj* (sujeto), *obj* (objeto indirecto), *det* (modificador que es un artículo o un pronombre), *adver* (adverbial), *cir* (circunstancial), *prep* (preposicional), *mod* (modificador que no es un artículo o un pronombre), *subord* (subordinativa), *coord* (coordinativa). De entre estas relaciones, la preposicional y la coordinativa son tratadas de un modo especial, como ya mencionamos. Las únicas relaciones que no se usan para detectar las combinaciones de palabras en la versión actual del método son la relación subordinativa y la relación circunstancial.

Una de las ventajas del método sugerido es que no se necesita un corpus para su entrenamiento y así no depende del tamaño del mismo o de su estructura léxica.

Vamos a presentar dos ejemplos del funcionamiento del método propuesto. La siguiente oración se analiza automáticamente.

Conocía todos los recovecos del río y sus misterios.

El siguiente árbol de dependencias corresponde a esta oración.

```
1  V(SG,1PRS,MEAN)                  // Conocía : conocer
2    CONJ_C {dobj}                  // y : y
3      N(PL,MASC)                   // recovecos : recoveco
4        PR {prep}                  // del : del
5          N(SG,MASC) {prep}        // río : río
6        ART(PL,MASC) {det}         // los : el
7        #*$$todo#                  // todos : todo
8      N(PL,MASC)                   // misterios : misterio
9        DET(PL,MASC) {det}         // sus : su
10   $PERIOD .                      // . : .
```

La jerarquía de profundidad en el árbol corresponde a las dependencias —se expresa con un número de espacios al inicio de cada línea. Normalmente se usarían flechas, pero en el procesamiento automático es más fácil la representación que presentamos. Las palabras dependientes se encuentran en el siguiente nivel inmediato —están alineados verticalmente. Por ejemplo, V(SG, 1PRS, MEAN) [*conocía*] es la cabeza de la oración y sus dependientes son CONJ_C [*y*] y $PERIOD. Por otro lado, CONJ_C tiene como sus dependientes a N(PL, MASC) [*recovecos*] y N(PL,MASC) [*misterios*], etc. Cada línea corresponde a una palabra y contiene la forma de la palabra y su lema, por ejemplo, *conocía* : *conocer,* etcétera.

Se detectaron las siguientes combinaciones de palabras (sin contar las combinaciones con palabras gramaticales):

1. *conocer (dobj) recoveco* {Pl}
2. *conocer (dobj) misterio* {Pl}
3. *recoveco (prep) [del] río* {Sg}

Se puede ver que la relación (*dobj)* corresponde a la conjunción coordinativa, y entonces se propaga a sus dependientes: *recoveco* y *misterio*. La preposición *del* es el tercer miembro de la combinación numero tres. Las combinaciones con los artículos y pronombres (*el, todo, su*), aunque las encuentre el algoritmo, se filtraron porque contienen palabras gramaticales.

Ahora presentamos el otro ejemplo.

Compró una pequeñita torta y pastel con una bailarina con zapatillas de punta.

El siguiente árbol corresponde a este ejemplo.

```
1   V(SG,3PRS,MEAN)                  // compró: comprar
2     CONJ_C {obj}                   // y: y
3       N(SG,FEM) {coord_conj}       // torta: torta
4         ADJ(SG,FEM) {mod}          // pequeñita: pequeñito
5         ART(SG,FEM) {det}          // una: un
6       N(SG,MASC) {coord_conj}      // pastel: pastel
7         PR {prep}                  // con: con
8           N(SG,FEM) {prep}         // bailarina: bailarina
9             PR {prep}              // con: con
10              N(PL,FEM) {prep}     // zapatillas: zapatilla
11                PR {prep}          // de: de
12                  N(SG,FEM) {prep} // punta: punta
13      ART(SG,FEM) {det}            // una: un
14  N(SG,FEM) {subj}                 // mamá: mamá
15  $PERIOD      // . : .
```

Se detectaron las siguientes combinaciones de palabras. Nótese que las combinaciones 4 y 7 son filtradas por la misma razón que las anteriores —son combinaciones con las palabras gramaticales.

1. *comprar* (obj) *torta*{Sg}
2. *comprar* (obj) *pastel* {Sg}

3. *torta* (mod) *pequeñito*
4. **torta* (det) *un*
5. *pastel* (mod) [*con*] *bailarina* {Sg}
6. *bailarina* (mod) [*con*] *zapatilla* {Pl}
7. **bailarina* (det) *un*
8. *zapatilla* (mod) [*de*] *punta* {Sg}
9. *comprar* (subj) *mamá* {Sg}

12.3 Evaluación del enriquecimiento automático

Para evaluar este método automático, hicimos el experimento con un texto en español elegido aleatoriamente de la Biblioteca Digital «Cervantes». El texto está formado por 60 oraciones que contienen 741 palabras, en promedio 12.4 palabras por oración. Para la evaluación, marcamos manualmente todas las relaciones de dependencias en las oraciones. Después comparamos las combinaciones de palabras encontradas automáticamente con las combinaciones de palabras marcadas manualmente.

También usamos un método base para comparar los resultados del método que usa el análisis sintáctico. Como método base tomamos el método de bigramas, que toma todos los pares de palabras que son vecinos inmediatos. Adicionalmente, agregamos cierta inteligencia a este método base —él ignora los artículos y toma en cuenta las preposiciones. En total, hubo 153 artículos y preposiciones en las oraciones, así que el número de palabras para el método base es 741 – 153 = 588.

Se obtuvieron los siguientes resultados. El número total de combinaciones de palabras correctas marcadas manualmente es 208. De estas, 148 combinaciones de palabras fueron encontradas por nuestro método. Al mismo tiempo, el método base encontró 111 combinaciones de palabras correctas. Del otro lado, nuestro método encontró solamente 63 combinaciones de palabras inco-

rrectas, mientras que el método base marca como una combinación de palabras 588 × 2 – 1= 1175 pares de palabras vecinas, de los cuales 1175 – 111 = 1064 son combinaciones erróneas.

Estos números nos dan los siguientes valores de precisión y especificidad (*recall*, en inglés). Recordemos que la precisión es la relación entre los resultados correctos sobre los resultados obtenidos en total, mientras que la especificidad es la relación entre los resultados correctos sobre los resultados que deberían haber sido obtenidos. Para nuestro método, la precisión es 148 / (148 + 63) = 0.70 y la especificidad es 148 / 208 = 0.71. Para el método base, la precisión es 111 / 1175 = 0.09 y la especificidad es 111 / 208 = 0.53. Es obvio que la precisión de nuestro método es mucho mejor y la especificidad es mejor que en los resultados del método base.

12.4 Conclusiones

Una base de datos de combinaciones de palabras es un recurso lingüístico muy importante. Sin embargo, la compilación y el enriquecimiento manual de este diccionario implican una tarea que consume demasiado tiempo y esfuerzo. Propusimos un método que brinda la posibilidad de construir bases de datos de este tipo semiautomáticamente. El método se basa en el análisis sintáctico automático, usando el formalismo de dependencias y la extracción de combinaciones de palabras.

Se presentó brevemente un ambiente que facilita el desarrollo y la depuración de los analizadores de textos en español. El sistema contiene un analizador morfológico del español y un *parser* sintáctico que incorpora la tecnología de ponderación de las variantes sintácticas desarrollada en nuestro Laboratorio. El sistema se está usando activamente para el desarrollo del analizador sintáctico de alta calidad que se apoya en los diccionarios de compatibilidad de las palabras en español.

Algunos tipos de relaciones y algunos tipos de nodos se filtran debido a que no contienen la información léxica importante. Se implementa un procedimiento especial para las relaciones coordinativas y preposiciones. El método requiere de un posprocesamiento de las combinaciones de palabras obtenidas, pero solamente para verificar que no se presenten errores del *parser*.

Los resultados se evaluaron sobre un texto en español elegido aleatoriamente. El método propuesto tiene mucha mejor precisión y especificidad que el método base que obtiene los bigramas.

Capítulo 13

EVALUACIÓN AUTOMÁTICA DE LA CALIDAD DE LOS DICCIONARIOS EXPLICATIVOS

Las palabras en los diccionarios explicativos contienen diferentes significados (sentidos), esto se conoce como el fenómeno de polisemia. Entre tanto, en los textos reales las palabras tienen un sentido único, dependiendo del contexto de uso. El problema al escoger un sentido de la palabra usado en el texto es conocido como la desambiguación de sentidos de palabras (DSP; en inglés WSD, *word sense disambiguation*), y es muy popular en la lingüística computacional moderna (Manning y Schütze, 1999).

En este capítulo, sin embargo, no vamos a trabajar directamente con esta desambiguación, sino que trataremos un problema parecido con un método de solución muy similar a algunos métodos de DSP. Sugerimos calcular la semejanza semántica entre diferentes sentidos de la misma palabra (Gelbukh *et al.*, 2003a). La tarea principal de nuestro experimento es obtener la posibilidad de evaluar la calidad de los diccionarios explicativos, ya que en la actualidad los diccionarios se evalúan intuitivamente. ¿Hay criterios objetivos para esta evaluación intuitiva? Proponemos hacer la evaluación de la calidad de un diccionario calculando la semejanza semántica de varios sentidos de la misma palabra;

la idea es que en un «buen» diccionario los sentidos son diferentes entre sí, mientras que en un «mal» diccionario son similares; este fenómeno puede ocurrir debido a la incorrecta diferenciación de los sentidos o a la granulación de sentidos demasiado fina o simplemente a la mala definición. La semejanza entre dos sentidos dados se calcula como un número relativo de palabras iguales o sinónimas en sus definiciones dentro del diccionario explicativo. Hicimos nuestros experimentos usando el *Diccionario explicativo Anaya de la lengua española*.

En este caso usamos la normalización morfológica, lo que significa que se pueden identificar las formas morfológicas diferentes de la misma palabra, por ejemplo, *trabajaré*, *trabajó, trabajábamos,* etc., como pertenecientes al mismo lema *trabajar*. También usamos un diccionario de sinónimos del español para detectar los sinónimos.

En el resto del capítulo, primero discutimos los datos experimentales más a detalle, luego describimos el experimento y sus resultados, y finalmente se dan algunas conclusiones.

13.1 Los datos para el experimento

Usamos el diccionario *Anaya* como fuente de las palabras y sus sentidos. Este diccionario contiene más de 30,000 palabras y más de 60,000 sentidos. Preferimos este diccionario sobre el diccionario *WordNet* en español (Fellbaum, 1998) debido a que el *WordNet* tiene las definiciones en inglés y las herramientas y los datos que tenemos son para el español.

Para el procesamiento morfológico aplicamos el analizador morfológico descrito en el capítulo 5.

Normalizamos todas las definiciones del diccionario y aplicamos el procedimiento de asignación de partes de la oración (*POS tagging*), como se plantea por ejemplo en (Sidorov y Gelbukh,

2001); este procedimiento tiene ventaja sobre los *taggers* tradicionales porque esta orientado a los diccionarios. A diferencia de (Sidorov y Gelbukh, 2001), en el experimento se ignoraron los posibles sentidos que se asignan a cada palabra de la definición, porque esta asignación da un porcentaje de errores que preferimos eliminar. Los diferentes sentidos de palabras en la definición pueden ser tomados en cuenta en futuros experimentos.

También usamos el diccionario de sinónimos del español que contiene alrededor de 20,000 entradas. Este diccionario se aplica en el algoritmo de medición de semejanza para la detección de las palabras sinónimas en definiciones.

A continuación mostramos un ejemplo de normalización morfológica de una definición del diccionario.

La definición de la palabra «*abad*» en uno de sus sentidos es como sigue:

Abad = Título que recibe el superior de un monasterio o el de algunas colegiatas.

La versión normalizada para esta definición quedaría así:

Abad = título#noun que#conj recibir#verb el#art superior#noun de#prep un#art monasterio#noun o#conj el#art de#prep alguno#adj colegiata#noun .#punct

Donde #*conj* es la conjunción, #*art* es el artículo, #*prep* es la preposición, #*adj* es el adjetivo, #*punct* es la marca de puntuación, y #*noun* y #*verb* se usan para el sustantivo y el verbo. Hay algunas palabras que no son reconocidas por el analizador morfológico (alrededor de 3%), a éstas se les asigna la marca #*unknown* («desconocida»).

Otra consideración importante relacionada con el análisis morfológico es que en la etapa de la comparación es deseable ignorar las palabras auxiliares, porque normalmente no agregan ninguna información semántica —como conjunciones o artículos— o agregan información arbitraria —como las preposiciones que dependen normalmente del marco de subcategorización del verbo.

13.2 El experimento

En el experimento medimos la semejanza entre diferentes sentidos de la misma palabra.

Usamos la medida natural de la semejanza entre dos textos, conocida como el coeficiente de Dice (Jiang y Conrad, 1999; Rasmussen, 1992). La fórmula para este coeficiente es la que sigue:

$$D(t_1,t_2) = \frac{2 \times |W_1 \cap W_2|}{|W_1| + |W_2|}$$

donde W_1 y W_2 son las palabras del texto t_1 y t_2. Este coeficiente caracteriza la intersección literal de las palabras en el texto, lo que se expresa a través de $W_1 \cap W_2$ donde tomamos las palabras que existen en ambos textos —o como nosotros, en ambas definiciones.

No obstante, en nuestro caso, queremos considerar también los sinónimos de las palabras presentes en las definiciones de los sentidos. Por lo que modificamos la fórmula para calcular la semejanza como sigue:

$$S(t_1,t_2) = \frac{|W_1 \cap W_2| + |W_1 \circ W_2|}{\max(|W_1|,|W_2|)}$$

Aquí el símbolo «o» significa que calculamos la intersección usando sinónimos (véase la descripción del algoritmo a continuación). Tuvimos que usar el valor máximo de número de palabras para la normalización, porque todas las palabras de cualquiera de las definiciones pueden ser sinónimas o coincidir literalmente con las palabras de otra definición. En esta fórmula no hay necesidad de multiplicar por dos porque no sumamos el número de palabras en ambos textos. Es obvio que los sinónimos pueden tomarse con cierto peso, pero para los propósitos de nuestro experimento es importante medir la semejanza máxima posible. Además, en nuestra opinión, los sinónimos, en este cálculo de

semejanza, deben tratarse igual que las palabras que tienen intersección literal, porque, por la definición, los sinónimos tienen significados similares y se distinguen normalmente sólo por el matiz de sus significados. Así, aunque a veces no es posible sustituir un sinónimo con el otro en un texto, los sinónimos expresan más o menos el mismo concepto.

Los pasos del algoritmo son los siguientes. Para cada palabra en el diccionario medimos la semejanza entre sus sentidos; obviamente, las palabras con un solo sentido se ignoraron —encontramos que hay alrededor de 13,000 palabras con un solo sentido de un total de 30,000. Puesto que la semejanza es una relación simétrica, se calcula solamente una vez para cada par de sentidos de las palabras.

Nótese que consideramos los homónimos como palabras diferentes y no como sentidos diferentes. Normalmente, ésta es la manera como se representan en los diccionarios —como diversos grupos de sentidos. Además, los homónimos tienen muy distintos significados, resultando de poco interés para nuestra investigación.

Medimos la semejanza de la siguiente manera. Al principio, el contador de la semejanza es cero. Las palabras no auxiliares se toman una por una del primer sentido en el par y se buscan en el otro sentido; si la palabra se encuentra en el otro sentido se incrementa el contador. Nótese que utilizamos palabras normalizadas con las características POS asignadas, es decir, no solamente el lema debe coincidir, también su característica POS. Eso permite considerar, sobre todo, únicamente las palabras significativas e ignorar las palabras auxiliares. Si la palabra se encuentra en la otra definición, esta se agrega a la lista auxiliar de las palabras ya procesadas para evitar que se vuelva a contar mientras se procesa el otro sentido del par.

Las palabras desconocidas se procesan igual a las palabras significativas, pero solamente si su longitud es mayor que un

umbral dado. En los experimentos utilizamos un umbral igual a dos letras, es decir, prácticamente todas las palabras desconocidas participan.

Si la palabra no existe textualmente en el otro sentido, buscamos sus sinónimos en el diccionario de sinónimos y los comparamos uno por uno con las palabras del otro sentido, comprobando antes que no están en la lista de las palabras ya procesadas. Si se encuentra el sinónimo, se incrementa el contador y el sinónimo se agrega a la lista de las palabras ya procesadas. Todos los sinónimos se verifican en esta lista antes de que se compare con las palabras del otro sentido. Esta lista se vacía al terminar el procesamiento de cada par de sentidos.

En el paso siguiente, el procedimiento se repite para el otro sentido en el par. Por supuesto, las palabras ya encontradas literalmente o a través de su sinónimo en la otra definición ya no participan en el conteo. Seguimos buscando los sinónimos porque no podemos garantizar que nuestro diccionario de sinónimos sea simétrico —si A es sinónimo de B, entonces no está garantizado que B es sinónimo de A; debido a la calidad del diccionario de sinónimos.

Finalmente aplicamos la fórmula para calcular el coeficiente de semejanza S entre los dos sentidos.

Puesto que la semejanza es una fracción, algunas fracciones son más probables que otras. Específicamente, debido a un alto número de definiciones cortas, había muchas fracciones con denominadores pequeños, tales como 1/2, 1/3, 2/3, etc. Para suavizar este efecto representamos los resultados experimentales por los intervalos de valores y no por los valores específicos. Utilizamos cuatro intervalos iguales para el porcentaje de semejanza entre los sentidos también se puede utilizar el otro número de los intervalos si no es muy grande. De cualquier forma, eso no cambia significativamente los resultados porque sólo usamos los intervalos para la estimación. Presentamos los resultados para la

semejanza cero por separado porque de antemano sabíamos que hay muchos sentidos sin intersección alguna.

Los resultados del experimento se muestran en la tabla 5.

Tabla 5. Número de pares de sentidos por intervalos.

Intervalo de semejanza	Número de los pares de sentidos	Porcentaje de los pares de sentidos
0.00–0.01	46205	67.43
0.01–0.25	14725	21.49
0.25–0.50	6655	9.71
0.50–0.75	600	0.88
0.75–1.00	336	0.49

Como se nota, alrededor de 1% de los pares de sentidos son muy similares, conteniendo más de 50% de las mismas palabras (incluyendo los sinónimos), y cerca de 10% de los pares de sentidos son significativamente similares conteniendo más de 25% de las mismas palabras. En nuestra opinión, 10% de definiciones significativamente similares es un valor bastante alto, así que las definiciones se deben revisar y así esperamos que el diccionario mejore.

Nótese que no estamos evaluando el número de las definiciones que usan sinónimos como un tipo especial de definición, sino el número de las definiciones parecidas basándonos en los sinónimos.

13.3 Conclusiones

Propusimos un método de evaluación automática de la calidad de los diccionarios explicativos usando la comparación de sentidos de la misma palabra —los sentidos no deben ser demasiado

similares—. Aunque es solamente un aspecto de la calidad de los diccionarios, esta característica es muy importante. El método consiste en calcular la intersección de los sentidos que se normalizan previamente; durante esta comparación se considera la intersección literal y la intersección basada en los sinónimos de las palabras en las definiciones de los distintos sentidos.

El experimento se realizó para el diccionario *Anaya* de la lengua española. Los resultados demuestran que cerca de 10% de pares de sentidos son significativamente similares —con más de 25% de palabras afines. En nuestra opinión, ese porcentaje es demasiado alto, por lo que el diccionario debe ser revisado. En el futuro planeamos realizar este experimento con otros diccionarios, como, por ejemplo, el *WordNet*.

Capítulo 14

DETECCIÓN AUTOMÁTICA DE LAS PRIMITIVAS SEMÁNTICAS

La manera natural para construir un diccionario semántico orientado a los sistemas computacionales de inferencia lógica e inteligencia artificial, es la definición de palabras a través de otras conocidas previamente.

Por ejemplo, en matemáticas los términos se definen a través de otros, de tal manera que en cualquier definición se puede sustituir cualquier término (digamos, *bisectriz*) por su definición (*línea que divide el ángulo en partes iguales*) sin alterar el sentido. El sistema de definiciones se construye de tal manera que al repetir este proceso iterativamente, se llega a una definición larga que consiste sólo de los términos que se llaman primitivos, tales como *punto* y *línea*.

Estos últimos no tienen ninguna definición dentro del sistema lógico, porque si la tuvieran causarían círculos viciosos: el *punto* se definiría a través del mismo término, lo que representa un problema grave para el razonamiento lógico. Entonces, cualquier sistema de definiciones lógicas sin círculos viciosos tiene que usar palabras primitivas no definidas en este sistema. El hecho es muy simple de demostrar matemáticamente usando el modelo de grafo descrito más adelante en este capítulo.

En la teoría lexicográfica también se reconoce que todas las palabras se deben definir usando unas pocas palabras primitivas, aunque hay controversias en las opiniones sobre el número de las primitivas necesarias, desde unas 60 (Wierzbicka, 1980; 1996) hasta unos miles (Apresjan, 1974; 1995).

En la lexicografía práctica se reconoce que un diccionario debe usar un número reducido de palabras en las definiciones. Por ejemplo, el *Longman dictionary of contemporary English* usa en sus definiciones sólo las palabras del vocabulario definidor (*defining vocabulary*) acotado, conocido como *Longman defining vocabulary* —alrededor de dos mil palabras.

En cuanto a los ciclos, es intuitivamente obvio que incluso en un diccionario orientado al lector humano (y mucho más en uno orientado a los sistemas de razonamiento automático) un conjunto cíclico de definiciones como:

gallina: hembra de gallo.
gallo: macho de gallina.
abeja: insecto que segrega miel.
miel: sustancia que producen las abejas.
convenio: pacto, acuerdo.
acuerdo: pacto, tratado.
tratado: convenio.

Es un defecto en el sistema de definiciones, ya que equivale a decir *gallina es una hembra del macho de gallina*, lo que no ayuda a entender qué es una gallina si no se sabe de antemano.

Sin embargo, estas definiciones son ejemplos reales (un poco simplificados) del Diccionario Explicativo del ruso de Ozhegov (uno de los más usados; primer ejemplo) y el *Diccionario Anaya de la Lengua Española*.

Ahora bien, ¿se puede convertir un diccionario explicativo tradicional en un sistema de definiciones lógicas para el razonamiento

automático en los programas de inteligencia artificial? De nuestra discusión es claro que, para esto, en primer lugar, se requiere detectar y eliminar los círculos viciosos en las definiciones.

En algunos casos esto se puede lograr cambiando manualmente las definiciones. Sin embargo, un paso inevitable en este proceso es declarar algunas palabras primitivas, es decir, no definidas dentro de este sistema lógico, de tal manera que todas las demás palabras se definan a través de éstas, ya sea directamente o indirectamente. Conviene que este conjunto definidor sea lo más reducido posible.

Aquí presentamos una solución basada en los métodos de la lexicografía computacional (Saint-Dizier y Viegas, 1995). Específicamente, hemos desarrollado una herramienta que permite al lexicógrafo detectar los círculos viciosos en el diccionario y elegir el conjunto definidor (Gelbukh y Sidorov, 2002b). La herramienta se basa en un algoritmo que genera automáticamente una variante del conjunto definidor mínimo (aunque no el menor posible; la diferencia se explicará más adelante).

En el resto del capítulo, describimos primero la herramienta, y después explicamos brevemente el algoritmo mencionado. Para esto, definimos la estructura de datos que usamos para la investigación del diccionario. Después describimos el algoritmo, la metodología experimental y los resultados de nuestros experimentos con un diccionario real. Finalmente, mencionamos las tareas futuras y formulamos las conclusiones.

14.1 La estructura de datos

Para el funcionamiento del algoritmo, así como para las definiciones y discusiones matemáticas, representamos el diccionario como un grafo dirigido (Evens, 1988; Fellbaum, 1990; Kozima y Furugori, 1993).

Los vértices de este grafo son las palabras que se mencionan en el diccionario —tanto las palabras encabezado como las que se usan en las definiciones. Si la misma palabra ocurre en diferentes contextos, se cuenta como el mismo vértice.

Las flechas del grafo se definen como sigue: la flecha desde la palabra v_1 hasta la v_2 significa que en la definición de la palabra v_1 ocurre la palabra v_2. Las palabras que no se definen en el diccionario no tienen flechas salientes, y las que no se usan en las definiciones de otras palabras no tienen flechas entrantes.

Nótese que hay diferentes maneras de considerar que dos ocurrencias textuales corresponden a «la misma palabra»: 1) cuando coinciden como cadenas de letras, 2) por el lema, 3) por la raíz común, 4) por el significado específico en el cual se usan en un contexto dado, etc. En la sección 14.3 se dan más detalles sobre los dos métodos que aplicamos (Grafo 1 —por el lema— y Grafo 2 —por significado).

14.2 El algoritmo

Esta sección se orienta al lector interesado en los aspectos matemáticos y técnicos del algoritmo. El lector interesado sólo en las aplicaciones y resultados, puede omitirla.

Desde el punto de vista matemático, el problema y su solución son los siguientes.

Definiciones

Sea $\mathbf{G} = \{\mathbf{V}, \mathbf{F}\}$ un grafo dirigido definido por los conjuntos $\mathbf{V}$ de N vértices y $\mathbf{F} \subseteq \mathbf{V} \times \mathbf{V}$ de flechas. Por ciclo en este grafo, entenderemos un ciclo dirigido.

Sea un subconjunto $\mathbf{P} \subseteq \mathbf{V}$ un *conjunto definidor* si cualquier ciclo en el grafo **G** contiene un vértice de **P**. En otras palabras, si el grafo $\mathbf{G'} = \{\mathbf{V'}, \mathbf{F'}\}$, donde $\mathbf{V'} = \mathbf{V} \setminus \mathbf{P}$ y $\mathbf{F'} = \mathbf{F} \cap (\mathbf{V'} \times \mathbf{V'})$, no tiene ciclos. Llamaremos a los vértices $p \in \mathbf{P}$ los definidores.

Un conjunto definidor $\mathbf{P} \subseteq \mathbf{V}$ es *mínimo* si ningún subconjunto $\mathbf{P'} \subset \mathbf{P}$ es definidor. Es decir, para cada vértice $p \in \mathbf{P}$, existe un ciclo en **G** que contiene *p* y no contiene ningún otro vértice de **P**.

El conjunto definidor mínimo no tiene que ser el menor (el que se contiene en todos los conjuntos definidores; tal subconjunto usualmente no existe) ni siquiera del tamaño menor posible: como es muy fácil mostrar con ejemplos y como también será claro de nuestro algoritmo, en el mismo grafo pueden existir muchos conjuntos definidores mínimos de tamaños diferentes.

El algoritmo que aquí presentamos resuelve el siguiente problema: *dado un grafo dirigido* **G**, *encontrar un subconjunto definidor* **P** *mínimo, aunque no del tamaño menor posible*.

El algoritmo encuentra una de los muchísimos posibles conjuntos definidores mínimos. La selección de la variante se puede controlar usando un ordenamiento σ que organiza los números de 1 a *N* en una secuencia $\sigma(1), ..., \sigma(N)$. Por ejemplo: 3, 5, 2, 4, 1 es un ordenamiento para $N = 5$.

Los conjuntos **P** generados basándose en diferentes ordenamientos σ son usualmente diferentes. Como se verá del algoritmo, el sentido del ordenamiento es el siguiente: los primeros vértices tienden a no ser definidores y los últimos tienden a entrar en **P**. Específicamente, el primer vértice en el ordenamiento nunca es definidor (si no tiene un lazo).

Este ordenamiento se puede definir de antemano, o bien generar el siguiente número, de entre los números todavía no usados según alguna estrategia (nosotros usamos este método), en cada paso del algoritmo.

Funcionamiento

Ahora bien, dado **G** y σ , el algoritmo funciona como sigue. Se construye, paso a paso, un subgrafo **G′** ⊆ **G** sin ciclos. Primero, es vacío. Se construye insertándole uno por uno —en el dado orden σ — los vértices de **G** con sus relaciones con los vértices ya insertados. En cada paso, **G′** se mantiene sin ciclos: si el vértice a insertar le causa ciclos, se considera un definidor y no se inserta en **G′**. Al terminar el proceso, se tiene el conjunto definidor **P**, que por su construcción es mínimo (porque cada $p \in \mathbf{P}$ tiene ciclos incluso en un subgrafo del **G′**).

El paso computacionalmente más costoso del algoritmo es la verificación de que el nuevo vértice no causa ciclos al **G′**. Para esto, para cada vértice $v_i \in \mathbf{V}$ el algoritmo mantiene un conjunto $\mathbf{A}_i$ de vértices accesibles en **G′** desde v_i, diciéndose del vértice u que es accesible desde v si existe un camino dirigido en el grafo desde v hacia u. El algoritmo aprovecha que es una relación transitiva.

También el algoritmo mantiene el conjunto **V′** de los vértices ya incluidos en el **G′** y el conjunto **P**. Al terminar el algoritmo, **P** será un conjunto definidor mínimo. Véase el algoritmo detallado en la ilustración 16.

Dado que el tamaño de las definiciones en un diccionario es limitado, el algoritmo tiene complejidad cuadrática en N, siendo las operaciones computacionalmente más pesadas los pasos 7 y 14 (la demostración está fuera del alcance de este libro). Dado el gran tamaño del diccionario (N = 30 mil en nuestro caso), empleamos un paso adicional (paso 2) para disminuir el tamaño del grafo antes de la aplicación del algoritmo, como se describe en la siguiente sección.

Depuración inicial del grafo

Para disminuir el tamaño de los datos a procesar, se pueden observar los siguientes hechos:

1. Formar el conjunto de los definidores **P** = Ø y de los no definidores **V′** = Ø.
2. Paso opcional: depuración del grafo, véase más adelante el algoritmo B. En este paso se pueden agregar elementos al **P** y eliminar vértices (y sus relaciones) del **G**. A continuación se supone que **V**, **F** y N son definidos por el grafo **G** ya depurado.
3. Para $i = 1, ..., N$ repetir:
4. Seleccionar σ (i)-ésimo vértice $v \in \mathbf{V}$.
5. Verificar si v se puede agregar al grafo sin causar ciclos. Para esto:
6. Para cada par de flechas $w, u \in \mathbf{V'}$ tal que $(w \rightarrow v)$, $(v \rightarrow u) \in \mathbf{F}$ repetir:
7. Verificar que $w \notin \mathbf{A}_u$.
8. Si una de estas pruebas falla, agregar v a **P**.
9. En el caso contrario, agregarlo a **G′**. Para esto:
10. Agregar v a **V′**.
11. Formar $\mathbf{A}_v = \cup \mathbf{A}_u$, donde la unión es por las flechas $u \in \mathbf{V'}$ tales que $(v \rightarrow u) \in \mathbf{F}$.
12. Para cada flecha $w \in \mathbf{V'}$, $(w \rightarrow v) \in \mathbf{F}$ repetir:
13. Para cada vértice $q \in \mathbf{V'}$ tal que $q = w$ ó $w \in \mathbf{A}_q$ repetir:
14. Agregar $\{v\} \cup \mathbf{A}_v$ a $\mathbf{A}_q$.

Ilustración 16. El algoritmo A.

- Los vértices que no tienen flechas entrantes tienen que estar en **P**. Estos vértices —aunque son pocos— se pueden de antemano agregar al **P** y quitar de **G**.
- Los vértices v con un lazo $(v \rightarrow v) \in \mathbf{F}$ también tienen que estar en **P** ya que no pueden estar en **G′**.
- Los vértices que no tienen ninguna flecha entrante o ninguna flecha saliente no pueden estar en **P** (recuérdese que **P** es un

conjunto mínimo). Entonces, estos vértices —en nuestro caso resultaron 20 mil— se pueden de antemano quitar de **G** considerándolos no definidores.

```
1. Para cada vértice con lazo o sin flechas entrantes, repetir:
2.         Agregarlo al P y eliminarlo del G.
3. Mientras los siguientes pasos cambian G repetir:
4.         Para cada vértice que no tiene flechas salientes, o entrantes repetir:
5.                 Eliminarlo de G considerándolo no definidor.
```

Ilustración 17. El algoritmo B.

La eliminación de un vértice puede hacer que otros vértices pierden todas sus flechas entrantes o salientes. El algoritmo que remueve los vértices redundantes del grafo se muestra en la ilustración 17.

El grafo **G** que se obtiene después de la depuración, satisface las siguientes condiciones:

- No tiene lazos.
- Cada vértice tiene tanto flechas entrantes como salientes.

En tal grafo, para cada vértice v existe un conjunto definidor mínimo **P** tal que $v \notin \mathbf{P}$. Generalizando más, para cada conjunto compatible $\mathbf{Q} \subseteq \mathbf{V}$ (diciéndose del conjunto **Q** que es compatible si no existen ciclos en **G** que contengan únicamente los vértices del **Q**) existe un conjunto definidor mínimo **P** tal que $\mathbf{Q} \cap \mathbf{P} = \emptyset$. Esto se comprueba por la aplicación del algoritmo A con un ordenamiento σ que empieza con los elementos del **Q**.

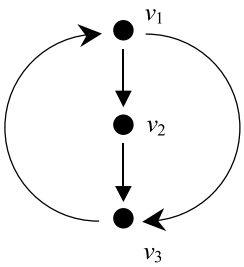


Ilustración 18. Contraejemplo.

Entonces, en tal grafo la característica de ser definidor no es propia del vértice sino sólo es relativa a la selección de un conjunto **P**. Es decir, ningún vértice —salvo los vértices con lazos y los que ya no tuvieron ninguna definición en el diccionario inicial— es un definidor «de por sí», o bien, puede entrar en cualquier conjunto definidor.

Lo contrario no es cierto: en el grafo depurado con el algoritmo B todavía pueden existir los vértices que no entran en ningún conjunto definidor mínimo, véase la ilustración 18, el vértice v_2. Nosotros no tenemos un algoritmo rápido para detectar tales vértices. Sin embargo, nuestros experimentos con el diccionario real mostraron que en este diccionario los vértices de este tipo, si existen, no constituyen más de 20% del grafo depurado, pues encontramos los conjuntos definidores mínimos cuya unión cubre 80% del grafo. Entonces, la detección previa de tales vértices no contribuiría significativamente al rendimiento del algoritmo.

14.3 La metodología experimental

Como se mencionó, el comportamiento del algoritmo depende del ordenamiento de los vértices del grafo. Nosotros no conocemos ningún algoritmo que encuentre el conjunto de menor tamaño posible. Probamos los siguientes ordenamientos.

Método 1: *aleatorio, uniforme*. Usamos el ordenamiento uniformemente aleatorio.

Método 2: *por frecuencias*. Ordenamos los vértices por la frecuencia de uso en las definiciones del mismo diccionario, de menor a mayor. Entonces, los vértices con menor frecuencia tendieron a entrar en **G'** y los que tenían mayor frecuencia tendieron a ser definidores y a entrar en **P**. Esperábamos que con esta heurística **P** sería menor porque los vértices que lo forman rompen más ciclos en **G**.

Método 3: *aleatorio, por frecuencias*. Este método es una combinación de los métodos 1 y 2. Usamos el ordenamiento aleatorio, pero con las probabilidades en función inversa a las frecuencias. Esperábamos que alguna alteración del ordenamiento rígido del método 2 produjera un conjunto menor.

Método 4: *por votación aleatoria*. En este método generamos 20 diferentes conjuntos definidores mínimos $\mathbf{P}_i$ con el método 1, y para cada vértice contamos el número de los conjuntos $\mathbf{P}_i$ en los cuales éste entra, asociando así con cada vértice un peso de entre 0 y 20. Enumeramos primero los vértices con el peso 0 usando sus frecuencias como en el método 2 y después los que entraron, en el orden inverso de su peso, desde 1 hasta 20. Esperábamos que los que entraron en un mayor número de conjuntos $\mathbf{P}_i$ fueran los «mejores» definidores y debieran entrar en el conjunto que buscábamos.

Cabe mencionar, que 80% de los vértices entraron por lo menos en un conjunto $\mathbf{P}_i$.

Hicimos dos experimentos con dos grafos diferentes.

Grafo 1: *por lexemas*. En este experimento consideramos como un vértice del grafo un lexema, es decir, una palabra normalizada morfológicamente: *piensa*, *pensó*, *pensaríamos* se contaron como el nodo *pensar*. No aplicamos ninguna resolución de ambigüedad, asignando las cadenas ambiguas a varios nodos. En este método, las definiciones de todos los significados de la palabra —o, en

su caso, de todos los homónimos de un lema— se consideraron como una sola definición.

Grafo 2: *por significados*. En este experimento, consideramos como un vértice del grafo un significado específico de palabra, por ejemplo: $gato_{1a}$ (animal), $gato_{1b}$ (tipo de animales), $gato_{2a}$ (herramienta), etc. Para desambiguar los significados de las palabras que forman las definiciones empleamos un etiquetador (*tagger*) para la normalización morfológica y desambiguación de la categoría gramatical, y después utilizamos un algoritmo parecido al de Lesk (Sidorov y Gelbukh, 2001) para desambiguar el significado de la palabra en el contexto.

Tabla 6. Número de vértices en los grafos.

Grafo	En total	No definidores	Depurado
Lexemas	30725	20366	**10359**
Significados	60818	47802	**13016**

Tabla 7. Número de definidores, con diferentes algoritmos.

Grafo	Método 1. Aleatoria, uniformemente	Método 2. por frecuencias	Método 3. Aleatoria, por frecuencias	Método 4. Por votación aleatoria
Lexemas	2789, $s = 25$	2302	2770	**2246**
Significados	2266, $s = 28$	1955	2257	**1913**

En ambos casos, sólo se consideraron las palabras significativas, es decir, no se consideraron las preposiciones, conjunciones, verbos auxiliares, etc. Nótese que en el Grafo 2, el número de flechas es exactamente el mismo que el número de palabras significativas en las definiciones del diccionario, mientras que en el Grafo 1 este número es ligeramente mayor por la ambigüedad léxica.

Los resultados de estos experimentos y su discusión se presentan a continuación.

14.4 Resultados y discusión

Para nuestros experimentos usamos el *Diccionario de la Lengua Española* del grupo Anaya. Este diccionario contiene 30971 artículos, de los cuales sólo 30725 corresponden a palabras significativas divididas entre 60818 significados específicos.

La aplicación del algoritmo de depuración —algoritmo B— a las dos variantes del grafo redujo cada uno de éstos a unos 10 mil vértices (véase tabla 6).

Un resultado inesperado fue que este núcleo de unos 10 mil vértices está fuertemente interconectado: prácticamente desde cualquier palabra del diccionario están accesibles todas las demás palabras del núcleo. La tarea de la selección del conjunto definidor en un grafo tan fuertemente interconectado es computacionalmente difícil.

A estos dos conjuntos de 10 mil palabras, les aplicamos el algoritmo *A* con las 4 variantes de ordenamiento. Los resultados se presentan en la tabla 7, mostrando los tamaños de los conjuntos definidores obtenidos.

Para el método 1, se muestra el promedio de los 20 experimentos y la desviación cuadrática promedia *s* (67% de los casos se desvían del promedio no mas que en *s* y 99% y no más que en 3*s*). Atrae la atención que la desviación sea muy baja, lo que significa que con el método 1 los tamaños de los conjuntos obtenidos son diferentes pero muy parecidos.

Para el método 3 sólo mostramos el resultado de un experimento. Fue sorprendente que los resultados con el método 3 casi no diferían de los obtenidos con el método 1, a pesar de que las probabilidades en este caso correspondieron a las frecuencias del método 2.

Éste último mostró un muy buen desempeño produciendo los conjuntos definidores mucho más reducidos. Sin embargo, el método 4 produjo los conjuntos más pequeños que hemos obtenido.

Aunque creemos que con métodos más sofisticados se pueden obtener conjuntos aun menores, no esperamos que el tamaño mínimo del conjunto definidor sea mucho más pequeño que los que hemos obtenido, de aproximadamente 2,000 palabras. Esto se debe a las siguientes consideraciones lingüísticas: según la opinión común, se supone que 2,000 es el número de palabras suficientes para definir todas las demás palabras del vocabulario general. Éste es el tamaño del vocabulario definidor de Longman. Según nuestro conocimiento, éste es el número de los ideogramas en el vocabulario chino básico. Creemos que el hecho de que el tamaño de nuestro conjunto definidor corresponda con tanta exactitud a la cifra esperada —2,000— es muy significativo.

Consideremos unos ejemplos de las palabras elegidas con el Método 4, Grafo 1. Las 20 palabras con la mayor frecuencia de flechas entrantes (las mejores) son:

cosa, *persona*, *acción*, *hacer*, *efecto*, *tener*, *parte*, *no*, *conjunto*, *dar*, *forma*, *cierto*, *cuerpo*, *relativo*, *nombre*, *poder*, *uno*, *formar*, *producir*, *animal*, *común*, *general*, *determinado*, *poner*, *estado*, *tiempo*, *decir*, *planta*, *obra*, etcétera.

Son buenos candidatos a primitivos semánticos.

Otras palabras entraron en el conjunto definidor porque tienen los ciclos cortos, aunque su frecuencia es baja (las peores):

almuerzo, *almuédano*, *almanaque*, *alinear*, *algarroba*, *alarmar*, *ahíto*, etcétera.

Estas palabras son buenos indicadores de la necesidad de cambiar las definiciones en el diccionario, para que se excluyan de la lista de las definidoras.

Finalmente, algunas palabras tienen que estar en cualquier conjunto definidor pues tienen lazos. Éstas indican o bien algún

problema con el algoritmo de identificación de las palabras, o bien una definición errónea en el diccionario. Encontramos 47 casos:

ático, *borgoña*, *lapón*, etcétera.

Es interesante investigar la longitud de los ciclos en los cuales estarían involucradas las palabras definidoras si fuesen insertadas en el diccionario (es decir, estos ciclos consisten sólo de las palabras no primitivas). Ejemplos de tales ciclos son:

1: *ático* → *ático*
2: *premura* → *prisa* → *premura*
3: *grano* → *cereal* → *centeno* → *grano*, etcétera.

En nuestro ejemplo las longitudes de tales ciclos se distribuyeron de la siguiente manera:

L	n	L	n	L	n
1	47	7	53	13	19
2	1496	8	58	14	9
3	177	9	45	15	8
4	67	10	38	16	12
5	47	11	29	17	11
6	72	12	32	18	3

donde L es la longitud del ciclo más corto causado por la palabra dada y n es el número de las palabras con tales ciclos. Sólo mostramos aquí los primeros 18 elementos. El ciclo más largo fue de longitud 52.

14.5 Trabajo futuro

Este capítulo presenta los resultados preliminares de nuestra investigación. Las tareas principales a futuro se agrupan en dos tipos de problemas:

- El problema lingüístico: la interpretación lingüística de los resultados.
- El problema técnico: el diseño de un algoritmo que encuentre un conjunto **P** óptimo en un sentido dado —técnico y/o lingüístico—.

Con mayor detalle, las tareas futuras específicas son las siguientes:

- Dar una interpretación lingüística clara al conjunto **P** obtenido.
- Elaborar criterios lingüísticos que permitirán preferencias en el proceso de inclusión de las palabras en el conjunto definidor (preferir que una palabra sea o no sea primitiva).
- Desarrollar un algoritmo —sea exacto o aproximado— para la construcción de un conjunto **P** del menor tamaño posible.

Sobre esta última tarea, una de las ideas técnicas que vamos a probar es un algoritmo genético para la construcción del conjunto **P** del menor tamaño posible, en forma similar al método 4 de la sección 14.3.

14.6 Conclusiones

Hemos presentado un método para la selección del conjunto mínimo de las palabras a través de las cuales se pueden definir todas las demás palabras en un diccionario explicativo. Este conjunto se denomina conjunto definidor.

Se necesita la construcción de tal conjunto para la conversión del diccionario tradicional en un diccionario semántico computacional orientado a los sistemas de razonamiento lógico automático, siendo un rasgo de tales sistemas lógicos el que no se permiten círculos viciosos en las definiciones.

Nuestro método permitió la construcción de una herramienta que detecta los problemas y defectos relacionados con la presencia de los círculos en las definiciones del diccionario y ayuda al lexicógrafo a corregirlos.

Queda para la investigación futura la interpretación lingüística del hecho de que en el diccionario con el cual experimentamos encontramos casi exactamente el número esperado de palabras primitivas —dos mil.

Bibliografía

(Aguirre and Rigau, 1996) Aguirre, E. and G. Rigau. Word Sense Disambiguation using Conceptual Density. In: *Proc. 16th international conference* COLING. Copenhagen, 1996.

(Alexandrov y Gelbukh, 1999) Alexandrov, M., A. Gelbukh. Measures for determining thematic structure of documents with Domain Dictionaries. In: *Proc. Text Mining workshop at 16th International Joint Conference on Artificial Intelligence (IJCAI'99)*, Stockholm, Sweden, 1999, pp. 10–12.

(Alexandrov *et al*., 1999) Alexandrov, M., P. Makagonov, and K. Sboychakov. Searching similar texts: some approaches to solution. Borsevich (ed.), *Acta Academia*, *Annual J. Intern. Inform. Academy*, Chisinau, Moldova, 1999, pp. 215–223.

(Alexandrov *et al*., 2000a) Alexandrov, M., A. Gelbukh, and P. Makagonov. Evaluation of Thematic Structure of Multidisciplinary Documents. In: *Proc.* DEXA*-2000, 11th International Conference and Workshop on Database and Expert Systems Applications, NLIS-2000, 2nd International Workshop on Natural Language and Information Systems*, England, 2000. IEEE Computer Society Press, pp. 125–129.

(Alexandrov *et al*., 2000b) Alexandrov, M., A. Gelbukh, P. Makagonov. On Metrics for Keyword-Based Document Selection and Classification. In: *Proc. CICLing-2000, International Conference*

on Intelligent Text Processing and Computational Linguistics, February, Mexico City, 2000, pp. 373–389.

(Alexandrov *et al*., 2001) Alexandrov, Mikhail, Alexander Gelbukh, George Lozovoi. Chi-square Classifier for Document Categorization. *Lecture Notes in Computer Science,* N 2004, Springer, 2001, pp. 455–457.

(Álvarez, 1977) Álvarez Constantino, J. *Gramática funcional de español*, Editorial Avante, 1977.

(Anaya, 1996) Grupo Anaya. *Diccionario de la lengua española*. 1996, www.anaya.es.

(Aone y McKee, 1993) Aone, Ch., and D. McKee. Language-independent anaphora resolution system for understanding multilingual texts. In: *Proceedings of the 31st meeting of the* ACL. The Ohio State University, Columbus, Ohio, 1993.

(Apresjan, 1974) Apresjan, J. D. Regular polysemy, *Linguistics*, 1974, 142: 5–32.

(Apresjan, 1995) Apresjan, J. D. *Selected works* (in Russian). Moscow, 1995, V 1, 472 p., V 2, 768 p.

(Ariel, 1988) Ariel, M. Referring and accessibility. *Journal of Linguistics*, 1988, 24: 67–87.

(Atserias *et. al*, 1998) Atserias, J., J. Carmona, I. Castellón, S. Cervell, M. Civit, L. Márquez, M.A. Martí, L. Padró, R. Placer, H. Rodriguez, M. Taulé, J. Turmó. Morphosyntactic analisis and parcing of unsrestricted Spanish texts. In: *Proc of* LREC*-98*, 1998.

(Baddorf y Evans, 1998) Baddorf, D. S. and M. W. Evens. Finding phrases rather than discovering collocations: Searching corpora for dictionary phrases. In: *Proc. of the 9th Midwest Artificial Intelligence and Cognitive Science Conference (*MAICS*'98),* Dayton, USA, 1998.

(Baeza-Yates y Ribeiro-Neto, 1999) Baeza-Yates, Ricardo, and Berthier Ribeiro-Neto. *Modern information retrieval*. Addison-Wesley Longman, 1999.

(Banerjee y Pedersen, 2002) Banerjee, Satanjeev, and Ted Pedersen. An adapted Lesk algorithm for word sense disambiguation

using WordNet. *Lecture Notes in Computer Science,* N 2276, Springer, 2002, pp. 136–145.

(Bank of English) *Bank of English*. Collins. http://titania.cobuild.collins.co.uk/boe_info.html.

(Basili *et al*., 1993) Basili, R., M. T. Pazienza, and P. Velardi. Semi-automatic extraction of linguistic information for syntactic disambiguation. *Applied Artificial Intelligence*, 7:339–64, 1993.

(Beesley y Karttunen, 2003) Beesley, K. B. and L. Karttunen. *Finite state morphology*. CSLI publications, Palo Alto, CA, 2003.

(Benson *et al*., 1986) Benson, M., E. Benson, and R. Ilson. *The BBI Combinatory dictionary of English*. John Benjamins Publishing Co., 1986.

(Biber *et al*., 1998) Biber, D., S. Conrad, and D. Reppen. *Corpus linguistics. Investigating language structure and use.* Cambridge University Press, Cambridge, 1998.

(Biber, 1993) Biber, D. Representativeness in corpus design. *Literary and linguistic computing*, 1993, 8:243–257.

(Bider y Bolshakov, 1976) Bider, I. G. and I. A. Bolshakov. Formalization of the morphologic component of the Meaning – Text Model. 1. Basic concepts (in Russian with a separate translation to English). *ENG. CYBER. R*., No. 6, 1976, pp. 42–57.

(Bolshakov y Gelbukh, 1998) Bolshakov, I. and A. Gelbukh. Lexical functions in Spanish. In: *Proc. CIC-98, Simposium Internacional de Computación*, November 11–13, Mexico D.F., 1998, pp. 383–395.

(Bolshakov y Gelbukh, 2000) Bolshakov, I. A. and A. Gelbukh. A Very Large Database of Collocations and Semantic Links. *Lecture Notes in Computer Science,* No. 1959, Springer Verlag, 2001, pp. 103–114.

(Bolshakov y Gelbukh, 2001) Bolshakov, I. A. and A.F. Gelbukh. A Large Database of Collocations and Semantic References: Interlingual Applications. *International Journal of Translation*, Vol.13, No.1–2, 2001.

(Bolshakov y Gelbukh, 2002) Bolshakov, I. A. and A. Gelbukh. Word Combinations as an Important Part of Modern Electronic Dictionaries. *Revista «Procesamiento de lenguaje natural»*, No. 29, septiembre 2002, pp. 47–54.

(Bolshakov y Gelbukh, 2003) Bolshakov, I. A. and A. Gelbukh. On Detection of Malapropisms by Multistage Collocation Testing. In: *Proc. NLDB-2003, 8th International Workshop on Applications of Natural Language to Information Systems*, June 23–25, 2003, Burg, Germany.Bonner Köllen Verlag, pp. 28–41.

(Bolshakov y Gelbukh, 2004) Bolshakov, I. A. and A. F. Gelbukh. *Computational Linguistics and Linguistic Models*. Series Lectures in Computational Linguistics. Colección en Ciencia de Computación. IPN – UNAM – Fondo de Cultura Económica, 2004, 187 pp.

(Bolshakov, 1994) Bolshakov, I. A. Multifunction thesaurus for Russian word processing. In: *Proc. 4th Conference on Applied Natural language Processing*, Stuttgart, 1994, pp. 200–202.

(Bolshakov, 2002) Bolshakov, I. A. Surface Syntactic Relations in Spanish. *Lecture Notes in Computer Science,* N 2276, Springer Verlag, 2002, pp. 210–219.

(Bosch, 1988) Bosch, P. Representing and accessing focussed referents. *Language and Cognitive Processes*, 3, 1988, pp. 207–231.

(Campos, 2001) Campos, L. M. de. *Un modelo de recuperación de información basado en redes bayesianas*. Universidad de Granada, España, 2001.

(Canals-Marote, R. *et al*., 2001) Canals-Marote, R. *et al*. El sistema de traducción automática castellano <—> catalán interNOSTRUM Alacant. Revista «*Procesamiento de Lenguaje Natural*» N 27, 2001, pp. 151–156.

(Carreras *et al*., 2004) Carreras, X., I. Chao, L. Padró and M. Padró. FreeLing: An Open-Source Suite of Language Analyzers. In: *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*. Lisbon, Portugal. 2004.

(Carter, 1987) Carter, D. *Interpreting anaphora in natural language texts*. Ellis Horwood, Chichester, 1987.

(Chafe, 1976) Chafe, W. Giveness, Contrastiveness, Definiteness, Subject, Topics, and Point of View. In: Ch. N. Li (Ed.), *Subject and Topic*. Academic Press, New York, 1976, pp. 27–55.

(Chafe, 1987) Chafe, W. Cognitive Constraints in Information Flow. In: R. Tomlin (Ed.), *Coherence and Grounding in Discourse.* Benjamins, Amsterdam, 1987, pp. 21–51.

(Chafe, 1994) Chafe, W. *Discourse, Consciousness, and Time*. The University of Chicago Press, Chicago – London, 1994, 327 p.

(Church *et al*., 1991) Church, K., W. Gale, P. Hanks, and D. Hindle. Parsing, word associations and typical predicate-argument relations. In: M. Tomita (Ed.), *Current Issues in Parsing Technology*. Kluwer Academic, Dordrecht, Netherlands, 1991.

(Cornish, 1996) Cornish, F. 'Antecedentless' anaphors: deixis, anaphora, or what? Some evidence from English and French. *Journal of Linguistics,* 1996, 32: 19–41.

(Cowan, 1995) Cowan, R. What are Discource Principles Made of? In: P. Downing and M. Noonan (Eds.), *Word Order in discource*. Benjamins, Amsterdam/Philadelfia, 1995.

(Cowie *et al*., 1992) Cowie, J., L. Guthrie, and G. Guthrie. Lexical disambiguation using simulated annealing. In: *Proceedings of Coling-92*, Nante, France, 1992, pp. 359-365.

(Dagan *et al*., 1999) Dagan, I., L. Lee, and F. Pereira. Similarity-based models of word cooccurrence probabilities. *Machine Learning*, 34(1), 1999.

(Dolan et al., 2000) Dolan, W., L. Vanderwende, and S. Richardson. Polysemy in a Broad-Coverage Natural Language Processing System. In: *Polysemy: Theoretical and Computational Approaches*. Y. Ravin and C. Leacock (eds). Oxford University Press. New York, 2000. pp 178-204.

(Downing y Noonan, 1995) Downing, P., and M. Noonan (Eds.). *Word Order in discourse*. Benjamins, Amsterdam/Philadelfia, 1995, 595 p.

(Elliott *et al*., 2000) Elliott J, Atwell, E, and Whyte B. Language identification in unknown signals. In: *Proc. of COLING'2000*, ACL and Morgan Kaufmann Publishers, 2000, pp. 1021–1026.

(Erku y Gundel, 1987) Erku, F., and J. K. Gundel. The pragmatics of indirect anaphors. In J. Verschueren and M. Bertuccelli-Papi (Eds.), *The pragmatic perspective: Selected papers from the 1985 International Pragmatics Conference*. John Benjamins, Amsterdam, 1987, pp. 533–545.

(Evens, 1988) Evens, M. N. (ed.), *Relational models of lexicon: Representing knowledge in semantic network*. Cambridge: Cambridge University Press, 1988.

(Fellbaum, 1990) Fellbaum, C. The English verb lexicon as a semantic net. *International Journal of Lexicography*, 1990, 3: 278–301.

(Fellbaum, 1998) Fellbaum, Ch. (ed.) *WordNet*: *an Electronic Lexical Database*. MIT Press, 1998.

(Fox, 1987) Fox, B. A. *Discourse structure and anaphora: written and conversational English*. Cambridge University Press, Cambridge, 1987.

(Frakes y Baeza-Yates, 1992) Frakes, W., and R. Baeza-Yates (Eds.) *Information Retrieval: Data Structures and Algorithms*. Prentice-Hall, 1992.

(Fraurud, 1992) Fraurud, K. *Processing noun phrases in natural discourse*. Doctoral dissertation, Stockholm University, Stockholm, 1992.

(Fraurud, 1996) Fraurud, K. Cognitive ontology and NP form. In T. Fretheim and J. K. Gundel (Eds.), *Reference and referent accessibility*. John Benjamins, Amsterdam, 1996, pp. 193–212.

(Fretheim and Gundel, 1996) Fretheim, T., and J. K. Gundel (Eds.). *Reference and referent accessibility*. John Benjamins, Amsterdam, 1996.

(Ghazfan, 1996) Ghazfan D. Toward meaningful Bayesian networks for information retrieval systems. In: *Proceedings of the IPMU´96 Conference*, 1996, pp. 841-846.

(Galicia Haro *et al*., 2001) Galicia Haro, S., A. Gelbukh, and I. A. Bolshakov. Una aproximación para resolución de ambigüedad estructural empleando tres mecanismos diferentes. *J. Procesamiento de Lenguaje Natural*, No 27, Septiembre 2001.

(Galicia-Haro *et al*., 2001) Galicia-Haro, S., A. Gelbukh, and I. A. Bolshakov. Acquiring syntactic information for a government pattern dictionary from large text corpora. In: *Proc. IEEE SMC-2001: Systems, Man, And Cybernetics. Tucson*, USA, 2001, IEEE, pp. 536–542.

(Garrido-Alenda y Forcada, 2001) Garrido-Alenda, Alicia, and Mikel L. Forcada. MorphTrans: un lenguaje y un compilador para especificar y generar módulos de transferencia morfológica para sistemas de traducción automática. *J. Procesamiento de Lenguaje Natural* N 27, 2001, pp. 151–156.

(Gel'bukh, 1992) Gel'bukh, A.F. Effective implementation of morphology model for an inflectional natural language. *J. Automatic Documentation and Mathematical Linguistics*, Allerton Press, vol. 26, N 1, 1992, pp. 22–31.

(Gelbukh y Sidorov, 1999) Gelbukh, A. and G. Sidorov. On Indirect Anaphora Resolution. In: *Proc. PACLING-99, Pacific Association for Computational Linguistics*, University of Waterloo, Waterloo, Ontario, Canada, August 25–28, 1999, pp. 181–190.

(Gelbukh y Sidorov, 2000) Gelbukh, A. y G. Sidorov. Sistema inteligente de recuperación de textos políticos de la base de datos documental del Senado de la República Mexicana. En: *Memorias CIC-2000, Congreso Internacional de Computación*, Noviembre 15–17, 2000, pp. 315–321.

(Gelbukh y Sidorov, 2001) Gelbukh, A. and G. Sidorov. Zipf and Heaps Laws' Coefficients Depend on Language. *Lecture Notes in Computer Science,* N 2004, Springer, 2001, pp. 330–333.

(Gelbukh y Sidorov, 2002a) Gelbukh, A. y G. Sidorov. Recuperación de documentos con comparación semántica suave. In: *Proc. TAINA-2002, Workshop on Soft Computing at MICAI'2002:*

2nd Mexican International Conference on Artificial Intelligence, Merida, Mexico, April 2002, pp. 253–261.

(Gelbukh y Sidorov, 2002b) Gelbukh, A. y G. Sidorov. Selección automática del vocabulario definidor en un diccionario explicativo. *J. Procesamiento de Lenguaje Natural*, No 29, September 2002. Sociedad Española para el Procesamiento de Lenguaje Natural (SEPLN), España, pp. 55–64.

(Gelbukh y Sidorov, 2003a) Gelbukh, A. and G. Sidorov. Approach to construction of automatic morphological analysis systems for inflective languages with little effort. *Lecture Notes in Computer Science*, N 2588, Springer, pp. 215–220.

(Gelbukh y Sidorov, 2003b) Gelbukh, A. y G. Sidorov, Hacia la verificación de diccionarios explicativos asistida por computadora. Revista *Estudios de lingüística aplicada*, UNAM, 21 (38), 2003, pp. 89–108.

(Gelbukh *et al*., 1998) Gelbukh, A., I. Bolshakov, S. Galicia Haro. Automatic Learning of a Syntactical Gobernment Patterns Dictionary from Web-Retrieved Texts. In: *Proc. Int. Conf. on Automatic Learning and Discovery*, Pittsburgh, USA, pp. 261–267, 1998.

(Gelbukh *et al*., 1999a) Gelbukh, A., G. Sidorov, A. Guzman-Arenas. Use of a weighted topic hierarchy for document classification. Václav Matoušek *et al*. (Eds.). Text, Speech and Dialogue. 2nd International Workshop TSD-99, Plzen, Czech Republic, 1999. Lecture Notes in Artificial Intelligence, N 1692, Springer, pp. 130–135.

(Gelbukh *et al*., 1999b) Gelbukh, A., G. Sidorov, and A. Guzmán-Arenas. A Method of Describing Document Contents through Topic Selection. In: *Proc. SPIRE'99, International Symposium on String Processing and Information Retrieval*, Cancun, Mexico, September 22–24, IEEE Computer Society Press, 1999, pp. 73–80.

(Gelbukh *et al*., 1999c) Gelbukh, A., G. Sidorov, A. Guzmán-Arenas. Document comparison with a weighted topic hierarchy.

In: *Proc. DEXA-99, 10th International Conference and Workshop on Database and Expert Systems Applications, DAUDD'99, 1st International Workshop on Document Analysis and Understanding for Document Databases*, Florence, Italy, September 1, 1999, IEEE Computer Society Press, pp. 566–570.

(Gelbukh *et al*., 2002a) Gelbukh, A., G. Sidorov, and L. Chanona-Hernández. Compilation of a Spanish representative corpus. *Lecture Notes in Computer Science,* N 2276, Springer, 2002, pp. 285–288.

(Gelbukh *et al*., 2002b) Gelbukh, A., G. Sidorov, S. Galicia Haro, I. Bolshakov. Environment for Development of a Natural Language Syntactic Analyzer. *Acta Academia 2002*, Moldova, 2002, pp. 206–213.

(Gelbukh *et al*., 2002c) Gelbukh, A., G. Sidorov, and A. Guzmán-Arenas. Relational Data Model in Document Hierarchical Indexing. *Lecture Notes in Computer Science*, N 2389, Springer, 2002, pp. 259–262.

(Gelbukh *et al*., 2003a) Gelbukh, A., G. Sidorov, SangYong Han, and L. Chanona-Hernandez. Automatic evaluation of quality of an explanatory dictionary by comparison of word senses. *Lecture Notes in Computer Science*, N 2890, Springer, 2003, pp. 555–561.

(Gelbukh *et al*., 2003b) Gelbukh, A., G. Sidorov, y F. Velásquez. Análisis morfológico automático en español a través de generación. Revista «*Escritos*», 28, 2003, pp. 9–26.

(Gelbukh *et al*., 2004) Alexander Gelbukh, Grigori Sidorov, San-Yong Han, and Erika Hernández-Rubio. Automatic enrichment of very large dictionary of word combinations on the basis of dependency formalism. *Lecture Notes in Computer Science,* N 2972, 2004, Springer, pp. 430–437.

(Gelbukh, 1997) Gelbukh, A.F. Using a semantic network for lexical and syntactical disambiguation. In: *Proc. CIC-97, Nuevas Aplicaciones e Innovaciones Tecnológicas en Computación, Simpo-*

sium Internacional de Computación, Mexico City, Mexico, 1997, pp. 352–366.

(Gelbukh, 2000a) Gelbukh, A. A data structure for prefix search under access locality requirements and its application to spelling correction. In: *Proc. of MICAI-2000: Mexican International Conference on Artificial Intelligence*, Acapulco, Mexico, 2000.

(Gelbukh, 2000b) Gelbukh, A. Lazy Query Enrichment: A Simple Method of Indexing Large Specialized Document Bases. *Lecture Notes in Computer Science*, N 1873, Springer, 2000, pp. 526–535.

(Gelbukh, 2003) Gelbukh, A. A data structure for prefix search under access locality requirements and its application to spelling correction. *Computación y Sistemas, Revista Iberoamericana de Computación*, vol. 6, N 3, 2003, pp. 167–182.

(Global Reach, 2002) *Global Reach*, 2002. http://global-reach.biz

(González y Vigil, 1999) González, B. M. y C. Ll. Vigil. *Los Verbos Españoles*. 3ª Edición, España, Ediciones Colegio de España, 1999, 258 p.

(Gundel *et al*., 1988) Gundel, J., N. Hedberg, and R. Zacharski. Giveness, Implicature and Demonstrative Expressions in English Discource. *Proceedings of 25th meeting of Chicago Linguistic Society, Part II (Parasession on Language in Context),* Chicago, 1998, pp. 89–103.

(Gusfield, 1997) Gusfield, Dan. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997.

(Hahn *et al*., 1996) Hahn, U., M. Strube, and K. Markert. Bridging textual ellipses. *Proceedings of the 16th International Conference on Computational Linguistics*, 1996, pp. 496–501.

(Hartmann, 2001) Hartmann, R.R.K. *Teaching and researching lexicography*. Pearson Education Limited, 2001, 211 p.

(Hausser, 1999a) Hausser, R. Three Principled Methods of Automatic Word Form Recognition. *Proc. of VEXTAL: Venecia per*

il Tratamento Automatico delle Lingue. Venice, Italy, 1999. pp. 91–100.

(Hausser, 1999b) Hausser, Roland. *Foundations of Computational linguistics*. Springer, 1999, 534 p.

(Hellman, 1996) Hellman, C. The 'price tag' on knowledge activation in discourse processing. In: T. Fretheim and J. K. Gundel (Eds.), *Reference and referent accessibility*. John Benjamins, Amsterdam, 1996.

(Hirst y Budanitsky, 2003) Hirst, G., and A. Budanitsky. Correcting Real-Word Spelling Errors by Restoring Lexical Cohesion. *Natural Language Engineering*, 11(1), March 2005, pp. 87–111.

(Hirst, 1981) Hirst, G. *Anaphora in Natural Language Understanding*. Springer Verlag, Berlin, 1981.

(Indirect anaphora, 1996) *Indirect Anaphora Workshop*. Lancaster University, Lancaster, 1996.

(Jiang y Conrad, 1999) Jiang, J.J. and D.W. Conrad. From object comparison to semantic similarity. In: *Proc. of Pacling-99 (Pacific association for computational linguistics)*, August, 1999, Waterloo, Canada, pp. 256–263.

(Jiménez-Salazar, 2003) Jiménez-Salazar, H. A Method of Automatic Detection of Lexical Relationships using a Raw Corpus. *Lecture Notes in Computer Science,* N 2588, Springer, 2003, pp. 325–328.

(Kameyama, 1997) Kameyama, M. Recognizing Referential Links: an Information Extraction Perspective. In: *Proceedings of ACL'97/ EACL'97 workshop on Operational factors in practical, robust anaphora resolution*. Madrid, 1997.

(Karov y Edelman, 1998) Karov, Ya. and Sh. Edelman. Similarity-based word-sense disambiguation. *Computational linguistics*, 1998, Vol. 24, pp. 41–59.

(Karttunen, 2003) Karttunen, L. Computing with realizational morphology. *Lecture Notes in Computer Science*, N 2588, Springer, pp. 203–214.

(Kilgariff, 2001) Kilgariff, A. Web as corpus. In: *Proc. of Corpus Linguistics 2001 conference*, University center for computer corpus research on language, technical papers vol. 13, Lancaster University, 2001, pp. 342–344.

(Kim *et al*., 2001) Kim, S., J. Yoon, and M. Song. Automatic extraction of collocations from Korean text. *Computers and the Humanities* 35 (3): 273–297, August 2001, Kluwer Academic Publishers.

(Kita *et al*., 1994) Kita, K., Y. Kato, T. Omoto, and Y. Yano. A comparative study of automatic extraction of collocations from corpora: Mutual information vs. cost criteria. *Journal of Natural Language Processing*, 1(1):21–33, 1994.

(Koskenniemi, 1983) Koskenniemi, K. *Two-level Morphology: A General Computational Model for Word-Form Recognition and Production*. University of Helsinki Publications, N 11, 1983.

(Kowalski, 1997) Kowalski, G. *Information Retrieval Systems Theory and Implementation*, Kluwer Academic Publishers, 1997.

(Kozima y Furugori, 1993) Kozima, H. and T. Furugori. Similarity between words computed by spreading activation on an English dictionary. In: *Proceedings of the 6 conference of the European chapter of* ACL, 1993, pp. 232–239.

(Kukich, 1992) Kukich, K. Techniques for automatically correcting words in texts. ACM *Computing Surveys*, 1992, 24(4), pp. 377–439.

(Lambrecht, 1994) Lambrecht, K. *Information Structure and Sentence Form. Topic, Focus and the Mental Representation of Discource Referents.* Cambridge University Press, Cambridge, 1994, 388 p.

(Landau, 2001) Landau, S. *Dictionaries: The art and craft of lexicography*. Cambridge University Press, 2001, 477 p.

(Lawrence, 2000) Lawrence, S. *El Acceso a la Información en la Web Limitado y Desigual*. NEC Research Institute, 2000, http://www.neci.nec.com/.

(LDOCE) *LDOCE* (*Longman Dictionary of Contemporary English*). Longman; www.longman.com/dictionaries/which_dict/ldocenew.html.

(Ledo-Mezquita *et al*., 2003) Ledo-Mezquita, Yoel, Grigori Sidorov, Alexander Gelbukh. Tool for Computer-Aided Spanish Word Sense Disambiguation. *Lecture Notes in Computer Science*, N 2588, Springer, 2003, pp. 277–280.

(Lesk, 1986) Lesk, M. Automatic sense disambiguation using machine-readable dictionaries: how to tell a pine cone from an ice cream cone. In: *Proceedings of ACM SIGDOC Conference*, Toronto, Canada, 1986, pp. 24–26.

(Levenshtein, 1966) Levenshtein, V. I.. Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*, 1966, 10(8), pp. 707–710.

(Llorac, 2000) Llorac, E. *Gramática de la Lengua Española*. España, Ed. Espasa, 2000, 406 p.

(Makagonov *et al*., 2000) Makagonov, P., M. Alexandrov, K. Sboychakov. A toolkit for development of the domain-oriented dictionaries for structuring document flows. In: H.A. Kiers *et al*. (Eds.) *Data Analysis, Classification, and Related Methods*, Studies in classification, data analysis, and knowledge organization, Springer, 2000, pp. 83–88.

(Malkovsky, 1985) Malkovsky, M. G. *Dialogue with an artificial intelligence system* (in Russian). Moscow State University, Moscow, Russia, 1985, 213 p.

(Manning y Schütze, 1999) Manning, C. D. and Schütze, H. *Foundations of statistical natural language processing*. Cambridge, MA, The MIT press, 1999, 680 p.

(McEnery y Wilson, 1996) McEnery, T. and A. Wilson. *Corpus linguistics*. Edinburg University Press, 1996.

(McHale, 1997) McHale, M. L. *A comparison of WordNet and Roget's taxonomy for measuring semantic similarity*, 1997.

(McRoy, 1992) McRoy, S. Using multiple knowledge sources for word sense disambiguation. *Computational Linguistics*, 1992, Vol. 18(1), pp. 1–30.

(Mel'čuk, 1988) Mel'čuk, I. A. *Dependency Synax: Theory and Practice*. The State University of New York Press, Albany, New York, 1988, 428 p.

(Mel'čuk, 1996) Mel'čuk, I. Phrasemes in language and phraseology in linguistics. In: *Idioms: structural and psychological perspective*, 1996, pp. 167–232.

(Mel'čuk, 1999) Mel'čuk, I. A. *Communicative Organization in Natural Language: The Semantic-Communicative Structure of Sentence*, 1999, 380 p.

(Mihalcea and Moldovan, 1999) Mihalcea, R. and D. Moldovan. A Method for word sense disambiguation of unrestricted text. In: *Proc 37th Annual Meeting of the* ACL, Maryland, USA, 1999, pp. 152-158.

(Mitkov, 1997) Mitkov, R. Factors in Anaphora Resolution: They are not the Only Things that Matter. A Case Study Based on Two Different Approaches. In: *Proc. of the ACL'97/EACL'97 workshop on Operational factors in practical, robust anaphora resolution*. Madrid, 1997.

(Montes y Gómez *et al*., 1999) Montes y Gómez, M., A. López López, and A. Gelbukh. Extraction of document intentions from titles. In: *Proc. of Text Mining workshop at 16th International Joint Conference on Artificial Intelligence (IJCAI'99)*, Stockholm, Sweden, July 31 – August 6, 1999, pp. 101–102.

(Montes y Gómez *et al*., 2001) Montes y Gómez, M., A. Gelbukh, A. López López y R. Baeza-Yates. Minería de texto empleando grafos conceptuales. *J. Procesamiento de Lenguaje Natural*, No 27, Septiembre 2001.

(Montes y Gómez *et al*., 2001a) Montes y Gómez, M, A. Gelbukh, A. López López and R. Baeza-Yates. Flexible Comparison of Conceptual Graphs. *Lecture Notes in Computer Science,* N 2113, Springer, 2001, pp. 102–111.

(Montes y Gómez *et al*., 2001b) Montes y Gómez, M., A. Gelbukh, A. López López, and R. Baeza-Yates. Text mining with conceptual graphs. In: *Proc of International IEEE SMC-2001 Conference: Systems, Man, And Cybernetics*. Tucson, USA, October 7–10, 2001, IEEE, pp. 898–903.

(Montes y Gómez *et al*., 2001c) Montes y Gómez, M., A. Gelbukh, and A. López López. Mining the news: trends, associations, and deviations. *Computación y Sistemas, Revista Iberoamericana de Computación*, Vol. 5 N 1, 2001, pp. 14–24.

(Montoyo, 2001) Montoyo, A. *Método basado en Marcas de Especificidad para WSD*. Grupo de Procesamiento del Lenguaje y Sistemas de Información. Universidad de Alicante, España, 2001.

(Moreno y Goñi, 1995) Moreno, A. and J. Goñi. GRAMPAL: A Morphological Processor for Spanish Implemented in PROLOG. In: M. Sessa and M. Alpuente (Eds.), *Proceedings of the Joint Conference on Declarative Programming (GULP-PRODE'95)*, Marina di Vietri (Italia), 1995, pp. 321–331.

(OALD) *OALD* (*Oxford Advanced Learner's Dictionary*). Oxford University Press, www1.oup.co.uk/elt/oald.

(OCD, 2003) *Oxford collocation dictionary*, Oxford, 2003.

(Ozhegov, 1990) Ozhegov, S. I. *Diccionario explicativo del idioma ruso.* (en ruso) edición 22a, Moscú, Rusia, 1990.

(Partee y Sgall, 1996) Partee, B., and P. Sgall (Eds.). *Discource and Meaning. Papers in Honour of Eva Hajičova.* Benjamins, Amsterdam/Philadelphia, 1996.

(Pazos *et al*., 2002) Pazos R., Rodolfo A., A. Gelbukh, J. Javier González, E. Alarcón, A. Mendoza, P. Domínguez. Spanish Natural Language Interface for a Relational Database Querying System. *Lecture Notes in Artificial Intelligence*, N 2448, Springer, 2002, pp. 123–130.

(Pimienta, 2000) Pimienta, D. Representación de las lenguas y culturas latinas en la Internet. Fundación Redes y Desarrollo. *Encuentro "Sociedad y Tecnología"*, Santiago de Chile, 2000.

(Rasmussen, 1992) Rasmussen E. Clustering algorithms. In: Frakes, W. B. and R. Baeza-Yates. *Information Retrieval: Data Structures and Algorithms*. Prentice Hall, Upper Saddle River, NJ, 1992, pp. 419–442.

(Ravin y Leacock, 2000) Ravin, Ya. and C. Leacock. Polysemy: an overview. In: *Polysemy: Theoretical and Computational Approaches*. Ya. Ravin and C. Leacock (eds). Oxford University Press. New York, 2000, pp. 1-29.

(Resnik, 1995) Resnik, Ph. (1995). Disambiguating noun groupings with respect to WordNet senses. In: *Proc. Third Workshop on Very Large Corpora*. Cambridge, MA, 1995, pp. 54-68.

(Resnik, 1999) Resnik, Ph. Semantic similarity in a taxonomy an information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, (11) 1999, pp 95-130.

(Ribeiro, 1996) Ribeiro, B. A belief network model for IR. In: *Proceedings of the 19th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*. SIGIR´96, Zurich, ACM, 1996, pp 253-260.

(Rigau *et al.*, 1997) Rigau, G., J. Atserias and E. Aguirre. Combining Unsupervised Lexical Knowledge Methods for Word Sense Disambiguation. In: *Proc 35th annual Meeting of the ACL*, Madrid, Spain, 1997, pp 48-55.

(Sag y Wasow, 1999) Sag, I. A., T. Wasow. *Syntactic theory: Formal Introduction*. CSLI Publ., Stanford. University of Chicago Press, Chicago & London, 1999.

(Saint-Dizier and Viegas, 1995) Saint-Dizier, P. and Viegas, E. (eds.) *Computational lexical semantics*. Cambridge: Cambridge University Press, 1995, 447 p.

(Sanford *et al.*, 1983) Sanford, A. J., S. C. Garrod, A. Lucas, and R. Henderson. Pronouns without explicit antecedents? *Journal of Semantics*, 1983, 2: 303–318.

(Santana *et al*., 1999) Santana, O., J. Pérez, *et al*. FLANOM: Flexionador y Lematizador Automático de Formas Nominales. *Lingüística Española Actual* XXI, 2. Ed. Arco/Libros, S.L. España, 1999.

(Saracevic, 1995) Saracevic, T. *A taxonomy of values for library and information services*. Rutgers University, New Brunswick, 1995.

(Sedlacek y Smrz, 2001) Sedlacek, R. and P. Smrz, A new Czech morphological analyzer AJKA. *Lecture Notes in Computer Science,* N 2166, Springer, 2001, pp. 100–107.

(Shank *et al*., 1980) Shank, R. C., M. Lebowitz, and L. Birnbaum. An Integrated Understander. *American Journal of Computational Linguistics*, 1980, 6 (l): 13–30.

(Sidorov y Gelbukh, 2001) Sidorov G. and A. Gelbukh. Word sense disambiguation in a Spanish explanatory dictionary. In: *Proc. of TALN-2001* (*Tratamiento automático de lenguaje natural*), Tours, France, July 2–5, 2001, pp. 398–402.

(Sidorov, 1996) Sidorov, G. O. Lemmatization in automatized system for compilation of personal style dictionaries of literature writers (in Russian). In: *Word by Dostoyevsky*, Moscow, Russia, Russian Academy of Sciences, 1996, pp. 266–300.

(Sierra y Alarcón, 2002) Sierra, G. and R. Alarcón. Recurrent patterns in definitory context. *Lecture Notes in Computer Science,* N 2276, Springer, 2002, pp. 438–440.

(Sierra y McNaught, 2000) Sierra, G., and J. McNaught. Analogy-based Method for Semantic Clustering. In: *Proc. CICLing-2000, International Conference on Intelligent Text Processing and Computational Linguistics*, February, Mexico City, 2000, pp. 358–372.

(Sierra y McNaught, 2003) Sierra, G., and J. McNaught. Natural Language System for Terminological Information Retrieval. *Lecture Notes in Computer Science,* N 2588, Springer, 2003, pp. 543–554.

(Singleton, 2000) Singleton, D. *Language and the lexicon: an introduction*. Arnold Publishers, 2000, 244 p.

(Smadja *et al*., 1996) Smadja, F., K. R. McKeown, and V. Hatzivassiloglou. Translating collocations for bilingual lexicons: A statistical approach. *Computational Linguistics*, 22(1):1–38, 1996.

(Smadja, 1993) Smadja, F. Retrieving collocations from texts: Xtract. *Computational linguistics*, 19 (1):143–177, 1993.

(Sproat, 1992) Sproat, R. *Morphology and computation*. Cambridge, MA, MIT Press, 1992, 313 p.

(Steele, 1990) Steele, J. (ed.) *Meaning – Text Theory. Linguistics, Lexicography, and Implications*. University of Ottawa press. 1990.

(Stetina *et al*., 1998) Stetina J., S. Kurohashi and M. Nagao. General word sense disambiguation method based on full sentencial context. In: *Usage of WordNet in Natural Language Processing. COLING-ACL Workshop*, Montreal, Canada, 1998.

(Strzalkowski *et al*., 1999) Strzalkowski, T., Fang Lin, Jin Wang, and J. Perez-Carballo. Evaluating natural language processing techniques in information retrieval. In: T. Strzalkowski (ed.) *Natural language information retrieval*. Kluwer, 1999, pp. 113–146.

(Sussna, 1993) Sussna, M. Word sense disambiguation for free-text indexing using a massive semantic network. In: *Proc. Second International CIKM*, Airlington, 1993, pp. 67-74.

(Tomlin, 1987) Tomlin, R. (ed.). *Coherence and Grounding in Discource*. Benjamins, Amsterdam, 1987, 512 p.

(Turtle and Croft, 1990) Turtle H. and W. B. Croft. Inference networks for document retrieval. In: *SIGIR´90, 13th International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Brussels, Belgium, 5-7 September 1990, pp. 1-24.

(Voorhees, 1993) Voorhees, E. M. Using WordNet to disambiguate word senses for text retrieval. In: *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Pittsburgh, Pennsylvania, 1993, pp. 171–180.

(Vossen, 2001) Vossen, P. Condensed meaning in EuroWordNet. In: P. Boillon and F. Busa (Eds), *The language of word meaning*. Cambridge University Press, 2001, pp. 363–383.

(Ward y Birner, 1994) Ward, G., and B. Birner. Definiteness and the English existential. *Language*, 1994, 71: 722–742.

(Wierzbicka, 1980) Wierzbicka, A. *Lingua Mentalis: The semantics of natural language*. New York: Academic Press, 1980.

(Wierzbicka, 1996) Wierzbicka, A. *Semantics: Primes and Universals*. Oxford: Oxford University Press, 1996.

(Wilks *et al*., 1993) Wiks, Y., D. Fass, C. Guo, J. McDonal, T. Plate and B. Slator. Providing Machine Tractable dictionary tools. In: J. Pustejowsky, (Ed.) *Semantics and the lexicon*, 1993, pp. 341-401.

(Wilks and Stevenson, 1996) Wilks, Y. and M. Stevenson. *The grammar of sense: Is word-sense tagging much more than part-of-speech tagging?* Technical Report CS-96-05, University of Sheffield, 1996.

(Wilks and Stevenson, 1998) Wilks, Y. and M. Stevenson (1998), Word sense disambiguation using optimized combination of knowledge sources. In*: Proceedings of ACL 36/Coling 17*, 1998, pp. 1398-1402.

(Wilks y Stevenson, 1999) Wilks, Y. and M. Stevenson. Combining weak knowledge sources for sense disambiguation. In: *Proceedings of IJCAI-99*, 1999, pp. 884–889.

(WordNet, 1998) *WordNet: an electronic lexical database*. C. Fellbaum (ed.), MIT, 1998, 423 p.

(Yarowksy, 1992) Yarowksy, D. Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In: *Proceeding of Coling-92*, Nante, France, 1992, pp. 454–460.

(Yu *et al*., 2003) Yu, J., Zh. Jin, and Zh. Wen. *Automatic extraction of collocations*. 2003.

(Yule, 1982) Yule, G. Interpreting anaphora without identifying reference. *Journal of Semantics*, 1982, 1: 315–322.

(Zgusta, 1971) Zgusta, L. *Manual of lexicography*. Hague: Mouton, Prague: Academia, 1971.

(Zipf, 1949) Zipf, G. K. *Human behavior and the principle of least effort.* Cambridge, MA, Addison-Wesley, 1949.

Índice analítico

A

acento fonético 90
acento gráfico 91
agrupamiento 60
algoritmo de Lesk 190
alternación en raíz..... 148
ambigüedad 76
ambigüedad de sentidos
de palabras 185
AMPLN - *Véase* Asociación
Mexicana para
el Procesamiento de
Lenguaje Natural
anafor 169
análisis a través
de generación 143
análisis morfológico 133
análisis sintáctico..... 155
antecedente 169
antonimia..... 123
árbol de dependencias..... 159
árbol sintáctico 159
Asociación Mexicana para
el Procesamiento .....
de Lenguaje Natural..... 83

B

biblioteca digital 68
buscador..... 41
búsqueda inteligente..... 49

C

CFG - *Véase* gramática
independiente de contexto
CICLING *Véase* Congreso
Internacional de Lingüística
Computacional y
Procesamiento Inteligente
de Texto
clasificación 60
coherencia, verificación de.. 169

Congreso Internacional de Lingüística Computacional y Procesamiento Inteligente de Texto 83
conocimiento extralingüístico 78
conocimiento lingüístico 78
contexto 157, 250
contraposición 134
corpus 4
corpus representativo 229
correferencia 98, 169
directa 169
indirecta 170

D

DCG - *Véase* gramáticas de cláusulas definidas
déixis 134
desambiguación de sentidos de palabras 183
desambiguación morfológica 200
diálogo 75
diccionario 73
de atracción léxica 124
de combinaciones de palabras 247
de subcategorización 247
FACTOTUM 142
morfológico 247
sintáctico 247
discurso 98

E

e-Gobierno 72
escenario prototípico 179
estilo, verificación de 40
estructuración de información 45
extracción de información 69, 83

F

filtrado de información 69
fonética 93
fonología 94
formalidad 25

G

generación de resúmenes 70
generación de texto 67
género morfológico 146
grafo conceptual 44
gramática 95
de constituyentes 95
de dependencias 95
gramática de adjunción de árboles 157
gramática de cláusulas definidas 140

gramática independiente
de contexto 99
gramática, verificación
de 39, 121
guiones, división con 39

H

habla 93
homonimia 105
HPSG 157

I

información tabular 51
interfaz en lenguaje natural . 159
Internet 72, 229

L

*Lecture Notes in Computer
Science* 83
lengua
eslava 135
románica 135
lenguaje
aglutinativo 135
flexivo 135
LEXESP, corpus 151
lexicografía 96
lexicología 96
ley de Zipf 212
lingüística 23
lingüística computacional 25
lingware 160

M

malapropismo 122
meronimia 123
método de Lesk 112, 190
minería de texto 70
modelo 89
modelo de conjugación
verbal 147
modelo de dos niveles 139
morfología 94,133
morfonología 93
motor de búsqueda 46

N

navegación 48
nivel de lenguaje 87
número gramatical 94

O

ortografía, verificación de 121

P

paradigma morfológico 142
paráfrasis 106
PAROLE, estándar 151
parser 95, 155

performativa, expresión 97
persona gramatical 135
pertinencia 197
PLN - *Véase* procesamiento de lenguaje natural
pluralia tantum 149
polisemia 105
polisemia regular 106
pragmática 97
precisión 162
precisión de búsqueda 58
primitivas semánticas ... 104, 267
procesador lingüístico 99
procesamiento de lenguaje natural 26, 37
psicolingüística 23

R

razonamiento lógico 44
recall 162
recuperación de información 41, 197
red semántica 96, 100, 101
representación de documento 26

S

semántica 96
sentidos de palabra 97,110
significado 174
sinonimia 123
sinónimos 43, 203
sintaxis 92, 95
sistema
AGME 136
Clasitex 179
FreeLing 134
GRAMPAL 140
MACO+ 134
PC-KIMMO 138
sociolingüística 23

T

TAG - *Véase* gramática de adjunción de árboles
tecnologías de lenguaje natural 69
traducción automática 63

V

verbo irregular 147
verbo regular 147
verbo semiregular 147
vocabulario definidor 96, 104, 268
voz 74

Procesamiento automático del español
con enfoque en recursos léxicos grandes

Alexander Gelbukh y Grigori Sidorov

Impreso en los talleres gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras, 27, Centro Histórico, México, DF,
Septiembre 2010
Edición 1000 ejemplares

Laura Rocío Ramos Osorno
Diseño editorial y portada