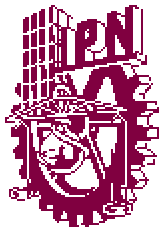


INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

LABORATORIO DE LENGUAJE NATURAL

DETERMINACIÓN AUTOMÁTICA DE ROLES SEMÁNTICOS USANDO PREFERENCIAS DE SELECCIÓN SOBRE CORPUS MUY GRANDES

TESIS QUE PARA OBTENER EL GRADO DE

DOCTOR EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA

FRANCISCO HIRAM CALVO CASTRO

DIRECTOR DE TESIS:

ALEXANDER GELBUKH

MÉXICO, D.F.

MAYO DE 2006

Agradecimientos

A todos los que han sido mis maestros en la vida:

A mis padres

A mi asesor Alexander Gelbukh

A mi alma mater, IPN

Este trabajo ha sido posible gracias al apoyo del gobierno de México:
CONACYT, PIFI-IPN, SNI, CGPI-IPN, COFAA-IPN y gracias al apoyo de RITOS-2.

Índice

Glosario

Actos del habla: el filósofo inglés J. L. Austin, elaboró en los años sesenta una teoría que se conoce como *Teoría de los Actos de Habla*, en ella propuso que hablar no es solamente "informar" sino también "realizar" algo. La propuesta fue conocida a través de su libro (publicado por primera vez en 1962) *How to do things with words*. Su postura iba en contra de las aproximaciones más tradicionales que veían al lenguaje en función de la mera transmisión de información. Se centró en el estudio de los que denominó verbos "performativos" como prometer, demandar, jurar, acusar, etc. Para Austin, el acto de habla tiene tres niveles, o se realiza a través de tres actos conjuntos, el acto locutivo, que consiste meramente en enunciar la frase en cuestión, el acto ilocutivo, que consiste en llevar a cabo algo a través de las palabras (prometer, amenazar, jurar, declarar) y el acto perlocutivo que consiste en provocar un cambio en el estado de cosas o una reacción en el interlocutor. Muchos investigadores han continuado trabajando con la teoría de actos de habla. El más destacado ha sido un discípulo de Austin, John Searle.

Actualizador: tipo de determinante que sitúa al núcleo del sintagma nominal en el espacio y en el tiempo con mayor o menor precisión, transformándolo de desconocido en conocido o prestándole concreción. Existen tres subtipos: los **artículos**, los **demostrativos** y los **posesivos**.

Ambigüedad: El lenguaje natural es inherentemente ambiguo a diferentes niveles: A **nivel léxico**, una misma palabra puede tener varios significados, y la selección del apropiado se debe deducir a partir del contexto oracional o conocimiento básico. Muchas investigaciones en el campo del procesamiento de lenguajes naturales han estudiado métodos de resolver las ambigüedades léxicas mediante diccionarios, gramáticas, bases de conocimiento y correlaciones estadísticas. **A nivel referencial**, la resolución de anáforas y catáforas implica determinar la entidad lingüística previa o posterior a que hacen referencia. **A nivel estructural**, se requiere de la semántica para desambiguar la dependencia de los sintagmas preposicionales que conducen a la construcción de distintos árboles sintácticos. Por ejemplo, en la frase: *Rompió el dibujo de un ataque de nervios*. **A nivel pragmático**, una oración, a menudo, no significa lo que realmente se está diciendo. Elementos tales como la ironía tienen un papel importante en la interpretación del mensaje.

Artículos: antaño denominados artículos determinados, presentan el núcleo del sintagma nominal, esto es, lo transforman de desconocido en conocido situándolo en el lugar y el tiempo de la enunciación, o sacándolos del pensamiento abstracto para situarlos en la situación ilocutiva. En castellano son los artículos masculinos *el* y *l* de las formas de artículo contracto con preposición *al* y *del* en singular y *los* en plural; los femeninos *la* y *el* (ante vocal a acentuada, aunque podían ser a y e átonas en la lengua del *Cantar de Mio Cid*) en singular y *las* en plural, y los neutros singulares *lo* y *el*, que se usan para sustantivar adjetivos (metábasis de sustantivación).

Atributo: función de la sintaxis tradicional. Es un sintagma que acompaña a los verbos copulativos (ser, estar y parecer en español) y que se refiere al mismo tiempo al sujeto, con el que concuerda en género y número. En ocasiones se puede sustituir por el pronombre "lo".

Aumentativo: denominación que recibe a los sufijos que aumentan el significado de las palabras, o en casos de objetos dan el significado de ser grandes. Consiste en agregar una raíz al final de la palabra, en idioma español son numerosos y existen varios que se usan dependiendo la palabra. A diferencias de los diminutivos no hay diferencias en los países hispanohablantes sobre su uso.

Barbarismo, incorrección que consiste en pronunciar o escribir mal las palabras, o en emplear vocablos impropios.

Campo semántico: conjunto de palabras de la misma categoría que poseen un núcleo de significación común (sema compartido) y se diferencian por una serie de rasgos o semas distinguidores.

Caso dativo: se aplica a sustantivos y pronombres. Este caso marca normalmente el complemento indirecto, por lo que sirve para expresar la persona o cosa que recibe el daño o provecho de la acción verbal. Así que responde a las preguntas: «¿a quién?» o «¿para quién?», formuladas al verbo. Por ejemplo: el niño escribe una carta *a su padre*. Pero además existen otros usos como el de posesión, como por ejemplo en latín vulgar y, en menor medida, el latín clásico. Su nombre viene del latín *datiuus*, del verbo *dare*, dar.

Caso genitivo: El Genitivo es uno de los casos que proceden del latín. El Genitivo es también denominado como posesivo, aunque no hay que confundirlo con el caso posesivo, debido a su corriente uso para denotar esa relación. Sin embargo, esta segunda denominación se queda corta ya que el genitivo no solo cubre relaciones de posesión.

Caso nominativo: es un caso que se aplica a sintagmas nominales. El nominativo, normalmente, marca el sujeto de una oración. El caso nominativo ya se empleaba en latín y en inglés antiguo, entre otras lenguas. Bastantes lenguas flexivas utilizan el nominativo como la forma *normal* de la palabra. Es decir, aquella que se recoge en el diccionario. Sin embargo, esto no ocurre en todas las lenguas flexivas, por ejemplo, el sánscrito a menudo cita los nombres empleando sólo el lexema, sin añadir los sufijos. Por ejemplo *ásva-* para «caballo», y no *ásvas*.

Caso vocativo: caso que se emplea para identificar el nombre al que se dirige el hablante. Se encuentra en latín, **polaco** y entre otras lenguas. Cuando se utiliza un vocativo, el elemento a quien se dirige el hablante se expone directamente. Por ejemplo, en la oración, «No te entiendo, Juan», *Juan* es un vocativo que indica el receptor del mensaje, o persona a quien el hablante se dirige. Algunas lenguas, (por ejemplo, el griego) tienen un caso nominal vocativo particular. En latín, la marca de vocativo de un nombre se corresponde con la marca de nominativo, excepto en el caso de los sustantivos masculinos singulares de segunda declinación. Un buen ejemplo es la famosa pregunta de Julio César, «*Et tu, Brute?*» («¿Y tú, Brutus?», que se entiende comúnmente como «¿Tú también, Brutus?»), donde «*Brute*» es el vocativo, mientras que «Brutus» sería el caso nominativo. Cuando traducimos nombres latinos al español, solemos emplear el nominativo. En español utilizamos algunas expresiones que sirven para marcar el vocativo, como por ejemplo «¡**Oye** muchacho!, ¿dónde está tu hermano?», pero las expresiones vocativas castellanas son interjecciones (*¡Oh!* y *¡Ay!*, acompañadas de un nombre) o verbos en imperativo, no declinaciones del nombre, o se recurre a aislar el nombre entre pausas marcadas en la escritura con comas.

Caso: flexión de una palabra, típicamente un sustantivo, adjetivo o pronombre, que adopta para mostrar determinadas relaciones gramaticales. Se realiza mediante sufijos en algunas lenguas procedentes del indoeuropeo, tales como el latín, griego, sánscrito, las lenguas eslavas y en menor medida en las lenguas germánicas. En las lenguas romances, la flexión de caso se ha perdido casi por completo y sólo permanece en el sistema pronominal.

Categoría sintáctica: El término **categoría sintáctica** se utiliza con sentidos diferentes en la literatura y en la lingüística. En ocasiones se refiere a los conceptos que se expresan mediante los morfemas flexivos (género, número, persona, tiempo, aspecto, etc.), pero lo más frecuente es que se refiera a las partes de la oración con función sintáctica o sintagmas.

Cconjunciones copulativas: sirven para reunir en una sola unidad funcional dos o más elementos homogéneos e indican su adición. Son: *y*, *e*, *ni*, *que*. Y es la conjunción más usada en la lengua coloquial: *Sergio 'y' Daniel pasean*; se repite frecuentemente en el lenguaje infantil, como expresión sucesiva de enunciados: El perro es mi amigo 'y' lo quiero mucho 'y' juega conmigo. Este uso pleonástico se mantiene en la lengua popular de las narraciones, y como recurso expresivo intensificador. Se emplea *e* cuando la palabra siguiente empieza por *i* o *hi*, para evitar la cacofonía: *Se reunieron 'e' hicieron los trabajos*. *Vinieron los padres 'e' hijos*. La conjunción *ni* equivale a *y no* y señala la adición de dos términos, pero implica que sean negativos: *No hizo los trabajos 'ni' estudió*. A fin de marcar la expresividad, se antepone a veces a todos los términos unidos: *'Ni' tengo trabajo 'ni' dinero*. La conjunción copulativa *que* es de uso arcaizante, aunque también figura en locuciones con valor intensificador: *Y tú llora 'que' llora*. *Lo mismo da* que *da lo mismo*.

Coma: La **coma(,)** es un signo de puntuación que señala una breve pausa que se produce dentro del enunciado.

Condicional: sujeto a condiciones. Término que indica que una situación se puede dar si se cumplen ciertos supuestos. Originalmente se llamaba "Modo Potencial" (forma hipotética o posible). Corresponde a un futuro hipotético de la forma verbal. **Ejemplos:** *Ganaría* la lotería si acertase todos los números, *Saldría* con ella si nos quisiéramos, ¿Me *mirarías* a los ojos al menos una vez en tu vida?

Conjunción: palabra o conjunto de ellas que enlaza proposiciones, sintagmas o palabras, como su etimología de origen latino explica: cum, 'con', y jungo, 'juntar'; por lo tanto, 'que enlaza o une con'. Constituye una de las clases de nexos. No debe confundirse con los marcadores del discurso. La conjunción es una parte invariable de la lengua que se utiliza para enlazar oraciones y establecer relaciones entre ellas: *Luisa va a trabajar* ***y*** *Pedro se queda en casa*. Hay otros muchos nexos, en su origen preposiciones, que encabezan oraciones y que adquieren valor de conjunción, aunque no tengan forma conjuntiva. A estas construcciones se les llama **giros conjuntivos**. Por ejemplo:**Al + inf.** = **Cuando + verbo conjugado**: Al cantar el gallo, San Pedro lloró = Cuando cantó el gallo... + **Por + inf.** = **Porque + verbo conjugado**: *Por venir tarde, no entró = Porque vino tarde...* + **Con + inf.** = **Aunque + verbo conjugado**: Con ser tan listo, no aprobó = Aunque era tan listo... + **De + inf.** = **Si + verbo conjugado**: De

llover hoy, nos refugiaremos en el kiosko = Si llueve hoy, nos refugiaremos... + **Para + inf.** = **Para que + verbo conjugado**: Hemos venido para cantar = Hemos venido para que cantemos +

Conjunciones adversativas: contraponen dos oraciones o términos sintácticos. La contrariedad puede ser parcial o total; la parcial expresa una corrección o restricción en el juicio de la primera oración, de modo que la coordinación es restrictiva: *mas*, *pero*, *aunque*. Existe una serie de conjunciones que proceden de formas lingüísticas más extensas y que se han gramaticalizado total o parcialmente que se usan como nexos adversativos: *sin embargo*, *empero*, *con todo*, *a pesar de*, *no obstante*, *más bien*, *excepto*, *salvo*, *menos*...

Conjunciones distributivas: indican distribución o alternancia; repiten los términos: *o*... *o*; se emplean a veces unidades de tipo adverbial: *bien*... *bien*, *ya*... *ya*, *ora*... *ora* también se usa la forma verbal inmovilizada *sea*, cuando los términos unidos expresan equivalencia: 'Ya' vienes, 'ya' te quedas.

Conjunciones disyuntivas: indican alternancia exclusiva o excluyente: *o*, *u*, se coloca entre los términos que indican la alternancia o antepuesta a cada uno de ellos: Llamó Pedro *o* Juan. Se emplea *u* cuando precede a una palabra iniciada por *o* u *ho*: Lo hará uno 'u' otro, también para evitar la cacofonía. Otras veces, *o* indica que los términos unidos son equivalentes y sirven para designar una misma realidad: Todo ocurrió 'o' sucedió en un momento.

Conjunciones explicativas: unen proposiciones que expresan lo mismo, pero de distinta forma, a fin de explicarse mutuamente. Son por lo general giros aislados entre comas como *o sea*, *esto es*, *es decir*, *mejor dicho*, *id est*, *es más*: *Se fue al otro mundo, es decir, se murió*.

Conjunciones impropias: enlazan oraciones dependientes, como son las locuciones o partículas subordinantes: cómo, cuándo, que, porque, para que... Las conjunciones subordinantes degradan la oración en que se insertan y la transponen funcionalmente a una unidad de rango inferior que cumple alguna de las funciones propias del sustantivo, del adjetivo o del adverbio: *Dijo **que** vendría*. *Lo hizo **porque** quiso*.

Conjunciones propias: unen oraciones o elementos del mismo nivel sintáctico, grupo nominal o adjetivo, como son las conjunciones coordinantes o coordinativas: *y*, *ni*, *pero*, *sino*...: *Luis caminaba triste **y** pensativo*.

Conjunciones subordinantes o subordinativas que introducen subordinadas sustantivas: introducen oraciones que desempeñan las funciones propias de un sintagma nominal (sujeto, atributo, complemento directo, complemento indirecto, suplemento, complemento del nombre). Las conjunciones sustantivas se clasifican según la función que la oración sustantiva desempeñe dentro de la oración principal. Se utiliza *que*, conjunción completiva, para la función de sujeto y de complemento directo: *Me molestó 'que' no me lo dijeras*; *Dijo 'que' lo haría*. A veces, se emplea *que* con alguna preposición, por ejemplo en función de suplemento: *Él se convenció 'de que' era importante*. También se emplea si para las interrogativas indirectas: "Me pregunto *si* vendrá". También pueden utilizarse pronombres y adverbios interrogativos: "Me preguntó *cómo* vendrían". "Me preguntó *cuántos* vendrían".

Connotación: está en función de determinadas experiencias y valores asociados al significado. De esta forma, mientras que "perro" y "chucho" denotan el mismo significado, sus connotaciones son muy diferentes. La connotación varía según a quien se le sugiera. De tal forma, la palabra "pacifista" tiene distintas connotaciones en la jerga militar y en un grupo de "*hippies*".

Corpus lingüístico: es un conjunto, normalmente muy amplio, de ejemplos reales de uso de una lengua. Estos ejemplos pueden ser textos (típicamente), o muestras orales (normalmente transcritas).

Cuantificadores: determinantes que miden el núcleo del sintagma nominal. Son de dos tipos: los que miden de forma precisa o **numerales**, y los que miden de forma imprecisa o cuantificadores **extensivos**, también llamados indefinidos.

Deixis de lugar: expresión deíctica que sitúa un participante en el espacio e indica cercanía o lejanía, como por ejemplo «aquí, allí, ahí».

Deixis de tiempo: referente temporal en relación con un momento en particular que suele ser el instante en que se articula el mensaje.

Deixis personal: expresión deíctica que se refiere al papel que desempeña un participante. Estas deixis pueden ser de primera, segunda o tercera persona. Algunos ejemplos de deixis de primera persona son los siguientes pronombres y determinantes «yo, nosotros, nuestro, mi, mío, míos».

Deixis social: expresión deíctica que se refiere a un participante. Puede tener una función distintiva en la relación social, como por ejemplo la expresión de cortesía «usted» en contraste con la expresión «tú».

Deixis: expresión que se emplea para referirse a algún asunto extralingüístico cuya interpretación puede variar dependiendo de determinados factores que forman parte del contexto extralingüístico.

Demostrativos: tipo de actualizadores que sitúan en el espacio y en el tiempo de forma más precisa que el artículo los núcleos de sintagma nominal. En castellano son *este*, *esta*; *ese*, *esa*; *aquel*, *aquella* y sus respectivos plurales. *Este esta* sitúa en el espacio y el tiempo más próximo al hablante; *ese esa* en el espacio y tiempo más próximo al oyente, y *aquel aquella* en el espacio y el tiempo más alejando tanto como para oyente como para el hablante. Por eso no podemos decir "este día de ayer" o "aquel día de hoy" ni "aquella tiza que tengo en la mano ahora mismo", por ejemplo.

Denotación: básicamente es la relación entre una palabra y aquello a lo que se refiere.

Determinante: función sintáctica desempeñada por diversos tipos de partículas que acompañan en castellano al núcleo del sintagma nominal situándose delante de él para especificarlo. Existen fundamentalmente cuatro tipos: **predeterminantes**, **actualizadores**, **cuantificadores** e **interrogativo-admirativos**.

Determinantes interrogativo-admirativos: son aquellos que preguntan por el núcleo del sintagma nominal o expresan admiración por el mismo: ¡Qué libro! ¿Qué libro?. Son *qué*, *cuál-es*, *cuánto-a-s*.

Diccionario de idiomas: son los diccionarios en que se indican las palabras equivalentes en otro idioma o en otros idiomas. Es habitual encontrar este tipo de diccionario en un mismo tomo junto con el idioma inverso, de tal forma que pueden consultarse las palabras en ambos idiomas.

Diccionario de la lengua: en ellos se explica brevemente el significado de las palabras de una lengua, y se proporcionan al mismo tiempo los datos principales gramaticales, como el género de la palabra (masculino, femenino o neutro) o el plural.

Diccionario de sinónimos y antónimos: en estos diccionarios se relacionan palabras de significado similar y opuesto, para facilitar la elección de éstas al redactar textos. Los más sencillos se limitan a dar una lista de palabras para cada entrada, pero algunos más completos indican además las diferencias de matiz con la palabra buscada, sin llegar a ser un tesauro, comentado más adelante.

Diccionario etimológico: son los diccionarios en los que se facilita información sobre el origen de las palabras de una determinada lengua. Quizá, el diccionario etimológico más prestigioso es el *Oxford English Dictionary*.

Diccionario: obra de consulta de palabras y media términos que se encuentran generalmente ordenados alfabéticamente. La disciplina que se encarga de elaborar diccionarios es la Lexicografía. La información que proporciona varía según el tipo de diccionario del que se trata.

Diccionarios de dudas: recogen palabras y frases cuyo significado se ha desvirtuado y no significan en la sociedad lo que un diccionario de la lengua indica. Estos diccionarios ayudan a un redactor o escritor a usar los términos correctos y no dejarse llevar por el significado popular. A diferencia del diccionario de uso práctico anterior, su objetivo no es dar a conocer el uso vulgar de una palabra, sino advertir de éste, y proponer alternativas adecuadas.

Diccionarios de gramática: en estos diccionarios no se ordenan palabras, sino estructuras gramaticales. Su uso principal es para personas que están aprendiendo un idioma extranjero, ya que les permite buscar estructuras gramaticales de un texto y consultar en ellos su significado y construcción.

Diccionarios de uso práctico: recogen acepciones en las palabras que no son reconocidas por el órgano competente (como la Real Academia de la Lengua en España) pero que sin embargo, se usan en la sociedad.

Diccionarios especializados: estos diccionarios están dedicados a palabras o términos que pertenecen a un campo determinado como, por ejemplo, informática, jardinería, lenguaje SMS, pesos y medidas o abreviaturas, y proporcionan una breve información sobre el significado de tales palabras o términos. Pueden ser también diccionarios de idiomas en los que se indica la traducción a otra lengua o a otras lenguas de las palabras o términos que incluyen.

Diccionarios inversos o de rimas: son diccionarios de la lengua con la particularidad de que están ordenados alfabéticamente según las últimas letras de cada palabra, en vez de las primeras. Su uso principal es buscar palabras

que rimen con otra, para la redacción de poesías y versos. Algunos diccionarios inversos reducidos no incluyen definiciones, sólo la lista de palabras ordenadas de esta forma.

Estructuralismo: enfoque de las ciencias humanas que creció hasta convertirse en uno de los métodos más utilizados para analizar el lenguaje, la cultura y la sociedad en la segunda mitad del siglo XX. El término, sin embargo, no se refiere a una escuela de pensamiento claramente definida, aunque la obra de Ferdinand de Saussure es considerado habitualmente como un punto de origen. El estructuralismo puede ser visto como un enfoque general con un cierto número de variantes. Sus influencias y desarrollos son complejos.

Extracción de la información *(IE)*: tipo de Recuperación de la información cuyo objetivo es extraer automáticamente información estructurada o semiestructurada desde documentos legibles por la máquina. Una aplicación típica de *IE* es el escaneado de una serie de documentos escritos en una lengua natural y rellenar una base de datos con la información extraída. Tendencias actuales en relación con la *(IE)* utilizan técnicas de Procesamiento de lenguaje natural que se centran en áreas muy restringidas. Por ejemplo, la *Message Understanding Conference (MUC)*, o **Conferencia para la Comprensión de Mensajes** es una competición que se ha centrado en los siguientes aspectos durante los últimos años: MUC-1 (1987), MUC-2 (1989) Mensajes para operaciones navales. MUC-3 (1991) Terrorismo en países latinoamericanos. MUC-5 (1993) Microelectrónica. MUC-6 (1995) Nuevos artículos a cerca de los cambios en la gerencia. MUC-7 (1998) Informes de lanzamiento de satélites. Otras táreas típicas de la *IE* son: Reconocimiento de nombres de personas, organizaciones, lugares, expresiones temporales y ciertas expresiones numéricas. Coreferencialidad: identificar distintos sintagmas nominales que se refieren al mismo objeto. La anáfora es un tipo de coreferencialidad.

Fonemas: unidades teóricas básicas del nivel fónico del lenguaje humano, que tienen una función distintiva: son sonidos del habla que permiten distinguir palabras en una lengua. Así, los sonidos /p/ y /b/ son fonemas del español porque existen palabras como /pata/ y /bata/ que tienen significado distinto y su pronunciación sólo difiere en relación con esos dos sonidos.

Flexión: alteración que experimentan las palabras, usualmente mediante afijos o desinencias, para expresar sus distintas funciones dentro de la oración y sus relaciones de dependencia o de concordancia con otras palabras o elementos oracionales. La conjugación y la declinación son formas de flexión. Cuando los afijos o desinencias se añaden directamente a la raíz se da la **flexión radical** y cuando son añadidos al tema se da la **flexión temática**.

Funcionalismo: escuela que sigue métodos y estudios fundamentados en una interpretación funcional de la lengua.por lo que el ser humano esta dado.

Generación de Lenguajes Naturales (GLN): proceso de la construcción de un texto en lenguaje natural para la comunicación con fines específicos. Texto se refiere aquí a un término general y repetitivo aplicable a expresiones, o partes de ellas, de cualquier tamaño, tanto habladas como escritas. En el ser humano, el que sea hablado o escrito tiene consecuencias en el nivel deliberativo y de edición que ha tenido lugar; si el lenguaje es hablado puede faltar revisión ya que la mayoría de los programas actuales pueden hablar, si bien casi todos sólo presentan palabras en una pantalla. La decisión de revisar o usar la palabra escrita o hablada no es una opción para la generación del programa en la actualidad; pero se debe abordar el tema en el diseño de un programa en particular. El principal énfasis de la generación de lenguajes naturales no es sólo el facilitar el uso del ordenador sino también el desarrollar una teoría computacional de la capacidad del lenguaje humano. En este sentido constituye una herramienta para extender, aclarar y verificar teorías que se han formulado en lingüística, psicología y sociología acerca de la comunicación entre humanos. Un generador de lenguaje natural típicamente tiene acceso a un gran conjunto de conocimiento del cual ha de seleccionar información para presentar a los usuarios en varias formas. El generar texto es, pues, un problema de toma de decisiones con múltiples restricciones: de conocimiento proposicional, de herramientas lingüísticas disponibles, de los objetivos de la comunicación del usuario a quien se dirige el texto, y de la situación y del discurso pasado. Se trata de identificar los factores involucrados en este proceso y de determinar la mejor forma de representar estos factores y sus dependencias.

Gramática sistémico funcional: modelo gramatical desarrollado por Michael Halliday. Este modelo ha sido utilizado por Richard Hudson para desarrollar la *Word Grammar*.

Gramática Transformacional: amplio término usado para describir gramáticas, casi exclusivamente aquellas que se refieren a lenguas naturales que han sido desarrolladas en la tradición chomskiana. Este término es normalmente sinónimo del ligeramente más específico *Gramática Generativa Transformacional*.

Gramática Universal: teoría lingüística de la escuela transformacional y generativa que afirma que subyacen determinados principios comunes a todas las lenguas naturales. En esta teoría se dice que estos principios son innatos dentro de nuestra condición humana. Esta teoría no afirma que todas las lenguas naturales tengan la misma gramática, o que todos los humanos estén "programados" con una estructura que subyace bajo todas las expresiones de lenguas humanas. Sino que afirma que hay una serie de reglas que ayudan a los niños a adquirir su lengua materna. Quienes estudian la gramática universal tienen el propósito de conseguir abstraer generalizaciones comunes a diversos idiomas, a menudo de la siguiente forma: "Si X es cierto, entonces Y ocurre". Este estudio se ha extendido a numerosas disciplinas lingüísticas, tales como la fonología y la Psicolingüística. Tres lingüistas que han tenido una influencia considerable en este área, ya sea directamente o mediante la escuela que han promovido, son Noam Chomsky, Edward Sapir y Richard Montague.

Gramática: es el estudio de la lengua, en cuanto a forma, estructura, y significado.

Holonimia: noción semántica que se opone a meronimia, del mismo modo en que se oponen el todo y la parte. Así, por ejemplo, *BICICLETA* es un holónimo mientras que *sillín*, *pedal*, *aro* y *manubrio* son merónimos. A diferencia de la relación "hiperonimia / hiponimia", que también distingue dos conceptos de distinto nivel, la relación "**holonimia** / **meronimia**" no es tanto de inclusión conceptual cuanto de inclusión material. En efecto, en la oposición "FLOR / Rosa, clavel, nardo", el hiperónimo (FLOR) es una categoría más abarcante que incluye entre sus miembros a la rosa, al clavel y al nardo, entre otras flores. En cambio, en la oposición "CASA / dormitorio, comedor, cocina", el holónimo nombra al todo que incluye materialmente a las partes (dormitorio, comedor, etc.).

Homónimos: palabras que tienen significados diferentes pero se escriben igual. Un ejemplo es *banco* (para estar sentado / de finanzas). Lo contrario son sinónimos. Palabras que tienen varios significados y el mismo origen también se llaman polisémicas.

Implicaturas: significados adicionales que el receptor de un mensaje infiere cuando el emisor parece estar violando una de las máximas del principio cooperativo.

Inferencia: acto que debe ser realizado por el receptor del mensaje (oyente, lector,...) para interpretar correctamente la referencia. Las palabras en sí no refieren, sino que el que refiere es quien las emplea.

Interpretación: En líneas generales puede entenderse como interpretación la reformulación oral de algo pronunciado en otro idioma. Los intérpretes distinguen la interpretación de la traducción, que se ocupa de la palabra escrita. Las interpretaciones se pronuncian, las traducciones se escriben. En España, antiguamente, a los intérpretes se les llamaba **lenguas**.

Lenguaje: conjunto de símbolos que en conjunto nos dejan transmitir un mensaje, y es una capacidad exclusiva del ser humano (los animales tienen sistemas de comunicación) que lo capacita para abstraer, conceptualizar y comunicarse. Los humanos creamos un número infinito de oraciones a partir de un número finito de elementos y también recreamos la lengua por ejemplo a través de esquemas y/o mapas conceptuales. La representación de dicha capacidad es lo que conocemos como lengua o idioma, es decir el código.

Léxico: puede significar una lista de palabras junto con otra información adicional (es decir, un diccionario), la palabras utilizadas en una región específica, las palabras de un idioma, o incluso de un lenguaje de programación. Léxico es una palabra de origen griego (λεξικόν) que significa vocabulario. Cuando los lingüistas estudian el léxico, estudian qué son las palabras, cómo se conforma el vocabulario de un idioma y su estructura, cómo las personas utilizan y memorizan palabras, cómo aprenden palabras, la historia y evolución de las palabras, relaciones y tipos de relaciones entre palabras, así como el proceso de creación de palabras. Cuando una palabra no pertenece al léxico de un lenguaje de programación, por otra parte, es correcto decir que estamos ante un error.

Lexicografía: ciencia que se ocupa de estudiar como los signos forman palabras válidas.

Lingüística computacional: campo multidisciplinario de la lingüística y la informática que utiliza la informática para estudiar y tratar el lenguaje humano. Para lograrlo, intenta modelar de forma lógica el lenguaje natural desde un punto de vista computacional. Dicho modelado no se centra en ninguna de las áreas de la lingüística en particular, sino que es un campo interdisciplinario, en el que participan lingüistas, informáticos especializados en inteligencia artificial, psicólogos cognoscitivos y expertos en lógica, entre otros. Algunas de las áreas de estudio de la lingüística computacional son: Corpus lingüístico asistido por ordenador; Diseño de analizadores sintácticos (en inglés: *parser*), para lenguajes naturales; Diseño de etiquetadores o lematizadores (en inglés: *tagger*), tales como el *POS-tagger*; Definicion de lógicas especializadas que sirvan como fuente para el Procesamiento de Lenguajes Naturales; Estudio de la posible relación entre lenguajes formales y naturales.

Lingüística: ciencia que estudia el lenguaje y sus fenómenos asociados.

Locuciones prepositivas: precisan algunos aspectos que las preposiciones existentes matizan mal: *acerca de*, *al lado de*, *alrededor de*, *antes de*, *a pesar de*, *en pos de*, *cerca de*, *con arreglo a*, *con objeto de*, *debajo de*, *delante de*, *dentro de*, *después de*, *detrás de*, *encima de*, *en cuanto a*, *enfrente de*, *en virtud de*, *frente a*, *fuera de*, *gracias a*, *merced a*, *junto a*, *lejos de*, *por culpa de*, *respecto a*, etc... Estas preposiciones preceden necesariamente a un sintagma nominal. En el caso de las preposiciones "a" y "de" ante el artículo determinado masculino singular "el" forman las contracciones o artículos contractos "al" y "del" respectivamente.

Meronimia: relación semántica. Un merónimo es el nombre atribuido a un constituyente que forma parte de, que es substancia de o que es miembro de algo. Meronimia es lo opuesto a la holonimia. Por lo tanto: X es merónimo de Y si X forma parte de Y. X es merónimo de Y si X es una sustancia de Y. X es merónimo de Y si X es un miembro de Y; *azul* es merónimo de *color; doctor* es merónimo de *oficio;* d*edo* es un merónimo de *mano*.

Metáfora: (del griego *meta*, «más allá», y *forein*, «pasar», «llevar») es un recurso literario (un tropo) que consiste en identificar dos términos entre los cuales existe alguna semejanza. Uno de los términos es el literal y el otro se usa en sentido figurado.

Neurolingüística: estudia los mecanismos del cerebro humano que posibilitan la comprensión, producción y conocimiento abstracto del lenguaje, ya sea hablado, escrito o con signos.

Nombre es una denominación que tiene una persona o que se le da a una cosa o a un concepto intangible, para distinguirla de otras. Los nombres se eligen de forma breve, para que la identificación de la persona, cosa o concepto sea fácil y rápida.

Numerales: pueden ser **cardinales** si corresponden a la serie de los números reales (un, dos, tres, cuatro, cinco...); **ordinales** si indican jerarquia, esto es, prelación o posteridad respecto a los demás de su serie (primer, sengundo, tercer, cuarto, quinto, sexto etc...); **multiplicadores** si multiplican el núcleo del sintagma nominal (*doble*, *triple*, *cuádruple*, *quíntuple*, *séptuple*, *ótuple*, *nónuple*, *décuple*, *undécuple*, *dodécuple*...); **divisores**, si dividen el núcleo del sintagma nominal (en el caso del castellano, sólo existe *medio*; para los demás se recurre a construcciones analíticas partitivas o al sufijo --*avo*) **distributivos** si reparten el núcleo del sintagma nominal (*cada*, *sendos*, *ambos*). Los **extensivos** indican cantidad o identidad imprecisa: bastante, mucho, poco, algún, ningún, cierto, bastante...

Oración bimembre verbal: es la oración "típica", por así decirlo, que se forma con dos sintagmas (uno de carácter nominal que constituye el sujeto y otro de carácter verbal que forma el predicado). La principal diferencia que tiene con la oración averbal antes vista es que en aquella se considera que la información que otorga el verbo es omitible, pues lo importante es lo que se quiere decir del tema que sea, mientras que en esta se considera esencial. Esto generalmente porque los verbos omitidos en las averbales son verbos copulativos (como "ser", o "estar"), mientras que los de las verbales son verbso que comunican acciones más específicas (como "prometer", "asesinar", o "derogar"). Por esto, la oración bimembre verbal es la que menos depende del contexto en el que se encuentr, y por lo mismo, la más autónoma. La oración bimembre verbal puede, además, ser clasificada según las propiedades de sus sintagmas, es decir, analizando las propiedades del sintagma nominal (separando entre oraciones personales e impersonales y sus clasificaciones) y las del sintagma verbal (separando entre oraciones complejas y simples).

Oración: es la mínima unidad comunicacional, con significado completo. Esto significa que es el fragmento más pequeño del enunciado que comunica una idea total, y posee independencia (es decir, podría sacarse del contexto y seguir comunicando, no lo mismo, pero algo). Las oraciones están delimitadas prosódicamente por pausas y gráficamente por comas o puntos. En las escuelas formalistas, es la unidad de análisis fundamental.

Oraciones bimembres averbales: no poseen verbos conjugados (los verboides, o "tiempos no personales del verbo" - gerundio, participio e infinitivo - no son parte del paradigma de conjugación) y se componen de dos partes: el soporte y el aporte. La relación entre estas dos aprtes es de **interdependencia**. Son extremadamente comunes en titulares de diarios y contextos por el estilo.

Oraciones bimembres: poseen dos o más miembros (o sintagmas) y pueden, por lo tanto, ser analizadas estructuralmente según sus partes. Se reconocen dos grandes grupos: las averbales y las verbales.

Oraciones complejas o **compuestas:** son aquellas en las que se une una serie de procesos verbales, generalmente subordinados unos a otros. El castellano permite la concatenación de cuantas oraciones se desee, siempre y cuando se respeten ciertas reglas pertinentes a la creación de clásusulas. Un ejemplo de oración compleja es "*María, cuyo*

hermano era piloto de la fuerza aérea, cruzó corriendo la pista de aterrizaje para encontrarse con él, a quien no veía hace tiempo", en la que encontramos sintagmas verbales en cláusulas adjetivas ("*cuyo hermano era piloto*", "*a quien no veía hace tiempo*"), adverbiales ("*corriendo*") y sustantivas ("*encontrarse con él*").

Oraciones simples o **sencillas:** son aquellas en cuyos predicados existe sólo un grupo verbal conjugado, es decir, que no contienen oraciones subordinadas. Un ejemplo de oración simple es "Los chicos *juegan* en el parque", donde sólo hay una expresión verbal: *juegan en el parque*.

Oraciones unimembres: también llamados "**predicados directos**" y están compuestas por una palabra o un grupo reducido de palabras. Estas se consideran oraciones en virtud de la definición dada antes: satisfacen las necesidades comunicativas del hablante, es decir, comunican.

Palabra: cada una de las unidades aislables de la cadena escrita, que se escriben separadamente (salvo en los casos en que se usa apóstrofo). Es la unidad formada por uno o varios fonemas, aislable y dotada de significado. La ciencia que estudia la composición y estructura interna de las palabras es la morfología.

Plural: rasgo del número que se contrapone al singular, y que denota más de un elemento al que se asocia. Distintas lenguas lo reflejan en distintas formas. En español se recoge esta información en los determinantes, los nombres, pronombres, verbos y adjetivos, utilizando casi siempre el morfema sufijo «-s».

Polisemia: capacidad que tiene una sola palabra para expresar muy distintos significados. Al igual que la homonimia, en el caso de la polisemia se asignan varios significados a un solo significante. Pero, mientras la homonimia se produce por coincidencia de los significantes de diversos signos, la polisemia se debe a la extensión del significado de un solo significante. La polisemia se puede producir por distintas causas. Manuel Justo Gil, en *Fundamentos del Análisis semántico*, Universidade de Santiago de Compostela, 1990, distingue cuatro causas: **Cambio de aplicación**. A lo largo de la historia, la realidad a la que se refiere una palabra ha cambiado de forma, o ha pasado a aplicarse a un nuevo referente: Por ejemplo, la palabra *tecla*, aplicada inicialmente a los instrumentos musicales, se ha aplicado después a las máquinas de escribir y finalmente a cualquier pieza móvil que puede pulsarse. **Especialización en un medio social**. En el lenguaje técnico de una profesión determinada, o en un estrato social en concreto, la palabra puede adquirir un significado especializado. Por ejemplo, la *masa* a la que se refiere un panadero no es la *masa* a la que se refiere un albañil que habla con su peón, y ninguna de estas dos es la *masa* a la que se refiere el profesor que explica una clase de física a sus alumnos. **Lenguaje figurado**. Los hablantes nombran los objetos mediante términos metafóricos (*pata* para nombrar la de la silla) o metonímicos (*copa* para nombrar el vino). **Homónimos reinterpretados**. Dos palabras homónimas con significados parecidos, cuya etimología se ha perdido pueden ser consideradas una sola palabra polisémica en la cabeza de los hablantes. Justo Gil pone como ejemplo la palabra Reja, con dos etimologías distintas: una para la reja del arado y otra para la ventana enrejada. **Influencia extranjera**. Por calco semántico, una palabra española puede adquirir significados que esa palabra tiene en una lengua extranjera. Por ejemplo, por influencia del inglés, la palabra *evento* ha adquirido el significado de 'acontecimiento importante'.

Posesivos: tipo de actualizadores que sitúan el núcleo del sintagma nominal como perteneciente a un posesor (mi, tu, su, mis, tus, sus) o varios posesores (*nuestro-a*, *vuestro-a*, *su* y sus respectivos plurales). También puede incluirse en esta categoría el pronombre relativo *cuyo-a-s*, una de cuyas múltiples funciones es la de determinante del sustantivo al cual precede y con el cual concuerda en género y número.

Pragmática: subcampo de la lingüística. Es el estudio del modo en que el contexto influye en la interpretación del significado. El contexto debe entenderse como *situación*, ya que puede incluir cualquier aspecto extralingüístico.

Predeterminante: clase de palabra que puede situarse delante de los demás determinantes (en castellano, solamente la palabra *todo*, como en "*todo* el libro")

Predicado: término que se emplea en lingüística para referirse a más de un concepto. En la gramática, tradicionalmente, se ha definido el predicado como la parte de la oración en la que se encuentra el verbo y, con frecuencia, otros sintagmas que, en el caso de haberlos, mantienen una relación con el verbo a distintos niveles de cercanía. Dentro del predicado aparecen todos los sintagmas que no tienen cabida en el sujeto de una oración. No obstante, los lingüistas transformacionales definen el predicado como la palabra que expresa un acontecimiento, el cual puede ser un estado, un suceso o una acción. Es decir, esta otra definición de predicado se refiere al verbo.

Preposición: clase de palabra invariable que introduce el llamado sintagma preposicional. Constituye una clase de nexo en tanto que liga palabras, sintagmas e incluso proposiciones, pero subordina una de estas unidades (el elemento regido) a la anterior (elemento regente), de la cual depende a través de la preposición. Su significado es

sumamente abstracto y gramatical y en la lengua precursora de las lenguas románicas, el latín, constituyó un procedimiento para evitar las imprecisiones y ambiguedades del morfema de caso, alcanzando tal éxito que vino a reemplazarlo en las lenguas románicas. Las preposiciones del idioma español son: *A*, *ante*, *bajo*, *cabe*, *con*, *contra*, *de*, *desde*, *en*, *entre*, *hacia*, *hasta*, *para*, *por*, *según*, *sin*, *so*, *sobre*, *tras*". A estas se pueden agregar también *vía*, *pro*, *mediante*, *durante*, *excepto*, *salvo*, *incluso*, *más* y *menos*, palabras que están menos gramaticalizadas. Las preposiciones pueden sufrir metábasis, es decir, cambio de función, y volverse conjunciones formando **locuciones conjuntivas**; en español suele ocurrir algunas veces cuando la preposición va seguida de un verbo en infinitivo: *Al* + inf. = *Cuando* + verbo conjugado; *De* + inf. = *Si* + verbo conjugado; *Con* + inf. = *Aunque* + verbo conjugado; *Por* + inf. = *Porque* + verbo conjugado: *Al cantar el gallo*...; *De venir Pedro*...; *Con ser tan guapo*...; *Por venir tarde*... Por otra parte, las preposiciones actúan algunas veces como nexos que unen los verbos auxiliares con los verbos en forma no personal en el caso de las perífrasis verbales: *Voy a cantar*, *He de volver*...

Preposiciones compuestas: están formadas por dos preposiciones unidas: *a por*, *por entre*, *por sobre*, *de entre*, *desde entre*, *para con*, etc...

Presuposiciones: según Strawson, son un tipo de inferencia pragmática bajo las siguientes condiciones: A presupone la afirmación B si y solamente si B es una precondición de la certeza o falsedad de A (Levinson 1984:172).

Procesamiento de Lenguaje Natural, (*PLN*, o *NLP*; *Natural Language Processing*), es una subdisciplina de la Inteligencia Artificial y la rama ingenieril de la lingüística computacional. El **PLN** se ocupa de la formulación e investigación de mecanismos eficaces computacionalmente para la comunicación entre personas o entre personas y máquinas por medio de lenguajes naturales. El **PLN** no trata de la comunicación por medio de lenguajes naturales de una forma abstracta, sino de diseñar mecanismos para comunicarse que sean eficaces computacionalmente -que se puedan realizar por medio de programas que ejecuten o simulen la comunicación. Los modelos aplicados se enfocan no sólo a la comprensión del lenguaje de por sí, sino a aspectos generales cognitivos humanos y a la organización de la memoria. El lenguaje natural sirve sólo de medio para estudiar estos fenómenos. El Procesamiento del Lenguaje Natural (PLN) es una de las piedras angulares tempranas de la inteligencia artificial (IA). La Traducción Automática, por ejemplo, nació a finales de la década de los cuarenta, antes de que se acuñara la propia expresión «Inteligencia Artificial». No obstante, el PLN ha desempeñado múltiples papeles en el contexto de la IA, y su importancia dentro de este campo ha crecido y decrecido a consecuencia de cambios tecnológicos y científicos. Los primeros intentos de traducir textos por ordenador a finales de los cuarenta y durante los cincuenta fracasaron debido a la escasa potencia de los ordenadores y a la escasa sofistificación lingüística. Sin embargo, los esfuerzos realizados en las décadas de los sesenta y los setenta para producir interfaces en lenguaje natural para bases de datos y otras aplicaciones informáticas obtuvieron un cierto grado significativo de éxito. La década de los ochenta y el principio de la de los noventa han visto resurgir la investigación en el terreno de la Traducción Automática.

Reconocimiento Automático del Habla (RAH) *o* **de voz:** parte de la Inteligencia Artificial que tiene como objetivo permitir la comunicación hablada entre seres humanos y computadoras electrónicas. El problema que se plantea en un sistema de **RAH** es el de hacer cooperar un conjunto de informaciones que proceden de diversas fuentes de conocimiento (acústica, fonética, fonológica, léxica, sintáctica, semántica y pragmática), en presencia de ambigüedades, incertidumbres y errores inevitables para llegar a obtener una interpretación aceptable del mensaje acústico recibido.

Reconocimiento Automático del Habla (RAH) *o* **de voz:** parte de la Inteligencia Artificial que tiene como objetivo permitir la comunicación hablada entre seres humanos y computadoras electrónicas. El problema que se plantea en un sistema de **RAH** es el de hacer cooperar un conjunto de informaciones que proceden de diversas fuentes de conocimiento (acústica, fonética, fonológica, léxica, sintáctica, semántica y pragmática), en presencia de ambigüedades, incertidumbres y errores inevitables para llegar a obtener una interpretación aceptable del mensaje acústico recibido.

Recuperación de información: llamada en inglés *Information retrieval* (IR), es la ciencia de la búsqueda de información en documentos, busqueda de los mismos documentos, la búsqueda de metadatos que describan documentos, o, también, la búsqueda en bases de datos, ya sea a través de internet, intranet, para textos, imágenes, sonido o datos de otras características. La IR es un estudio interdisciplinario. Cubre tantas disciplinas que eso genera normalmente un conocimiento parcial desde tan solo una u otra perspectiva. Algunas de las disciplinas que la participan de estos estudios son la psicología cognitiva, la arquitectura de la información, diseño de la información, el comportamiento humano hacia la información, la lingüística, la semiótica, informática y biblioteconomía. Los buscadores, tales como Google y Lycos, son algunas de las aplicaciones más populares de la recuperación de

información. Algunos de los estudiosos más destacados dentro de esta subdisciplina son Gerald Salton, W Bruce Croft, Karen Spärck Jones, C. J. van Rijsbergen.

Referencia: acto realizado por un comunicante que envía un mensaje (ya sea hablado, escrito o mediante otros códigos lingüísticos) para identificar algo. Para este fin utiliza determinadas expresiones.

Referente: es aquello que la palabra denota. Por ejemplo: Nombres propios se refieren a individuos. Nombres comunes se refieren a grupos de individuos. Adjetivos se refieren a cualidades. Verbos se refieren a acciones... Sin embargo, el concepto de referente conlleva ciertos problemas. Por un lado, no funciona siempre ya que no todos los verbos denotan acción, ni todos los adjetivos, cualidades... Además, Tampoco funciona cuando el nombre se refiere a una entidad que no existe. Algo imaginario. Por último, varias expresiones pueden compartir el mismo referente pero significar cosas muy distintas.

Respuesta a preguntas: llamado en inglés *Question Answering (QA)* es un tipo de recuperación de la información. Dada una cierta cantidad de documentos (tales como *World Wide Web*), el sistema debería ser capaz de recuperar respuestas a preguntas planteadas en lengua natural. *QA* es observado como un método que requiere una tecnología de Procesamiento de lenguaje natural más compleja que otros tipos de sistemas para la Recuperación de documentos, y, en algunos casos, se le observa como un paso por delante de la tecnología del buscador.

Semántica estructural: Esta tendencia observa la lengua como un sistema perfectamente estructurado. Esta corriente se desarrolla de distinta forma en Europa y en Estados Unidos. Mientras que en Europa se plantea el estudio de la **Teoría del Campo Semántico**, en Estados Unidos se impone una corriente de **análisis componencial**. La diferencia básica entre ambas corrientes es que, mientras que en la Teoría del Campo Semántico se considera que existe **interdependencia** entre diversos subsistemas lingüísticos, en el análisis componencial se descompone el significado de las palabras en unidades **independientes**, es decir, para el análisis componencial los subsistemas son independientes.

Semántica: subcampo de la gramática y, por extensión, de la lingüística. Proviene del griego "*semantikos*", que quería decir "significado relevante", derivada de "*sema*", lo que significaba "signo". Se dedica al estudio del significado de los signos lingüísticos y de sus combinaciones, desde un punto de vista sincrónico o diacrónico.

Sentido: La imagen mental de lo que algo es. Puede que, incluso, no exista en el mundo real. Es más conceptual que el referente. Por ejemplo: "amistad, felicidad"

Signo diacrítico: signo gráfico que confiere a los signos escritos (no necesariamente letras) un valor especial. Son **diacríticos**, por ejemplo: los acentos ortográficos, la diéresis, los signos empleados en el alfabeto fonético, como la oclusión (^) o la nasalización (˜), la tilde de la *ñ*, la cedilla, el ogonek, etcétera.

Sinónimos: palabras que tienen igual significado, pero tienen distinto significante, se refieren a las mismas cosas. Deben pertenecer a la misma categoría gramatical. *Por ejemplo, sinónimos de desastre son calamidad, devastacion, ruina, catástrofe y cataclismo*. Los lingüístas suelen distinguir entre: - *sinónimos totales*, que son palabras que tienen el mismo significado en todos los contextos lingüísticos (como "micrón" y "micra", dejando aparte consideraciones terminológicas). - *sinónimos parciales*, palabras que tienen el mismo significado en muchos contextos lingüísticos pero no en todos, como en el caso de coche y automóvil: así, podemos decir "Mi padre subió a un automóvil" o "Mi padre subió a un coche", pero aunque podemos decir "La locomotora arrastraba tres coches" no podemos decir "La locomotora arrastraba tres automóviles". El hablante suele tener que elegir un sinónimo dependiendo del contexto, existe una palabra más adecuada para cada situación: por el contexto geográfico: *Papa* (en canarias) / *patata* (en la península); según el lenguaje literario o común: *Estío* / *verano*; según si el lenguaje es científico o común: *Cloruro sódico* / *sal*; también según el contexto social: *Morir una planta* / *Fallecer una persona* (para una planta no se aplica *fallecer*).

Sintagma adjetivo: agrupación de palabras en torno a un adjetivo que funciona como núcleo de todas ellas, constituyendo este la palabra con más relaciones sintácticas o sintagmáticas del mismo. Por ejemplo: *Muy cercano a este lugar*, donde el núcleo es el adjetivo *cercano*. Por lo general, el sintagma adjetivo funciona como complemento adyacente de un sustantivo o de un sintagma nominal, como atributo de un verbo copulativo o semipredicativo o como complemento predicativo: Si es adyacente de un sustantivo, concuerda en género y número con el mismo: "Libro *muy interesante*". Si es atributo de un verbo copulativo, las más de las veces concuerda con el sujeto: "Esas flores son *bonitas*". Si es complemento predicativo, puede concordar o no: "Los coches corren *rápidos* o *rápido*".

Sintagma adverbial: sintagma en que el adverbio desempeña la función sintáctica de núcleo o palabra más importante y con más relaciones sintácticas: "Muy tarde para mí", por ejemplo, donde el adverbio tarde es núcleo.

Sintagma nominal (SN): es el sintagma o grupo de palabras cuyo núcleo está constituido por un sustantivo o pronombre o adjetivo sustantivado. Desempeña las mismas funciones sintácticas que puede desempeñar un sustantivo: sujeto, complemento directo de cosa, aposición, vocativo, atributo, complemento circunstancial etc... Si lleva una preposición al principio es denominado sintagma preposicional. Como sintagma preposicional puede desempeñar las funciones de complemento del nombre, complemento de un adjetivo, complemento de un adverbio, complemento directo de persona, complemento indirecto, complemento de régimen, complemento agente. atributo y complemento circunstancial. El sintagma nominal es endocéntrico, el preposicional es exocéntrico. Ejemplo: "El coche rojo es el más veloz". *El coche rojo* sería sintagma nominal. En expansión máxima, la estructura del sintagma nominal en español es la siguiente: Predeterminante + Determinante actualizador + Determinante cuantificador + Adjetivo en función de adyacente + Sustantivo o equivalente en función de núcleo + 1.º adjetivo en función de adyacente o 2.º Sustantivo o sintagma nominal en función de aposición o 3.º Sintagma preposicional en función de complemento del nombre o 4.º Proposición subordinada adjetiva en función de adyacente.

Sintagma preposicional (SP): Sintagma constituido por una preposición que funciona como núcleo y un sintagma nominal que complementa al núcleo preposicional. El núcleo puede ser una de las siguientes preposiciones: *a, ante, bajo, (cabe), con, contra, de, desde, en, entre, hacia, hasta, para, por, según sin, (so), sobre, tras*. También puede ser uno de los siguientes elementos de carácter preposicional *durante, mediante, salvo, vía, pro* y *excepto*.

Sintagma: es una estructura sintáctica en la que no existe la relación de sujeto y el predicado y consiste en un conjunto de palabras relacionadas con un núcleo (palabra más importante o con más relaciones sintácticas) que se encuentra en el interior de tal sintagma. El sintagma posee además una función sintáctica en su contexto y, a diferencia de la oración, no posee una entonación específica, al menos en español. Esto es, sintagma significa la unidad lingüística de rango superior a la palabra, constituida por un conjunto de elementos lingüísticos organizados jerárquicamente en torno a un núcleo y caracterizados por desempeñar la misma función. Se trata, por tanto, de una unidad de función. Todas las oraciones están compuestas por sintagmas (la oración misma puede considerarse un macrosintagma) y los sintagmas pueden engancharse, depender o girar unos en torno a otros mediante relaciones sintácticas de parataxis (coordinación), hipotaxis (subordinación) o relaciones morfosintácticas de concordancia, también por relaciones semánticas de cohesión y congruencia denominadas coherencia textual. La composición interior del sintagma varía desde sintagmas con una sola palabra que funciona como núcleo, hasta aquellos en los que se encuentran varios sintagmas dependientes de uno central o incluso una proposición subordinada al núcleo del sintagma. Un sintagma puede ser obligatorio por haber sido seleccionado por el predicado. A ese sintagma se le denomina argumento. El sintagma no seleccionado por el predicado es siempre opcional y se le denomina adjunto.

Sintaxis: subdisciplina de la lingüística. Es la parte de la gramática que se encarga de estudiar las reglas que gobiernan la forma en que las palabras se organizan en sintagmas y, a su vez, estos sintagmas en oraciones. La escuela sistémico funcional incluye en sus análisis sintácticos el modo en que las oraciones se organizan en estructuras de texto. Se cree que el padre de la disciplina fue Apolonio Díscolo, cuya obra *Sintaxis* es un clásico de la materia. La escuela del generativismo, también llamada transformacional centra sus estudios en la sintaxis, con el fin de poder llegar a entender elementos de lo que ellos llaman Gramática Universal.

Sintetización del habla: también llamada «síntesis del discurso» o «síntesis de voz», es la producción de discurso humano sin utilizar directamente la voz humana, mediante un proceso de síntesis que crea una voz artificial bautizada como voz sintética.

Subjuntivo: el que manifiesta lo expresado por el verbo con marcas que indican subjetividad. Es el modo de la oración adjunta a cuya acción el contenido de la principal o la clase de nexo le da carácter de posible, probable, hipotética, creída, deseada, temida, necesaria... Es el modo de lo virtual, ofrece la significación del verbo sin actualizar. **Ejemplo**: Espero que él *sienta lo mismo*. En la enseñanza de ELE (español como lengua extranjera) el modo subjuntivo del verbo es uno de los puntos que presenta más dificultades.

Sujeto agente: es el que realiza, controla o preside la acción que ejecuta el verbo, y por tanto aparece siempre en las *oraciones activas*: *Pedro* come peras. *El rey* ganó la regata. *Felipe II* construyó El Escorial.

Sujeto causativo: es el que no ejecuta directamente la acción, pero la preside: "*Felipe II* construyó El Escorial"

Sujeto expreso: es el que aparece en la oración: **Alfonso** corre mucho.

Sujeto múltiple: es aquel cuyo sintagma nominal posee dos núcleos: "*Pedro y Luis* salieron a pescar.

Sujeto omitido o elíptico: es el sujeto que no aparece pero que nos descubre el verbo: Corren mucho. *(Ellos/as)*

Sujeto paciente: es el que padece la acción realizada por el verbo y ejecutada por un complemento agente con la preposición por o de, que puede aparecer o no; por eso es el sujeto de las oraciones pasivas: "*La circulación* fue desviada por la carretera (por el policía de tráfico)". "*El paciente* fue operado por el doctor". "*Lorca* era conocido de todos". "Se vende *piso*".

Sujeto: en la parte denominada Sintaxis de la Gramática, el **sujeto** es en una oración aquello que no forma parte del predicado y se constituye en soporte del mismo. El sujeto es, pues, aquello (persona, animal o cosa) de que se dice o comenta algo en una oración y que concierta en morfema de número y morfema de persona con el verbo que es núcleo del predicado.

Sustantivación: acto de creación léxica que tiene como consecuencia la formación de un sustantivo partiendo de otro tipo de palabra. En español se suele realizar precediendo un artículo a la expresión que se sustantiva. Por ejemplo: «bueno» (adjetivo) >> «lo bueno».

Sustantivo: clase de palabra que se caracteriza por ser la única que puede funcionar como núcleo del sintagma nominal y consiste en la persona, institución, animal o cosa concreta o abstracta que ejecuta o recibe directamente la acción del verbo. En español admite como acompañantes a artículos y otros determinantes y adjetivos que concuerden en género y número con ellos y a sustantivos en aposición que pueden no concordar. **Funciones del sustantivo en una oración**: Núcleo del sintagma nominal sujeto: La **niña** va a la escuela; Núcleo del predicado nominal o **atributo** en las oraciones con el verbo *ser*, *estar* o *parecer* u oraciones atributivas: Mi casa es de **madera** maciza; Complemento preposicional de otro sustantivo (complemento del nombre): La falda de **María** tiene lunares; Aposición, esto es, complemento de otro sustantivo, pero sin preposición: Río **Tajo**, Calle **Alcalá**, Madrid **capital**; Complemento de un adjetivo: Me gusta el color verde **pistacho**; Complemento del verbo. El sustantivo puede ser: Objeto o complemento directo: Me he comprado un **coche** nuevo; Objeto o complemento indirecto: No des besos al **perro**; Complemento circunstancial: Los niños están jugando en la **calle**; Complemento agente de la pasiva: El puerto fue destruido por el **huracán**; Complemento de régimen o suplemento directo: Pedro habló de **política**; Complemento de régimen o suplemento indirecto: El camarero limpió el suelo de **colillas;** Complemento predicativo de verbos que significan "nombrar" o "elegir": La asamblea eligió **presidente** a Pedro; Locuciones adverbiales:En realidad realmente no sé qué hacer; **Vocativo:** ¡**Hombre**, no me digas eso!

Texto: composición de signos codificado en un sistema de escritura (como un alfabeto) que forma una unidad de sentido. Su tamaño puede ser variable, desde una obra literaria como "El Quijote" al mensaje de volcado de pila de Windows NT. También es texto una composición de caracteres imprimibles (con grafía) generados por un algoritmo de cifrado que aunque no tienen sentido para cualquier persona si puede ser descifrado por su destinatario texto claro original.

Traducción automática: consiste en convertir un texto de un idioma a otro automáticamente, por medio del ordenador. Se trata de una disciplina que ha contribuido de manera determinante al desarrollo de la lingüística computacional. Es seguramente también una de las aplicaciones informáticas que mayores recursos humanos y económicos ha recibido. El mercado ofrece en la actualidad un amplio abanico de productos y es difícil para el profano elegir el más adecuado para sus necesidades. Con todo, es importante saber que un texto producido por un sistema de traducción automática debe ser revisado con cuidado antes de darlo por válido y publicarlo. Hay veces, sin embargo, que no es necesario obtener resultados de calidad y basta con una aproximación al contenido, si lo que queremos es detectar por ejemplo información relevante o crítica.

Transitividad: característica propia de algunos verbos que les confiere la propiedad de poder seleccionar un complemento directo, permitiendo precisar el alcance del verbo. Ejemplos: **Matar**. Siempre es necesario especificar a que/quien se mata para completar la frase. **Tener**. Siempre se tiene algo o a alguien, el verbo en si mismo no precisa la acción. Los verbos transitivos pueden aparecer sin complemento directo, cuando este está claramente determinado por el contexto o en los casos de uso absoluto, es decir, cuando la acción expresada por el verbo es importante en si misma. Algunos verbos transitivos admiten voz pasiva.

Verbo: categoría gramatical que funciona como núcleo del predicado y suele indicar acción (traer, leer, etc.), proceso (pensar, creer, etc.) o estado (existir, vivir, permanecer, ser, etc.). En español constituye la clase de palabra más variable.

Verbos copulativos son: ser, estar, parecer, resultar

Verbos defectivos: aquellos en los que no se cumple el paradigma de conjugación completo. Para estos verbos no existen conjugaciones en algunos tiempos y personas, principalmente debido a razones de eufonía o de uso. El ejemplo más conocido de esta categoría es el verbo "abolir".

Verbos impersonales impropios: verbos que si bien en algunos contextos poseen una conjugación normal, pueden ser usados como impersonales (de ahí su categoría de impropios). Por ejemplo: el verbo "*hacer*" puede ser usado en contextos como "*Ella hace pasteles*" o en frases como "*Hace calor*".

Verbos impersonales propios: verbos que, en su sentido original (es decir, no-metafórico) se conjugan sólo en la 3° persona del singular (*él*). Dicha categoría está compuesta por los llamados "verbos meteorológicos" o "climáticos" (*llueve*, *nieva*, etc.). Estos verbos son intransitivos.

Verbos impersonales: aquellos que no son compatibles con la idea de un sujeto (y por lo mismo con una coordinación con una persona), y se separan en los que son considerados **propios** (también llamados "unipersonales"), y los **impropios**.

Verbos intransitivos: aquellos que no necesitan acompañarse de un complemento directo, tienen significado completo:Pedro canta. En el uso lingüístico los verbos no son en sí mismos transitivos o intransitivos, sino que se denominan así según su uso: Pedro canta ópera (uso transitivo) Pedro canta muy bien (uso intransitivo).

Verbos irregulares: aquellos que poseen conjugaciones particulares para los llamados "tiempos verbales primitivos" o simplemente "tiempos primitivos" que son el Presente del Modo indicativo ("Yo quepo"), el Pretérito perfecto simple del indicativo ("Yo cupe") y el Futuro del mismo modo ("Yo cabré").

Verbos regulares: aquellos que se atienen estrechamente a los paradigmas o modelos de conjugación más usados en la lengua. En español hay tres de esos paradigmas: la primera conjugación, cuyos infinitivos terminan en **-ar**; la segunda, en la que terminan en **-er** y la tercera, en la que terminan en **-ir**. Dentro de la conjugación regular puede considerarse también una copnjugación extendida por medio de perífrasis verbales que señalan distintos tipos de aspecto y modo verbal.

Verbos terciopersonales: se asocian a un número reducido de verbos que se conjugan exclusivamente en la 3° persona, ya sea del singular o el plural (*él* y *ellos*). Sin embargo, y a diferencia de la categoría recién mencionada, estos sí cuentan con un sujeto y concuerdan con él. Los verbos terciopersonales son: Acontecer, Suceder, Ocurrir, Constar, Parecer, Bastar (en su forma "bastar s/preposición"): "Me basta tu presencia".

Índice de tablas y figuras

Resumen

Determinación automática de roles semánticos usando preferencias de selección sobre corpus muy grandes

Esta tesis inicia con un estudio general del estado del arte de la ciencia que estudia formalmente la interacción entre el lenguaje natural y las computadoras: la **lingüística computacional** (sección 2.1). En la sección 1.1 planteamos el objetivo específico de esta tesis, dentro del amplio marco descrito anteriormente. Para lograr este objetivo, existen diversos caminos. En el **Capítulo 2** exploramos el estado del arte de diversos enfoques y señalamos sus ventajas y desventajas: en la **sección 2.2.1**, buscaremos una representación de textos mediante **formalismos de constituyentes**. En la **sección 2.2.2** buscaremos la comprensión de las expresiones del usuario a través de **reglas de reescritura**. Finalmente compararemos estos dos enfoques con el **formalismo de dependencias** en la **sección 2.2.3**.

En este trabajo utilizaremos este último enfoque, por ser el que mejor se adecúa a nuestro objetivo. Las razones principales para ello se describen en la **Motivación**, **Sección 1.2**.

En el **Capítulo 3** describimos la estructura general del sistema propuesto. Este sistema usa un conjunto ordenado de reglas heurísticas simples para determinar iterativamente las relaciones entre palabras a las cuales no se les ha asignado aún un gobernante. En el caso de ambigüedades de ciertos tipos, se utilizan estadísticas de co-ocurrencia de palabras reunidas previamente de una manera no supervisada a partir de un corpus grande, o a partir de la Web (a través de un buscador como Google). La recopilación de estas estadísticas se realiza mediante **preferencias de selección**, tema que abordamos en detalle en el **Capítulo 4**. Una ambigüedad particularmente importante que hemos decidido tratar a detalle, es la **desambiguación de unión de sintagma preposicional**. Este tema es tratado a detalle en el **Capítulo 5**.

Con el objeto de evaluar nuestro sistema, desarrollamos un **método para convertir un estándar de referencia**, en formato de gramática de constituyentes, a formato de dependencias. La descripción de este método aparece en el **Capítulo 6.3.2**. Una vez que se cuenta con el recurso del **estándar de referencia**, procedemos a evaluar nuestro sistema como se describe en el **Capítulo 6**. Adicionalmente, cada uno de los módulos del sistema (**obtención de preferencias de selección** y

desambiguación de unión de sintagma preposicional), fueron evaluados de manera separada e independiente para garantizar su correcto funcionamiento.

En el **Capítulo 7** presentamos algunas **aplicaciones** de nuestro sistema: **Desambiguación de sentidos de palabra** (**Capítulo 7.1**) y **Esteganografía lingüística** (**Capítulo 7.2**). Finalmente en el **Capítulo 8** anotamos nuestras conclusiones.

Abstract

Automatic Determination of Semantic Roles Using Selectional Preferences on Very Big Corpora

This thesis begins with a general state of the art overview of the science which formally studies the interaction between natural language and computers: **Computational Linguistics** (section 2.1). In Section 1.1 we expose the specific goal of this thesis, within the wide framework described previously. To reach this goal, there are several paths. In **Chapter 2** we explore the state of the art of several approaches and we point at their advantages and disadvantages: in **Section 2.2.1**, we will search for a representation of texts by means of constituent formalisms. In **Section 2.2.2** we will search for the understanding of expressions through **Rewriting Rules**. Finally we will compare both approaches with the **Formalism of Dependencies** in **Section 2.2.3**.

Through this work we will use the approach of **dependencies**, because it is more adequate for reaching our goal. The specific reasons for this can be found in the section **Motivation**, **Section 1.2**.

In **Chapter 3** we will describe the general structure of the proposed system. This system uses an ordered set of simple heuristic rules for determining iteratively the relationships between words to which a governor has not been yet assigned. For resolving certain cases of ambiguity we use co-occurrence statistics of words collected previously in an unsupervised manner, wether it be from big corpora, or from the Web (through a search engine such as Google). Collecting these statistics is done by using **Selectional Preferences**, subject which we study in detail in **Chapter 4**. A particularly interesting ambiguity which we have decided to analyze deeper, is the **Prepositional Phrase Attachment Disambiguation**. This subject is covered in **Chapter 5**.

In order to evaluate our system, we developed a **Method for Converting a Gold Standard** from a constituent format to a dependency format. The description of this method appears in **Chapter 6.3.2**. Once we have an suitable **gold standard**, we proceed to evaluate our system as it is described in whole in **Chapter 6**. Aditionally, each one of the modules of the system (**Selectional Prefererences Acquisition** and **Prepositional Phrase Attachment Disambiguation**), is evaluated in a separate and independent way to verify that they work properly.

In **Chapter 7** we present some **Applications** of our system: **Word Sense Disambiguation** (**Chapter 7.1**) and **Linguistic Steganography** (**Chapter 7.2**). Finally in **Chapter 8** we draw or Conclusions.

1 Introducción

> ***El tesoro más valioso de la raza humana es el conocimiento****. Las computadoras tienen una capacidad mucho más grande que las personas para manejar el conocimiento: usarlo para razonar, buscar información nueva, buscar respuestas a preguntas... Sin embargo, nuestro tesoro —que existe en la forma de textos en lenguaje natural: mensajes de noticias, periódicos y libros que están en bibliotecas digitales y en la biblioteca mundial que es Internet— simplemente no es entendible para las computadoras; lo tratan como cadenas de letras y no como conocimiento.*
>
> A. F. Gelbukh

1.1 Objetivo

Esta tesis propone un modelo para obtener la estructura de una oración basándose en las características sintácticas y semánticas de los componentes que la constituyen. El modelo considera un algoritmo de desambiguación basado en conocimiento lingüístico y semántico obtenido a partir de una gran cantidad de texto.

La estructura propuesta pone especial énfasis en segmentar adecuadamente las estructuras que corresponden a entidades mencionadas en la oración, como por ejemplo {*el hombre con traje gris que se encuentra parado en aquella esquina*} *es mi padre*. De esta manera se busca facilitar tareas posteriores de análisis de textos como búsqueda de respuestas, búsqueda de información, traducción automática, o formalización lógica de textos.

Para lograr este objetivo, analizamos características del español para aplicar heurísticas relativamente simples en la agrupación de estructuras. Por ejemplo, una heurística muy sencilla es que un determinante casi siempre antecede a un sustantivo: *el libro*, *la casa*, etc. Poco a poco estas heurísticas se van complicando hasta tener reglas para procesar oraciones subordinadas y relativas. Durante este proceso existen muchos casos de ambigüedad, los cuales son atacados mediante conocimiento lingüístico extraído automáticamente a partir de colecciones grandes de textos. Este conocimiento es conocido como **preferencias de selección**.

La investigación descrita en esta tesis incluye nuevas contribuciones en el aspecto de extracción automática de preferencias de selección, y sus múltiples aplicaciones, así como el establecimiento de algunas convenciones para la representación de una oración en estructura de dependencias.

1.2 Motivación

Este trabajo surge de una doble motivación. Una tarea que teníamos en mente era el estudio de la compatibilidad léxica de palabras específicas, y en particular, la compilación y el uso de un diccionario de colocaciones (combinaciones estables o frecuentes de palabras, como *comer pan* o *sueño profundo*, en oposición a ?*comer sueño* y ?*pan profundo* [17]. Dichas combinaciones han demostrado ser útiles en tareas que van desde el análisis sintáctico [199] y traducción automática [21] a corrección de errores semánticos [22] y esteganografía [14]. El enfoque de dependencias al análisis sintáctico parece mucho más apropiado para dicha tarea.

Nuestra segunda motivación fue la construcción de la representación semántica del texto, incluso parcialmente, para un rango de aplicaciones desde recuperación de información y minería de texto [133, 132] hasta especificaciones de software [68]. Todos los enfoques semánticos conocidos (como grafos conceptuales [176], Recursión de Semántica Mínima (MRS) [57], o redes semánticas [124]) se parecen a grandes rasgos a un conjunto de predicados, donde las palabras individuales representan predicados de sus argumentos (quienes a su vez pueden ser también predicados). Las estructuras resultantes están en una correspondencia mucho más directa con un árbol de dependencias que con un árbol de constituyentes de la oración en cuestión, de tal forma que la sintaxis de dependencias parece ser más apropiada para su traducción directa en estructuras semánticas. Específicamente, la estructura de dependencias hace que sea mucho más fácil hacer que coincidan (por ejemplo, en recuperación de información) paráfrasis del mismo significado (como la transformación de voz pasiva en activa y viceversa), o transformar de una estructura equivalente a otra.

Adicionalmente, encontramos que la estructura producida por un analizador de dependencias puede obtenerse fácilmente de una manera más robusta que un analizador de constituyentes. Los enfoques conocidos del análisis de dependencia tratan mucho más fácilmente tanto con gramáticas incompletas y oraciones no gramaticales, que los enfoques estándar del análisis libre de contexto.

Un analizador estándar libre de contexto construye la estructura incrementalmente, de tal forma que una falla al construir un constituyente implica la imposibilidad de construir todos los constituyentes posteriores que deberían haber contenido a éste. Lo que es peor, una decisión incorrecta en una etapa inicial de análisis conduce a un resultado final completa o ampliamente incorrecto.

En contraste, en el análisis de dependencias la selección de un gobernante para una palabra dada, o la decisión acerca de si dadas dos palabras están conectadas o no con una relación de

dependencias, es mucho más (aunque no del todo) independiente con respecto a la decisión correspondiente en otro par de palabras. Esto hace posible continuar el proceso de análisis incluso si algunas decisiones no pudieran haberse hecho exitosamente. La estructura resultante puede ser incompleta (con algunas relaciones faltantes), o no correcta del todo (con algunas relaciones identificadas erróneamente). Sin embargo, una decisión incorrecta sobre un par particular de palabras usualmente no causa una bola de nieve de errores en cascada en pasos futuros de análisis.

1.3 Justificación

A pesar de existir actualmente diversos trabajos sobre representaciones lingüísticas de oraciones en lenguaje natural, existen muy pocos que se centren en el problema particular del análisis computacional de dependencias para el español. En este trabajo presentamos un analizador capaz de producir una estructura con roles semánticos que es capaz de competir con los mejores analizadores existentes que realizan tareas similares.

Actualmente el español ocupa el segundo lugar entre los idiomas más hablados del mundo[1] , precedido únicamente por el chino, y seguido del inglés, en tercer lugar. Esta es una de las razones por las cuales este trabajo cobra particular relevancia e importancia, contribuyendo al avance de la ciencia de la Lingüística Computacional.

En la siguiente sección comentamos las principales aportaciones de este trabajo.

1.4 Aportaciones

Las aportaciones principales de este trabajo son:

DILUCT: Un analizador sintáctico de dependencias para el español (realizamos pruebas contra analizadores similares, logrando un mejor desempeño. Vea el capítulo 6)

Una base de preferencias de selección para 3 millones de combinaciones diferentes, 0.43 millones de ellas involucran preposiciones (Vea el capítulo 4)

Diversos algoritmos para unión de frase preposicional. Mejora de algoritmos existentes. (Vea el Capítulo 5)

Creación de un tesauro distribucional para el español siguiendo el método de Lin (Sección 5.3.6.3.1)

[1] Según el *ethnologue* del Instituto Lingüístico de Verano (SIL), 1999.

Comparación de diccionarios manuales *vs*. diccionarios obtenidos automáticamente. El resultado de esta investigación sugiere que los diccionarios obtenidos automáticamente por computadora pueden sustituir a los diccionarios creados manualmente en ciertas tareas, ahorrando años de trabajo. (Vea Sección 5.3.6)

Un método para convertir un corpus anotado de constituyentes en un corpus de dependencias (Vea Sección 6.3.2)

2 Estado del arte

2.1 Procesamiento de Lenguaje Natural

2.1.1 Lenguaje natural y lingüística computacional[2]

La ciencia que estudia el lenguaje humano es la lingüística. Dentro de esta gran ciencia existen ramas que representan su intersección con otras ramas 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, una relación muy especial e interesante de gran beneficio mutuo existe 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 para nuestra incipiente sociedad informática –por ejemplo la búsqueda y el manejo de conocimiento, las interfaces en lenguaje natural entre el humano y las computadoras o los robots, la traducción automática, entre un sinnúmero de otras aplicaciones de alta tecnología.

Por otro lado, las tecnologías computacionales pueden dotar al lingüista con herramientas inalcanzables para los investigadores de tiempos tan cercanos como hace un par de décadas, y de las cuales hace unos cuántos años los lingüistas no podían disponer para sus labores cotidianas por el prohibitivo costo de las computadoras. Entre estas herramientas se pueden mencionar la inmediata búsqueda de ejemplos de uso de las palabras y construcciones en enormes cantidades de textos; las estadísticas complejas conseguidas milagrosamente rápido; el análisis, marcaje y clasificación casi instantáneas –en comparación con hacerlas con lápiz y goma de borrar– de cualquier texto; la detección automática de la estructura en un lenguaje desconocido, para mencionar sólo algunos. Los buscadores avanzados de Internet han abierto la puerta a todo un mundo de lenguajes, a un corpus tan enorme que puede considerarse como todo el lenguaje humano disponible en forma palpable y medible –a diferencia de un corpus tradicional que sólo representa una gotita del océano del uso colectivo del lenguaje.

---

[2] Tomado de [82]

Entre estos beneficios, destaca la posibilidad de la verificación masiva de las teorías, gramáticas y los diccionarios lingüísticos. Hace unos años, para verificar una gramática propuesta por un estudioso colega, el lingüista esforzaba su intuición en busca de un ejemplo no cubierto por ella, y si no encontraba tal ejemplo, tenía que admitir que la gramática era completa –lo que no es un buen ejemplo 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 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 es 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 formular sus observaciones y pensamientos en forma de las 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 –aún cuando los mismos autores no lo noten– a su sentido común, su intuición y su conocimiento propio del lenguaje. Históricamente el primer reto para tales descripciones –el cual ayudó muchísimo a elevar su claridad y lo que ahora se llama formalidad– fue la descripción de los lenguajes extranjeros, en la cual ya no se puede apelar al sentido propio lingüístico del lector. Sin embargo, incluso en estas descripciones muy a menudo se apoyaba 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 una propiedad singular: uno que no sabe nada de antemano, no tiene ninguna intuición ni sentido común, y sólo es capaz –enormemente capaz– de interpretar y aplicar literalmente las descripciones de lenguaje que el lingüista le proporciona: una computadora. Como cuando un niño nos hace preguntas que nos hacen pensar profundamente en las cosas que siempre hemos creído obvias –pero de hecho muy difíciles de explicar– y que no hubièramos pensado si no nos hubiera preguntado, así la computadora hace al lingüista afilar y completar sus formulaciones, y a veces buscar las respuestas a las preguntas tan difíciles de responder que antes era más simple considerarlas «obvias». De la misma manera la computación convierte a la lingüística –que era tradicionalmente una rama de las humanidades– en una ciencia exacta, y además le da nuevos retos, nuevas motivaciones y nuevas direcciones de

investigación. Esta transformación se puede comparar con las que en distintos momentos hicieron las matemáticas con la física.

El amplio campo de la interacción e intersección entre la lingüística y la computación se estructura a su vez en varias ciencias más específicas. Una de éstas se llama la lingüística computacional. Esta ciencia trata de la construcción de los modelos de lenguaje «entendibles» para las computadoras, es decir, más formales que los modelos tradicionales orientados a los lectores humanos.

2.1.2 Niveles de procesamiento lingüístico[3]

En el aspecto técnico, el procesamiento de lenguaje natural enfrenta gran complejidad del conocimiento involucrado. La compilación de este conocimiento es uno de los problemas de la ingeniería de sistemas lingüísticos; una de las soluciones a este problema es el aprendizaje automático del conocimiento a partir de los corpus grandes de textos.

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

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

[3] Tomado de [82]

No existen criterios exactos para la separación de cada uno de los niveles; más bien las diferencias entre los niveles se basan en el enfoque de análisis en cada nivel. 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 como en *acordar-acuerdo*, *dirigir-dirijo*, entre otros casos.

A continuación vamos a discutir brevemente cada nivel de lenguaje y sus avances computacionales.

2.1.2.1 Fonética / fonología

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

Los problemas en fonética computacional están relacionados con el desarrollo de sistemas de reconocimiento de voz y síntesis de habla. Aunque 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 aún es bastante bajo. En 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 `loquendo.com` para hacer pruebas con varios módulos de generación. Hablando de los sistemas de síntesis de habla hay que decir 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 está en la posición del sonido en el sistema de 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 el español usualmente tienen un acento hablando ciertos idiomas, cuando no pueden pronunciar [l duro], como se pronuncia [l] en inglés? La respuesta es la misma en todos los casos: en sus 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 las 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 hablando inglés, donde el fonema [l] se pronuncia duro, lo pronuncian de manera suave como en su idioma natal.

2.1.2.2 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 lenguajes que tienen bastantes diferencias con respecto a lo que tenemos en español. Por ejemplo, en el árabe la raíz contiene tres consonantes, y las diferentes formas gramaticales de la palabra se hacen por medio de la inserción de vocales entre las consonantes (*KiTaB <el libro>, KaTiB <leyendo>*, etc.); en el chino casi no existen las formas morfológicas de palabras, lo que se recompensa en el nivel de 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 traducir 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 en el español. Nótese que pueden existir las 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 engorroso porque hay que hacer grandes diccionarios de raíces (alrededor de cien mil). En general existe la metodología de tal desarrollo y existen sistemas funcionando para muchos idiomas. Lo que hace falta aquí es un estándar de tales módulos. En el CIC hemos desarrollado un sistema de análisis morfológico para el español disponible para todos los que lo necesiten. Véanse últimas secciones.

2.1.2.3 Sintaxis

La tarea principal en este nivel es describir cómo las palabras de la oración se relacionan y cuál es la función que cada palabra realiza en esa oración; es decir, construir la estructura de la oración de un lenguaje.

Las normas o reglas para construir las oraciones se definen para los seres humanos en una forma prescriptiva, indicando las formas de las frases correctas y condenando las formas desviadas, es decir, indicando cuáles se prefieren en el lenguaje. En contraste, en el procesamiento lingüístico de textos, las reglas deben ser descriptivas, estableciendo métodos que definan las frases posibles e imposibles del lenguaje específico de que se trate.

Las frases posibles son secuencias gramaticales, es decir, que obedecen leyes gramaticales, sin conocimiento del mundo, y las no gramaticales deben postergarse a niveles que consideren la noción de contexto en un sentido amplio, y el razonamiento. Establecer métodos que determinen únicamente las secuencias gramaticales en el procesamiento lingüístico de textos ha sido el objetivo de los formalismos gramaticales en la Lingüística Computacional. En ella se han considerado dos enfoques para describir formalmente la gramaticalidad de las oraciones: las dependencias y los constituyentes.

La sintaxis se dedica a los estudios de relaciones entre las palabras de la frase. Principalmente existen dos modelos 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 frase, o generar la frase basándose en su estructura. El desarrollo de los generadores es una tarea más fácil y es claro qué algoritmos son los necesarios 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 palabras es fijo, por eso las teorías basadas en inglés no son tan fácilmente adaptables para el español. Vamos a presentar un ejemplo de *parser* en las siguientes secciones.

2.1.2.4 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 en 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 es ya de por sí una tarea muy difícil aún con trabajo manual. Los resultados de tal definición de los sentidos de las palabras existen en la forma de

diccionarios. Aquí el problema principal es que siempre [4] existe un círculo vicioso en las definiciones de las palabras, porque las palabras se definen a través de otras palabras. Por ejemplo, si definimos a *gallo* como "el macho de la gallina" y a *gallina* como "la hembra del gallo", eso no ayudará 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 buscando un conjunto de las palabras a través de las cuales se definirán 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 tanto diccionarios buenos, como 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*. Cuál de los sentidos se usa en un contexto dado, se puede tratar de averiguar analizando las demás palabras presentes en el contexto aplicando diferentes métodos. 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, para eso tanto un humano como una computadora requieren un contexto más amplio.

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

2.1.2.5 Pragmática

Usualmente se dice que la pragmática se trata de 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 que pasara 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.

---

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

Otro ejemplo del dominio de la pragmática es la clase de oraciones que tienen una característica muy interesante: ellas son las acciones por sí mismas (se llaman performativas). Por ejemplo, decir *prometo* es precisamente la acción de *prometer*.

Como nos topamos con muchas dificultades ya en nivel semántico, normalmente es difícil continuar la cadena de análisis en el siguiente nivel, aunque siempre hay que tomarlo en cuenta.

2.1.2.6 Discurso

Normalmente hablamos no con una oración aislada, sino con varias oraciones. Esas oraciones tienen ciertas relaciones entre sí, lo que las hace algo más que sólo oraciones. Lo que aparece entonces es una nueva entidad llamada *discurso*.

En el análisis de 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, las cuales tienen que ser interpretadas correctamente por la computadora para poder construir las representaciones semánticas.

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

2.1.3 Ambigüedades en lenguaje natural

La ambigüedad en el proceso lingüístico se presenta cuando pueden admitirse distintas interpretaciones a partir de la representación, o cuando existe confusión al tener diversas estructuras y no tener los elementos necesarios para eliminar las incorrectas. Para desambiguar, es decir, para seleccionar los significados o las estructuras más adecuados de un conjunto conocido de posibilidades, se requieren diversas estrategias de solución en cada caso.

Relacionada a la sintaxis, existe ambigüedad en el marcaje de partes del habla, esta ambigüedad se refiere a que una palabra puede tener varias categorías sintácticas, por ejemplo *ante* puede ser una preposición o un sustantivo, etc. Conocer la marca correcta para cada palabra de una oración ayudaría en la desambiguación sintáctica, sin embargo la desambiguación de este marcaje requiere a su vez cierta clase de análisis sintáctico.

En el análisis sintáctico es necesario tratar con diversas formas de ambigüedad. La ambigüedad principal ocurre cuando la información sintáctica no es suficiente para hacer una decisión de

asignación de estructura. La ambigüedad existe aún para los hablantes nativos, es decir, hay diferentes lecturas para una misma frase. Por ejemplo en la oración *Javier habló con el profesor del CIC*, puede pensarse en *el profesor del CIC* como un complemento de *hablar* o también puede leerse que *Javier habló con el profesor* sobre un tema, *del CIC*.

También existe ambigüedad en los complementos circunstanciales. Por ejemplo, en la frase *Me gusta beber licores con mis amigos*, el grupo *con mis amigos* es un complemento de *beber* y *no de licores*. Mientras un hablante nativo no considerará la posibilidad del complemento *licores con mis amigos,* para la computadora ambas posibilidades son reales.

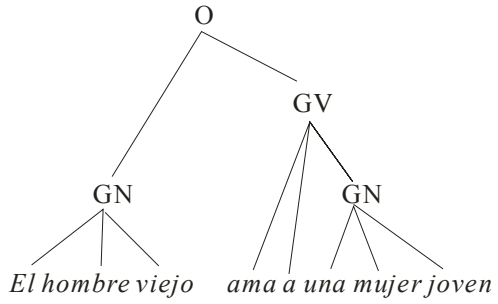
Como mencionamos, la información léxica puede ayudar a resolver muchas ambigüedades, en otros casos la proximidad semántica puede ayudar en la desambiguación. Por ejemplo: *Me gusta beber licores con menta* y *Me gusta beber licores con mis amigos;* en ambas frases la clase semántica del sustantivo final ayuda a resolver la ambigüedad, esto es, con qué parte de la frase están enlazadas las frases preposicionales, *con menta* y *con mis amigos*. Ni *menta* ni *amigos* son palabras ambiguas pero *amigos* está más cercana semánticamente a *beber* que a *licores* y *menta* está más cercana a *licor* que a *beber*.

2.2 Enfoques de análisis sintáctico

Los dos enfoques principales al análisis sintáctico están orientados a la estructura de constituyentes y de dependencias, respectivamente. En el enfoque de constituyentes, la estructura de la oración se describe agrupando palabras y especificando el tipo de cada grupo, usualmente de acuerdo con su palabra principal [47]:

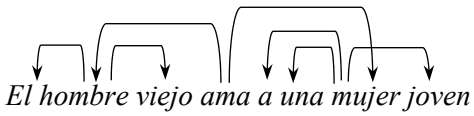
$$[\textit{El hombre viejo}]_{GN}\ [\textit{ama a}\ [\textit{una mujer joven}]_{GN}]_{GV}]_{O}$$

Aquí GN quiere decir Grupo Nominal, GV Grupo Verbal, y O la oración completa. Dicho árbol puede ser representado gráficamente:

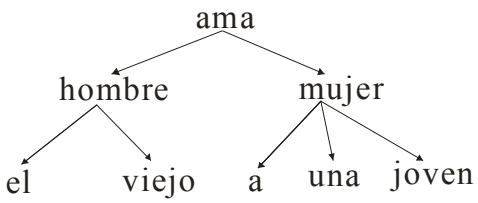


donde los nodos son partes de texto (constituyentes) y los arcos son relaciones de "consiste en".

En el enfoque de dependencias, las palabras se consideran "dependientes" de, o que modifican otras palabras [124]. Una palabra (gobernada) modifica otra palabra (gobernante) en la oración si añade detalles a esta última, mientras que la combinación completa hereda las propiedades sintácticas (y semánticas) del gobernante: *hombre viejo* es un tipo de *hombre* (y no un tipo de *viejo*); *hombre ama a mujer* es un tipo de (situación de) *amor* (y no, digamos, un tipo de *mujer*). Dicha dependencia se representa por una flecha del gobernador a la palabra gobernada:



o, en forma gráfica:



donde los arcos representan la relación de dependencia entre palabras individuales. Las palabras de los niveles inferiores contribuyen con detalles a aquellos de los niveles superiores mientras que se preservan las propiedades sintácticas de estos últimos.

A pesar de la discusión en la literatura de ya más de 40 años, no existe un consenso sobre cuál formalismo es mejor. Aunque formalismos combinados como HPSG [166] han sido propuestos, parecen heredar tanto las ventajas como las desventajas de ambos enfoques, lo cual ha impedido su uso amplio en la práctica del procesamiento de lenguaje natural. Probablemente la pertinencia de uno de los dos enfoques depende de la tarea específica.

En los siguientes capítulos trataremos de ambos enfoques con más detalle.

2.2.1 Análisis usando gramáticas de constituyentes

En este capítulo abordamos la estructura de una oración siguiendo el enfoque de constituyentes. En este enfoque se presentan dos problemas: el primero, la extracción de la información acerca de caracteres y acciones a partir de un reporte factual como noticias, páginas web o historias circunscritas. Para construir la estructura de las oraciones, debe existir interacción con conocimiento adquirido previamente. A su vez, este conocimiento debe ser expresado de una manera estructurada de tal manera que puedan hacerse inferencias al usarlo. En la Sección 2.2.1.1 tratamos este tema.

Otro problema a enfrentar cuando queremos obtener una estructura de constituyentes a partir de una oración, es el de mantener los índices semánticos a través de diversas oraciones. Actualmente no

existe un mecanismo inherente a los formalismos de constituyentes más conocidos como HPSG [166] que considere este fenómeno, por lo que abordamos este problema en la siguiente sección.

2.2.1.1 Representación usando estructuras con características tipificadas (Typed Feature Structures)

2.2.1.1.1 Introducción

En este capítulo examinamos la extracción de información acerca de los roles semánticos utilizando Estructuras con Características Tipificadas (Typed Feature Structures). Enfocamos nuestro estudio en extraer información acerca de personajes y acciones de una historia auto-contenida, como cuentos para niños. Esta información se almacena en estructuras llamadas situaciones. Mostramos cómo las situaciones pueden construirse al unificar los constituyentes del análisis de la oración con conocimiento previamente almacenado en Estructuras con Características Tipificadas (TFS). Estas situaciones a su vez pueden ser usadas subsecuentemente en la forma de conocimiento. La combinación de situaciones construye una supra-estructura que representa la comprensión del reporte factual. Esta estructura puede utilizarse para responder preguntas acerca de hechos y sus participantes.

Decidimos centrarnos en historias para niños porque estas historias son textos en los cuales los hechos se describen de una manera ordenada y todos los participantes de estos hechos están circunscritos en el texto. Estas características nos permiten aplicar técnicas de manejo de conocimiento de complejidad práctica, es decir, aplicaciones intermedias factibles. Para ejemplificar el tipo de estructuras que queremos obtener, nos basaremos en el mismo fragmento de texto utilizado por Minsky en [129] con el cuál él ejemplifica qué quiere decir con *comprender* una historia. El texto es el siguiente:

"Había una vez un lobo que vio a una oveja bebiendo en un río y quería una excusa para comerla. Para tal propósito, a pesar de que se encontraba río arriba, acusó a la oveja de agitar el agua e impedir que bebiera de ella"

Minsky argumenta que comprender este texto implica entender las siguientes situaciones:

1. El hecho de que la oveja está agitando el agua produce lodo,
2. Si el agua tiene lodo, no puede ser bebida
3. Si el lobo está río arriba, el hecho de que la oveja agite el agua no afecta al lobo, y por lo tanto,

4. El lobo está mintiendo.

Estas inferencias requieren un sistema de conocimientos estructurados bastante grande, de tal manera que la computadora pudiera mostrar sentido común. La construcción de dichos sistemas de conocimiento tan generales es una tarea muy grande que podría tomar años o quizá décadas para ser concluida. Como una solución práctica a mediano plazo, proponemos resolver las siguientes tareas con objeto de acercarnos hacia la comprensión de la historia:

1. Identificar a los personajes, lugares y objetos de la historia,
2. Identificar las acciones descritas,
3. Identificar las acciones que no se realizan pero sí se mencionan dentro de la historia,
4. Determinar los argumentos para cada una de estas acciones. Estos argumentos pueden ser vistos como respuestas a preguntas como: quién, dónde, qué, cuándo, por qué y a quién. Cada acción con sus argumentos es una estructura que llamamos *situación*, y
5. Establecer una secuencia temporal entre situaciones que corresponda al flujo de la historia.

Siguiendo este enfoque, podemos decir que para el pasaje anterior del lobo y la oveja:

Los personajes son: Lobo y Oveja

Los lugares son: Río y Río-arriba.

Los objetos son: Agua.

Las situaciones son:

Lobo ve Oveja

Oveja bebe

Oveja está *en* Río

Lobo quiere (Lobo come Oveja)

Lobo está Río-arriba

Lobo acusa Oveja *de* ((Oveja agita Agua) y (Oveja no-permite (Lobo bebe Agua)))

Las situaciones están compuestas por otras sub-situaciones que escribimos entre paréntesis. Note que estas situaciones no necesariamente ocurren. En este caso no ocurre (y aún no sabemos si ocurrirá) que (Lobo come Oveja). Tampoco ocurre que (Oveja agita Agua) ni que (Oveja no deja (Lobo beber Agua)).

Las palabras en mayúsculas apuntan a instancias particulares de personajes, lugares y objetos para esta historia en particular.

2.2.1.1.2 Representación de situaciones con Estructuras con Características Tipificadas (TFS)

Para construir y representar las situaciones, proponemos utilizar Estructuras con Características Tipificadas (Typed Feature Structures, TFS). Este formalismo nos permite cubrir prácticamente cada nivel de descripción lingüística [42], desde la construcción de tipos básicos de la oración (categorías gramaticales), pasando por la construcción de tipos intermedios (por ejemplo, individuos), hasta la representación de las situaciones con sus complementos, y finalmente, la construcción completa de la historia.

Podemos representar una situación como una estructura con características, como se muestra en la Figura 1. Esta representación es una Matriz de Atributo-Valor (AVM). Seguimos la convención de representar atributos en mayúscula y los valores en minúscula.

sit_cosa denota que el valor para QUÉ o POR_QUÉ puede ser una *situación* o una *cosa*. TIEMPO tiene un valor numérico que corresponde a la secuencia en la cual se mencionan las situaciones. OCURRE es la característica que indica si la situación ocurre o no dentro de la historia.

El hecho de que las estructuras con características sean tipificadas nos permite manejar una jerarquía ontológica de objetos, por ejemplo que un hombre es un humano, los humanos son individuos, y por tanto, pueden ser participantes en una acción como valores de QUIÉN o A_QUIÉN.

2.2.1.1.3 Interacción entre sintaxis y conocimiento

$$\begin{bmatrix} \textit{situación} & \\ \text{ACC} & \textit{acción} \\ \text{TIEMPO} & \textit{n} \\ \text{QUIÉN} & \textit{individuo} \\ \text{QUÉ} & \textit{sit\_cosa} \\ \text{DÓNDE} & \textit{lugar} \\ \text{A\_QUIÉN} & \textit{individuo} \\ \text{POR\_QUÉ} & \textit{sit\_cosa} \\ \text{CON\_QUÉ} & \textit{cosa} \\ \text{NEG} & *\textit{boleano}* \\ \text{OCURRE} & \textit{occ} \end{bmatrix}$$

Figura 1: AVM para el tipo *situación*.

Antes de continuar con la explicación de cómo se construyen las situaciones, comentaremos brevemente sobre la interacción entre sintaxis y conocimiento.

Tradicionalmente, las TFS han sido utilizadas principalmente para análisis sintáctico, mientras que se utilizan sistemas basados en marcos para manejar el conocimiento. Ejemplos del uso de dichos formalismos son HPSG [166], un formalismo bien conocido que combina el uso de gramáticas generativas con los beneficios de las TFS, y NEOCLASSIC [143], un sistema de representación del conocimiento basado en marcos.

Creemos que con objeto de construir exitosamente una situación, estas dos etapas tradicionalmente separadas tienen que ser unidas en una sola de tal manera que sea posible la interacción. Es por eso que elegimos un mismo formalismo para representar tanto sintaxis como conocimiento: TFS, pues combina características con los sistemas basados en marcos. Estas características son:

1. Los marcos y las TFS están organizados en jerarquías,
2. Los marcos están compuestos de *ranuras* (equivalentes a los atributos de las TFS) para los cuales se usan *rellenos* (equivalentes a los valores de las TFS o a las referencias a otras TFS como valores) que deben especificarse o calcularse [137].
3. Los marcos y las TFS son declarativos, y
4. La lógica de las TFS es similar a la lógica usada por los marcos: Lógica de Descripción.

Las Lógicas de Descripción (DL por Description Logics) forman parte de las lógicas de primer orden y son utilizadas por los sistemas basados en marcos para el razonamiento. En las DL es posible construir una jerarquía de conceptos a partir de conceptos atómicos y atributos, usualmente llamados *roles* [138]. La única diferencia entre la Lógica de Características (FL por Feature Logics) usada por las TFS y la DL usada por los sistemas basados en marcos es que los atributos de las FL son univaluados, mientras que en las DL los atributos pueden ser multivaluados. Esto podría parecer una diferencia simple, sin embargo podría llegar a constituir la diferencia entre problemas de razonamiento decidibles y no decidibles [139].

2.2.1.1.3.1 Construcción de Situaciones.

El sistema de construcción de conocimiento lingüístico LKB (Linguistic Knowledge Building) es un entorno de programación para TFS. LKB sigue el formalismo tal como fue introducido por Shiber en *una introducción a enfoques basados en unificación para la gramática* [170].

```
n := *top*.

occ := *boolean*.
occ_if := occ &
    [IF situation].

whatsit := situation &
    [WHAT sit_thing].

whysit := situation &
    [WHY  sit_thing].

sit_thing  :=  situation
& thing.

tpi := *top* &
     [ORTH string].

thing := tpi.
place := tpi.
individual := tpi.
action := tpi.

situation := *top* &
    [ACT action,
     TIME n,
     WHO individual,
     WHERE place,
     WITH thing,
     NEG *boolean*,
     WHOM individual,
     OCCURS occ].

story := *top* &
     [INDIVIDUALS
*list*,
      PLACES *list*,
      OBJECTS *list*,
      ARGS *list*].
```

Figura 2: Tipos en LKB para representar situaciones e historias

Aunque LKB ha sido probado principalmente con gramáticas basadas en HPSG [166] como ERG del proyecto LinGO [56], LKB ha sido creado con el propósito de ser independiente de algún entorno de trabajo [55].

LKB se utiliza principalmente para el análisis sintáctico de oraciones. Sin embargo, LKB puede ser utilizado no sólo para analizar oraciones, sino también para almacenar e interactuar con conocimiento.

Actualmente, para tratar con una representación semántica los autores de programas en LKB hacen uso de una gramática que tiene marcadores especiales (como LISZT y HANDEL) para posteriormente construir una representación en Semántica de Recursión Mínima (MRS por Minimal Recursion Semantics) [58], lo cual produce una representación lógica plana cuyo propósito principal es manejar fenómenos lingüísticos de transferencia y cuantificación. La salida de MRS es adecuada principalmente para traducción, y ha sido probada con éxito en el proyecto Verbmobil [41]. Sin embargo, para nuestros propósitos MRS no considera información sintáctica específica que nos permita identificar el rol gramatical que cada constituyente juega, haciendo difícil determinar si un constituyente corresponde al atributo para QUIÉN o A_QUIÉN, por ejemplo.

Como propusimos en la Sección 2.2.1.1, usaremos estructuras con características tipificadas para construir una situación. La construcción de situaciones corresponde a cada oración.

Para manejar *situaciones* como TFS en LKB, establecemos los tipos mostrados en la Figura 2 con su jerarquía correspondiente. Usamos la notación estándar de LKB para listas, TFS y unificación (&). Asumimos que los tipos encerrados en asteriscos están predefinidos, siendo *top* el tipo más general en la jerarquía de tipos.

Para ilustrar cómo se construyen las situaciones, usaremos el fragmento de texto del lobo y la oveja tomado de *Un entorno para representar el conocimiento* [129], el cual reproducimos aquí.

"Había una vez un lobo que vio a una oveja bebiendo en un río y quería una excusa para comerla. Para tal propósito, a pesar de que se encontraba río arriba, acusó a la oveja de agitar el agua e impedir que bebiera de ella"

El ejemplo que presentamos en esta sección ilustra la construcción de situaciones. Es por esto que detalles más particulares acerca de la sintaxis y otros fenómenos no son cubiertos a detalle. El análisis sintáctico mostrado en este ejemplo se realiza mediante coincidencia de patrones. Para sistemas a más grande escala, pueden utilizarse formalismos como HPSG, puesto que son relativamente fáciles de combinar debido a que ya utilizan TFS.

Para construir una situación, asumimos que el sistema tiene información previamente acerca de los posibles roles que cada entidad puede tener. Por ejemplo, Río y Río-arriba son lugares, Lobo y Oveja son individuos, y Agua es un objeto. Vea la Figura 3.

Las entidades pueden estar formadas por más de una palabra. No sabemos nada *a priori* de cualquier propiedad de estas entidades (como por ejemplo Oveja *grande*, Oveja *llamada* Dolly, etc.). Estas propiedades serán llenadas a medida que se analiza la historia.

El hecho de que las situaciones ocurran o no, es importante para entender el flujo de la historia. En el ejemplo del lobo y la oveja, no ocurre realmente que la oveja agita el agua y evita que el lobo tome agua. Esta es una situación mencionada como una consecuencia de que el lobo quiere una excusa para hacer algo. Para definir si una situación ocurre o no, consideramos entonces que cuando una situación está subordinada (o está contenida dentro) por una situación, la situación subordinada no ocurre.

Analizaremos el fragmento de la historia presentado anteriormente palabra a palabra siguiendo un orden específico. Note que la Lógica de Características (Feature Logic – FL) es declarativa, así que este análisis podría ser realizado en cualquier orden conduciendo a los mismos resultados.

Comenzaremos con la oración (1):

`Había una vez un lobo que vio a una oveja bebiendo en un río y quería una excusa para comérsela` (1)

Las primeras palabras de (1) (mostradas en (2) más adelante, coinciden con un patrón que introduce a `lobo` como un *individuo*. Este patrón es: `había + una + vez + un(a) + individuo`, lo cual nos lleva a la representación mostrada en (3), líneas adelante:

`Había una vez un lobo` (2)

`[individuo`

```
 NOMBRE lobo                                                    (3)
 ORT "lobo"]
```

Esta estructura puede unificarse con la estructura correspondiente en la base de conocimiento (implementada como TFS) para encontrar las posibles propiedades de los lobos en general.

Para evitar escribir de nuevo las TFS que vamos identificando en este análisis, escribiremos una referencia a ellas. Siguiendo la notación de LKB las etiquetas comienzan con el signo #.

```
#lobo [individuo
      NOMBRE lobo                                               (4)
       ORT "lobo"].
```

Así que podemos reescribir la oración siendo analizada (1) como (5):

```
#lobo que vio a una oveja bebiendo en un río                    (5)
```

Una TFS del tipo *individuo* seguida por el lexema *que*, hace que *que* absorba al individuo. La regla que hace esto se muestra en (6):

```
individuo_que := individuo &
 [ NOMBRE #1,
   ORT #2,                                                      (6)
   ARGS < individuo & [NOMBRE #1,
          ORT #2],
          lexema_que]>].
```

La oración es ahora:

```
#lobo vio una oveja bebiendo en un río                          (7)
```

Consideremos ahora parte de *una oveja bebiendo en un río*. Esta es otra situación, pero primero debemos añadir al individuo `oveja` a nuestra historia.

```
#oveja [individuo
       NOMBRE oveja                                             (8)
       ORT "oveja"].
```

`una oveja bebiendo en un río` se convierte entonces en (9):

```
#oveja bebiendo en un río.                                      (9)
```

El léxico previamente definido (vea la Figura 3) provee la información de que `río` puede ser un lugar. `río` no está restringido a pertenecer a una sola categoría; en caso de que exista más de una opción, la unificación se encargará de elegir aquella(s) que sea(n) correcta(s). `río` es considerado entonces como:

```
#río [lugar
        NOMBRE río                                                    (10)
        ORT "río"]
```

Con respecto al mecanismo de resolución de referencias que establecería la diferencia entre *un río* y *el río* de acuerdo a entidades previamente introducidas, asumimos que cada vez que se menciona una entidad, su equivalente en TFS se añade. Cuando se forma la supra-estructura de la historia, dos TFS que corresponden a la misma entidad se unificarán. Si dos TFS del mismo tipo tienen características particulares que entran en conflicto (como *río rojo* y *río azul*), la unificación fallará, y entonces se considerarán dos entidades diferentes. Detalles específicos pueden consultarse en la Sección 2.2.1.2.

#río puede unificarse posteriormente con una base de conocimientos de tal manera que el sistema podría inferir que #río está hecho de agua. Para efectos de claridad, en este ejemplo asumimos que este tipo de información no ha sido implementado.

Regresando a nuestro análisis de (9), podemos verificar en nuestro léxico que `bebiendo` se unifica con el tipo *acción* (verbo). Llamaremos a la TFS de esta acción en particular, #beber (12), de tal manera que obtenemos (13).

```
#beber [acción
        NOMBRE beber                                                  (12)
        TIEMPO gerundio
        ORT "bebiendo"]
```

```
#oveja #beber en #río                                                 (13)
```

Ahora podemos aplicar la regla de TFS que crea una situación cuando se encuentra la secuencia: `individuo, acción` en `lugar:`

```
[ situación
  ACTO #2
  QUIÉN #1                                                            (14)
  QUÉ
  DÓNDE #3
  ARGS <#1, #2, lexema_en, #3 >]
```

Las excepciones a la regla (14) pueden ser manejadas con reglas de restricción adicionales. Al aplicar esta regla tenemos la situación #s2:

```
#s2 [ situación
      ACTO beber
```

```
#lobo [individuo]
#oveja [individuo]
#río [lugar]
#río-arriba [lugar]
#agua [objeto]
```

Figura 3: Entidades del léxico consultadas

```
QUIÉN #oveja                                                    (15)
QUÉ
DÓNDE #río]
```

Si regresamos a la oración principal (7), y sustituimos la última situación que acabamos de encontrar tenemos:

```
#lobo vio #s2.                                                  (16)
```

Y esto forma otra situación:

```
#s1 [ situación
      ACTO ver                                                  (17)
      QUÉ #s2 ]
```

Finalmente, la primer oración es una situación.

```
#s1                                                             (18)
```

`#s1` tiene a `#s2` como situación subordinada. El resto del fragmento de la historia del lobo y la oveja puede ser analizado de manera similar. Las entidades consultadas del léxico se muestran en la Figura 3, y la estructura de la historia obtenida después de este análisis se muestra en la Figura 4.

2.2.1.1.4 Los marcos de Minsky y las Situaciones

Minsky expone en *Un entorno para representar el conocimiento* [129] que los marcos son como una red de nodos y relaciones; los niveles superiores de un marco están fijos y representan cosas que siempre son ciertas con respecto a una situación supuesta. Los niveles inferiores tienen terminales (ranuras) que deben ser llenadas con instancias específicas de datos. Existen condiciones especificadas por marcadores que requieren que el relleno asignado a una ranura sea una persona, un objeto, o un apuntador a un sub-marco de cierto tipo. Un terminal que ha adquirido un marcador de *persona femenina* rechazará asignaciones de pronominal *masculino*. En este sentido los marcos de Minsky son muy parecidos a una TFS.

```
historia & [
                INDIVIDUOS    <#lobo & lobo1, #oveja & oveja1>,

                LUGARES       <#río & río1, #río-arriba &
                                          río-arriba1>,

                OBJETOS       <#agua & agua1>,

                SITUACIONES   <#S1 [situación      #S2 [situación
                                    TIEMPO 1          TIEMPO 1
                                    ACTO ver          ACTO beber
                                    QUIÉN #lobo       QUIÉN #oveja
                                    QUÉ #s2           QUÉ (líquido)
                                    OCURRE sí],       DÓNDE #río
                                                      OCURRE sí],

                              #s3 [situación      #s4 [situación
                                   TIEMPO 2           TIEMPO 2
                                   ACTO querer        ACTO comer
                                   QUIÉN #lobo        QUIÉN #lobo
                                   QUÉ #s4            QUÉ #oveja
                                   OCURRE sí],        OCURRE no],

                              #s5 [situación      #s6 [situación
                                  TIEMPO 3            TIEMPO 4
                                  ACTO estar          ACTO acusar
                                  QUIÉN #lobo         QUIÉN #lobo
                                  DÓNDE #río-arriba   QUÉ #s8
                                  OCURRE sí],         POR_QUÉ #s3
                                                      OCURRE sí],

                              #s7 [situación      #s8 [situación
                                   TIEMPO 4           TIEMPO 4
                                   ACTO acusar        ACTO agitar
                                   QUIÉN #lobo        QUIÉN #oveja
                                   QUÉ #s9            QUÉ #agua
                                   POR_QUÉ #s3        OCURRE no],
                                   OCURRE sí],

                              #s9 [situación      #s10 [situación
                                   TIEMPO 4           TIEMPO 4
                                   ACTO no-dejar      ACTO beber
                                   QUIÉN #oveja       QUIÉN #lobo
                                   QUÉ #s10           QUÉ (líquido)
                                   OCURRE no],        OCURRE no] >]
```

Figura 4: TFS para el fragmento de la historia del lobo y la oveja

Cada marco puede ser visto como una TFS, siendo las ranuras los atributos de la estructura atributo-valor. Sin embargo, existe una diferencia importante entre los marcos de Minsky y nuestro punto de vista sobre la representación de las situaciones: Minsky habla de los marcos como una estructura de datos para representar una situación estereotipada, como estar en un tipo especial de cuarto, o ir a una fiesta infantil. Minsky considera que los marcos deben contener información acerca de cómo deben ser usados, información acerca de lo que se espera e información de qué hacer si lo que se espera no se cumple. En contraste, nosotros consideramos que una situación es una unidad simple

transitoria del estado de las cosas dentro de una historia. Considere la oración *El hombre quiere bailar con María*. Esta oración contiene dos situaciones: (Situación 1) *quiere* ¿Quién? *El hombre,* ¿Qué? – Se refiere a la Situación 2. ¿Ocurre? – Sí. (Situación 2) *bailar*. ¿Quién? *El hombre* (el mismo hombre), ¿Con quién? *María*. ¿Ocurre? – No. En estas situaciones no consideramos (en contraste con Minsky) información acerca de cómo usar un marco, información acerca de lo que se espera ni qué hacer si no se confirma aquello que se espera.

2.2.1.1.5 Resumen

El formalismo de Estructuras con Características Tipificadas (TFS) permite su utilización a distintos niveles para la representación de una historia. Las TFS son un formalismo bien estudiado que garantiza la computabilidad de su lógica. El trabajo preliminar presentado aquí contiene ideas útiles para extraer situaciones de historias circunscritas de tal forma que posteriormente es posible hacer preguntas simples sobre el texto como quién hizo algo, o dónde alguien hizo algo. Esto puede ser usado en un sistema de búsqueda de respuestas para encontrar resultados relevantes acerca de eventos descritos en una historia.

La resolución de referencias fue considerada de forma muy ligera en esta sección, sin embargo, un mecanismo fuerte para resolución de referencias es esencial para una operación a mayor escala del sistema. Este tema se tratará en la siguiente sección.

Con respecto al enfoque de los marcos de Minsky, como un trabajo futuro, a través del análisis de individuos a través de una historia, la conducta de los personajes podría ser generalizada en un modelo para predecir sus reacciones e interacciones, lo cual tendería a la adquisición del sentido común, y saber lo que se espera en el sentido de los marcos de Minsky.

2.2.1.2 El problema de la continuidad de entidades en TFS al representar un texto

En la sección anterior hemos descrito un mecanismo en el que interactúan sintaxis y conocimiento en una representación de estructuras con características tipificadas (Typed Feature Structures, TFS). En esta sección mostraremos cómo se mantiene la continuidad entre entidades en una representación TFS, particularmente HPSG mediante el uso de la base de conocimientos que va creándose conforme se analiza el texto. Esta base de conocimientos puede usarse para su consulta en un análisis subsecuente para resolver correferencia entre oraciones.

2.2.1.2.1 Introducción

Entender un texto implica identificar las entidades descritas en él, las propiedades de estas entidades y las situaciones en las que éstas participan. Las gramáticas modernas usualmente especifican esta

información usando índices de referencias. Por ejemplo, en HPSG la entrada para el verbo *give* 'dar' en inglés se define como sigue [166]:

$$\left[\begin{array}{ll} \textit{dtv-lxm} & \\ \text{ARG-ST} & \left\langle [\,]_i, [\,]_j, [\,]_k \right\rangle \\ \text{SEM} & \left[\begin{array}{ll} \text{INDEX} & s \\ \text{RESTR} & \left\langle \left[\begin{array}{ll} \text{RELN} & \text{give} \\ \text{SIT} & s \\ \text{GIVER} & i \\ \text{GIVEN} & j \\ \text{GIFT} & k \end{array}\right] \right\rangle \end{array}\right] \end{array}\right]$$

Aquí, en la sección semántica de la definición (SEM), se hace referencia a las entidades participantes de la situación *s* con los índices *i*, *j* y *k*. Diferentes entidades son referidas por índices diferentes, y la misma entidad por el mismo índice (correferencia).

Sin embargo, hasta donde sabemos, las implementaciones de este tipo de gramáticas mantienen dicha correspondencia sólo dentro de una oración. Para cada nueva oración la cuenta de los índices se reinicia desde uno, de tal manera que la correspondencia uno-a-uno entre las entidades e índices se destruye: la entidad referida por el índice 1 en una oración no tiene nada que ver con la otra referida por 1 en la siguiente oración. Para mantener la coherencia semántica en el discurso, es importante correlacionar todos los índices que hacen referencia a la misma entidad a través del texto.

Para esto, proponemos un mecanismo que crea y mantiene estructuras semánticas separadas del análisis de la oración. Mantenemos estas estructuras en una base de conocimientos que se construye mientas se analiza el texto. En esta base, las estructuras se encuentran en correspondencia uno-a-uno con las entidades mencionadas en el texto.

Aparte de representar la semántica del texto, esta base de conocimientos puede ser consultada durante el análisis de texto para resolver la correferencia: cuando una entidad con ciertas propiedades se menciona en el texto en el contexto que implica correferencia (por ejemplo, en un artículo definido), podemos buscar en la base de datos una entidad que corresponda con la misma o con propiedades compatibles. Se aplican varias heurísticas y se consideran diferentes fuentes de evidencia para hacer una decisión final sobre la presencia o ausencia de correferencia.

El resto de esta sección está organizada como sigue: En la subsección 2.2.1.2.2 consideramos el formalismo de Estructuras con Características Tipificadas (TFS) para hacer la representación tanto del análisis sintáctico como del conocimiento. En la subsección 2.2.1.2.3 explicamos los contenidos deseados de nuestra base de conocimientos. En la subsección 2.2.1.2.4, discutimos cómo se construye. Finalmente, en la subsección 2.2.1.2.5 señalaremos los puntos más importantes de este tema.

2.2.1.2.2 TFS como representación del conocimiento

Puesto que queremos combinar la funcionalidad de gramáticas del tipo HPSG con una base de conocimientos, es deseable usar un solo formalismo tanto para el análisis oracional como para la representación del conocimiento.

Para la representación del conocimiento se usan tradicionalmente sistemas basados en Lógica de Descripción (DL). Estos sistemas se conocen también como Sistemas de Lógica Terminológica, o sistemas parecidos a KL-ONE. Ejemplos de estos sistemas son: NEOCLASSIC [143], BACK [97], CRACK (online) [26], FaCT [10], LOOM [118], y RACER [91]. Sin embargo, dicho formalismo no está diseñado para análisis oracional.

Por otra parte, el mismo formalismo (TFS, estructuras con características tipificadas) que es utilizado en gramáticas del tipo HPSG para análisis de oraciones, puede ser utilizado para representación del conocimiento, específicamente para construir la base de conocimientos mientras se hace el análisis de textos e incluso para implementar mecanismos simples de razonamiento [42]. La lógica implementada con TFS mediante la unificación es conocida como Lógica de Características (Feature Logic, FL). Así como en DL, en FL es posible construir una jerarquía de conceptos a partir de conceptos atómicos llamados usualmente *roles* en DL. La diferencia principal entre FL y DL es que los atributos de FL tienen un solo valor, mientras que los atributos en DL puede tener diversos valores [138]. Sin embargo esta es una pequeña diferencia puesto que en FL los atributos puede ser lista de valores.

Consideremos un ejemplo que muestra similitud entre FL y DL. En el sistema DL NEOCLASSIC, uno puede crear un individuo utilizando la función `createIndividual (sandy persona)`. Esto puede ser representado como la TFS `sandy`, un subtipo de `persona`. Posteriormente, `addToldInformation (sandy fills (has vestido))` puede ser visto como una operación en la TFS `sandy` que le añade una característica. La TFS resultante es

```
sandy
[HAS dress].
```

Para recuperar la información se utiliza `getInstances (<concepto>)`, donde `<concepto>` puede ser por ejemplo, `TIENEVESTIDO` definido como `createConcept (TIENEVESTIDO fills (has vestido))`. De esta manera el comando `getInstances (TIENEVESTIDO)` es equivalente a la unificación de todas las instancias disponibles con la TFS `[HAS vestido]`, obteniendo la TFS completa `sandy`.

2.2.1.2.3 Estructura de la base de conocimientos de TFS

Ahora que hemos mostrado que las TFS pueden ser utilizadas para representación del conocimiento, podemos discutir la estructura de la base de conocimientos que queremos obtener de un texto.
Una gramática tipo HPSG tiene un léxico que relaciona una cadena de palabras con los tipos (categorías gramaticales) que pueden ser asignadas a la cadena. La combinación gradual de los términos construye categorías gramaticales, que a su vez forman sintagmas.

En un punto apropiado, convertimos éstos términos en entidades: representaciones TFS de objetos animados o inanimados, reales o abstractos. Estas entidades se añaden a la base de conocimientos.

La Figura 5 muestra las estructuras resultantes en la base de conocimientos después de analizar el fragmento de texto:

Hay un librero$_i$ grande rojo en la sala$_j$. Los libros$_i$ de Juan$_j$ están bien puestos en él$_j$. (1)

Las referencias a las entidades se marcan aquí con los índices correspondientes al análisis de la oración, los cuales hacen clara la correspondencia entre las estructuras TFS y el texto. Sin embargo estos índices no apuntan a cadena de letras, son números secuenciales producidos por el análisis de la oración. En la Figura 5, las cadenas como *Juan*, *librero*, o *él* se incluyen sólo para mantener la claridad y no son parte propiamente de las estructuras.

El nombre de una TFS se forma por su tipo indexado por un número. Ent_0, ..., Ent_3 son entidades; note que almacenan el rol que se les ha dado cuando son utilizadas en situaciones. En la figura, ACT quiere decir acción, REF quiere decir referencia, ADVG quiere decir grupo adverbial, ADV adverbio, y ADJ adjetivo.

S_0 y S_1 son situaciones. La situaciones son formadas por los atributos ACT (acción), QUIÉN, QUÉ, DÓNDE, A_QUIÉN, POR_QUÉ, y CON_QUÉ, entre otros.

Note que algunas veces la relación semántica no puede ser fácilmente obtenida del contexto inmediato. Por ejemplo el texto *Los libros de Juan* no necesariamente hace referencia a los libros

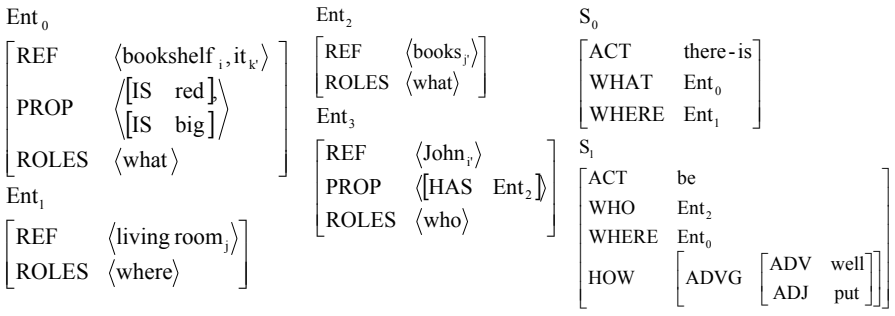


Figura 5. Estructuras TFS extraídas del texto (1)

que Juan *tiene*, sino quizás aquéllos que él ha *escrito*. Una forma posible de resolución de dichas ambigüedades es preguntar a un operador humano; pueden utilizarse otras formas, las cuales no discutimos aquí.

La representación semántica del texto como en la Figura 5, permite la búsqueda de las entidades mencionadas en él a partir de un conjunto dado de propiedades. Esto puede ser utilizado tanto en resolución de co-referencias durante el análisis de texto y en búsqueda de respuestas. Por ejemplo, un sistema de diálogo puede permitirle a un usuario preguntar: *¿Hay un librero en la sala?¿Dónde están los libros de Juan?* Las respuestas pueden ser fácilmente encontradas al unificar las propiedades mencionadas en las preguntas con los objetos en la base de conocimientos.

2.2.1.2.4 Construcción de la base de conocimientos TFS

En la sección anterior hemos visto cómo cuatro entidades (*un librero, la sala, los libros, y Juan*) extraídas de dos oraciones pueden ser representadas. Ahora describiremos brevemente el mecanismo que permite a un sistema de análisis TFS construir y utilizar dichas representaciones.

Para mantener la base de conocimientos, utilizaremos tres funciones: `INTRO`, `ADD`, y `GET`. Los objetos introducidos en la base con la función `INTRO` persisten en un alcance más amplio que una sola oración y pueden ser modificados con `ADD` o recuperados con `GET` mientras se analizan otras oraciones. Puesto que las entidades se representan como TFS, la unificación es la única operación que utilizaremos en estas funciones.

2.2.1.2.4.1 Función ***INTRO***

Añade una entidad a la base de conocimientos. Regresa un apuntador a la entidad recientemente añadida. Este apuntador se utiliza como un término en las reglas TFS. El argumento para `INTRO` es una <`descripción_TFS`>. Una <`descripción_TFS`> es una TFS con la especificación de los valores de los atributos en la notación `ATTRIBUTO:valor`. Esta especificación puede estar incompleta (under-specified). Si algún valor es otra TFS lo incluiremos entre paréntesis.

`INTRO` es similar a `createConcept` de NEOCLASSIC. Un ejemplo de utilización de esta función es:

```
INTRO(IND:(REF:i))
```

en este ejemplo, `i` será tomada como el índice del individuo actual como su característica `REF`.

2.2.1.2.4.2 Función ***ADD***

Con esta función podemos añadir atributos de las entidades previamente creadas en la base de conocimientos. El argumento de `ADD` es una <`descripción_TFS`>, donde la TFS de la descripción TFS es una TFS instanciada previamente creada en la base de conocimientos con `INTRO`. Para esta función, la <`descripción_TFS`> debe estar completa, es decir, todos los valores en todos los niveles deben estar especificados. Si la entidad referenciada por la <`descripción_TFS`>, ya tiene previamente llenado el atributo que nosotros pretendemos añadir, pero su valor es distinto, el valor del atributo se convertirá en una lista que contiene ambos valores. Si el valor del atributo ya es una lista y no contiene el valor que estamos añadiendo entonces el nuevo valor se agrega.

Los elementos de la lista pueden ser seleccionados posteriormente mediante métodos estándares de unificación sin tomar en cuenta el orden de estos elementos dentro de la lista. Un ejemplo de esta función que añade un adjetivo como propiedad de un individuo es:

```
ADD(IND:(PROP:(ADJ))
```

2.2.1.2.4.3 Función GET

Esta función regresa la entidad por las entidades que significan con la <`descripción_TFS` > provista. Por ejemplo,

```
GET(S:(WHO:IND1,WHOM:IND2))
```

obtiene todas las situaciones donde el agente se unifica con `IND1`, y el beneficiario se unifica con `IND2`. `IND1` e `IND2` son términos que corresponden a entidades específicas derivadas de un análisis previo.

2.2.1.2.5 Resumen

En esta sección hemos sugerido utilizar una base de conocimientos con objetos persistentes (entidades y situaciones) para mantener la co-referencia entre oraciones en formalismos gramaticales tipo HPSG. La base de conocimientos se construye a partir del texto en una manera semiautomática. Las entidades están disponibles durante el análisis completo del texto (el lugar de sólo una oración) y pueden ser utilizadas también después de que el texto ha sido analizado, por ejemplo para búsqueda de respuestas o como una representación semántica del texto.

2.2.2 Análisis mediante reglas de reescritura

En esta sección presentamos el resultado de aplicar análisis mediante reglas de reescritura como parte del estudio del estudio del arte de la comunicación humano-máquina. Para efectos de claridad, situaremos este estudio en situaciones del mundo real como un diálogo que consiste en las solicitudes del usuario que piden información o la realización de una acción, seguida de la respuesta de la computadora que puede ser una respuesta verbal o la ejecución de la tarea solicitada. Las reglas de reescritura serán exploradas dentro de un sistema que maneja este tipo de interacción a través de una gramática de reglas de reescritura con modificación de propiedades, substitución de comodines y funciones en línea. Nuestro objetivo es producir una o más instrucciones de computadora específicas que se derivan de la solicitud del usuario. Los objetos referenciados dentro de la solicitud son traducidos por símbolos por las reglas gramaticales y el uso de objetos especiales de contexto llamados escenas. Después de esto, las instrucciones son ejecutadas por un sistema externo. Presentamos ejemplos para la tarea de colocar objetos tridimensionales usando instrucciones en español.

La Composición Espacial de Objetos (CEO) se refiere a manipular piezas físicas o virtuales prefabricadas (como partes de muebles) para ensamblarlas creando nuevos objetos o escenas (como el diseño de oficinas). Existen muchas aplicaciones computacionales que tratan con la CEO, por ejemplo, los sistemas para diseño asistido por computadora de un cuarto o una casa. Los objetos a ser colocados en el cuarto están predefinidos (muebles, puertas, ventanas, etc.) y pueden ser seleccionados de un catálogo para ser colocados en una escena virtual donde el usuario desea colocarlos.

Obviamente, la CEO no está limitada al diseño de casas. Puesto que vivimos en un mundo espacial de objetos descomponibles, existen muchas aplicaciones de este tipo. Por ejemplo, supone que un usuario quiere construir un librero. Para esto selecciona primero unos tablones de un catálogo de tablones prefabricados y después une estas partes hasta que se construye el librero como él o ella desea. Este es un ejemplo de cómo se puede construir un nuevo objeto.

Por su naturaleza, dichos sistemas deben estar diseñados para no requerir ningún conocimiento o habilidades relacionadas con computadoras. Es por ello que sus interfaces deben ser intuitivas y no deben requerir ningún entrenamiento. Una forma perfecta de dicha interacción es el lenguaje natural en la misma forma en la que se utilizaría para interactuar con un trabajador humano. De hecho, la interacción humano–computadora en dichos sistemas es principalmente imperativa: el usuario da un comando y la computadora ejecuta la tarea requerida. Estos comandos pueden ser dados en lenguaje

natural utilizando tiempo verbal interactivo. De aquí nuestra motivación para desarrollar un entorno para la integración de interfaces de lenguaje natural con sistemas CEO.

Dentro del entorno de trabajo propuesto es posible traducir la oración de entrada *¿Podrías poner la silla junto a la mesa, por favor?* En una secuencia de comandos directamente interpretados por el motor del sistema: `move(obj_chair1,getpos(obj_table1)++)` [5], lo cual significa que el sistema debe colocar `obj_chair1` (la silla a la cual el usuario hace referencia) en la posición más cercana disponible cerca de la posición de `obj_table1` (la mesa que está previamente en la escena). Realizamos esto al transformar la oración original paso a paso como sigue:

¿Podrías poner la silla junto a la mesa, por favor?

Podrías poner la silla junto a la mesa

poner la silla junto a la mesa

poner obj_chair1 junto a la mesa

poner obj_chair1 junto a obj_table1

poner obj_chair1 junto_a obj_table1

poner obj_chair1 (getpos(obj_table1)++)

move (obj_chair1,(getpos(obj_table1)++))

Un sistema de requerimiento de acción es un sistema integrado en el cual un usuario interactúa con una computadora haciéndole preguntas o solicitando acciones a ser realizadas dentro de un área o tema específico. La computadora responde o lleva a cabo la acción solicitada. Este tipo de sistema tiene un dominio específico y un número limitado de objetos y acciones que pueden ser realizados sobre ellos. Esta limitación nos permite pensar en una interfaz de lenguaje natural donde las acciones solicitadas y las preguntas hechas se plantean al sistema en una forma libre y natural. El sistema debe reaccionar a estas solicitudes realizando lo que se le pide de una manera coherente. De esta forma, el usuario puede verificar que el sistema ha entendido su solicitud.

Nuestro objetivo es distinto al de las gramáticas generativas, creadas para verificar si las oraciones de cierto lenguaje están bien formadas [166]. Es por esto que proponemos un formalismo diferente que permite a un sistema mostrar su *comprensión* del lenguaje natural llevando a cabo lo que se le

---

[5] Aquí, `obj_` es un objeto, `getpos` es una function que obtiene la posición del objeto, y `++` es la operación que cambia una posición a la siguiente disponible más cercana.

pide que haga. Este formalismo es una gramática de reglas de reescritura con modificación de propiedades, sustitución de comodines y funciones en-línea. El propósito de esta gramática es reducir las expresiones del lenguaje a una forma lógica (un conjunto de instrucciones) aplicando directamente las reglas de reescritura sobre las expresiones del usuario.

Para ilustrar cómo funciona este sistema, asumimos una tarea donde objetos geométricos como esferas, toroides, planos y otras figuras geométricas pueden ser combinados en un lienzo tridimensional. Esta tarea gráfica fue elegida porque permite que la representación de los objetos sea compartida entre el humano y la computadora al ser visualizada. Esta es una característica importante que nos permite tener un contexto común para el usuario y el sistema. Sin esta característica particular, la búsqueda de objetos referenciados no sería posible tal como lo explicamos más adelante.

Adicionalmente, una tarea visual como ésta nos permite crear objetos de una mayor complejidad a través de geometría constructiva de sólidos (Constructive Solid Geometry) usando operadores como unión, intersección, diferencia y combinación (merge) [87]. De esta manera pueden ser creados nuevos objetos. Los objetos pueden tener propiedades como textura, reflectividad, índice de refracción, etc., que pueden ser modificados a través de lenguaje natural.

Como ya vimos en la sección de motivación, uno de los primeros sistemas que manejaba objetos mediante lenguaje natural era SHRDLU, de 1970. Nuestro objetivo no es revivir a SHRDLU en particular, sino presentar un sistema que puede manejar tareas que involucran objetos en un entorno espacial e instrucciones en lenguaje natural de manera genérica.

Como un ejemplo del tipo de expresiones que el usuario puede plantearle al sistema, considere la siguiente solicitud: *¿Me puedes poner el tercero junto al toroide?* 'Can you put the third one next to the torus?'. Después de aplicar 7 reglas de reescritura, que detallaremos más adelante (Sección 2.2.2.6), obtenemos las siguientes instrucciones:

move`(objcat021_03, getpos(obj002)); sos.push(obj003); oos.push(obj002).`

La instrucción `move(objcat021_03)` es llevada a cabo por una función externa que mueve el objeto solicitado. `sos.push(obj003)` y `oos.push(obj002)` son funciones en línea usadas para manejo del objeto y contexto. Estas funciones las detallaremos en la sección 2.2.2.4.2.6. En las siguientes secciones detallaremos la gramática, después el mecanismo de manejo de objetos y contexto, y por último, presentaremos un ejemplo de cómo procesamos las 4 oraciones.

2.2.2.1 Trabajo relacionado

Históricamente los primeros sistemas con una interfaz en lenguaje natural fueron desarrollados en una base *ad hoc* para una aplicación específica. Algunos ejemplos de estos sistemas son

- DEACON (Direct English Access Control), un sistema para responder preguntas [59];
- SHRDLU, que le permitía al usuario mover bloques geométricos virtuales utilizando comandos verbales [194];
- LUNAR, que permitía consultar una base de datos de rocas lunares [196],
- LADDER, que respondía preguntas en inglés acerca de datos de logística naval [93].

Debido a que el modelo del mundo que manejan está interconstruido en la operación de estos programas, cambiar el dominio de aplicación para estos sistemas es un proceso caro y complicado.

Posteriormente surgieron otros sistemas con una interfaz en lenguaje natural diseñados para un ámbito de aplicación más amplio aunque estaban orientados principalmente a la recuperación de formación en bases de datos, por ejemplo: INTELLECT [92], TEAM [89], JANUS [192], y SQUIRREL [8].

Existen trabajos desarrollados recientemente que pueden manejar lenguaje imperativo para múltiples propósitos. Por ejemplo KAIRAI (que significa 'títere') tiene diversos robots virtuales (avatares) que pueden mover hacia delante, girar o empujar un objeto [171, 5]. Al manipular los robots usando estos comandos, el usuario puede mover y colocar los objetos en el mundo virtual. Este sistema fue desarrollado para el japonés. Un sistema similar AnimAL utiliza una interfaz en lenguaje natural para controlar los movimientos de un avatar en un entorno virtual [66, 67, 189]. Di Eugenio [66, 67] consideró el problema de entender frases de la forma *hacer x para hacer y* como en *corta un cuadrado en mitades para hacer dos triángulos*. No tenemos noticia, sin embargo, de trabajos recientes específicamente dedicados a proveer una interfaz de lenguaje natural para sistemas CEO en general.

2.2.2.2 Características de los sistemas CEO

Los sistemas CEO en general restringen el uso de lenguaje natural en diversas maneras. En nuestro entorno de trabajo, nos basamos en estas restricciones para simplificar los mecanismos correspondientes. Específicamente los sistemas CEO tienen las siguientes características relevantes para diseñar interfaz en lenguaje natural:

1. **Los objetos tienen objetos básicos definidos que pueden ser utilizados para construir nuevos objetos.** Esto nos permite comenzar con un conjunto reducido de nombres de objetos a ser reconocidos.
2. **Los objetos tienen propiedades,** mediante las cuales pueden ser referidos, por ejemplo *tablón rojo* en oposición a *tablón verde*. Las propiedades nos permiten mantener pequeño nuestro conjunto de nombres de objetos.
3. **Existe una representación visual espacial común al usuario y a la computadora.** Con esto, el usuario sabe que los únicos objetos que existen son aquellos que pueden ser observados en los catálogos y en la escena actual. Sólo los objetos observables son relevantes para la tarea de composición.
4. **Los objetos tienen un número limitado de acciones que pueden ser aplicadas a ellos.** Estas acciones corresponden a instrucciones de computadora.

El usuario y la computadora manipulan un conjunto finito de objetos con propiedades y acciones pertenecientes a estos objetos. Para diseñar una interfaz de lenguaje natural adecuada, debemos encontrar un mecanismo que relaciona oraciones en lenguaje natural con las instrucciones correspondientes de computadora. Esta relación se implementa a través de la Gramática de Traducción Directa presentada en la siguiente sección.

2.2.2.3 Gramática de Traducción Directa

Desde que el modelo transformacional de Chomsky apareció en 1957 [47], diversos modelos dentro del paradigma generativo han sido sugeridos, como la Gramática de Casos (Case Grammar) [73], Gramáticas Funcionales [105], y recientemente, Gramáticas de Estructura de Frase (Phrase Structure Grammars) [78]. Tradicionalmente las gramáticas generativas están diseñadas para modelar el conjunto completo de oraciones que un hablante nativo de un lenguaje natural considera aceptable [150]. Los lingüistas generativos ven al lenguaje como un objeto matemático y construyen teorías similares a los conjuntos de axiomas y reglas de inferencia en matemáticas. Una oración es gramatical si existe alguna derivación que demuestra que su estructura corresponde al conjunto de reglas dado, de manera similar a la que una demostración prueba que una proposición matemática es correcta [195].

Las gramáticas de estructura de frase (Phrase Structure Grammars, PSG), de la cual HPSG [166] es la más conocida, siguen este paradigma generativo. Para analizar una oración, se estructura jerárquicamente para formar árboles de frase-estructura. Las PSG se utilizan para caracterizar estos árboles de frase-estructura. Estas gramáticas consisten en un conjunto de símbolos no-terminales (categorías de estructura de frase como Sustantivo, Verbo, Determinante, Preposición, Sintagma

Nominal, Sintagma Verbal, Oración, etc.), un conjunto de símbolos terminales (elementos léxicos como *comprar*, *Juan*, *comido*, *en*, *el*, etc.), y un conjunto de reglas que relacionan un no-terminal con una cadena de terminales o símbolos no-terminales [103]. Para analizar una oración, deben aplicarse reglas adecuadas a la cadena de símbolos terminales hasta que se alcance al símbolo no-terminal S.

El árbol de estructura de frase obtenido durante este proceso puede ser analizado posteriormente para generar instrucciones de computadora equivalentes a la oración de entrada. Sin embargo, este proceso puede hacerse directamente si cambiamos el propósito de nuestra gramática a aquél de usar las reglas gramaticales para alcanzar instrucciones de computadora directamente en lugar de romper las oraciones de lenguaje natural en categorías gramaticales (estructuras de frase) y luego convertir esta estructura en instrucciones de computadora. De esta manera, nuestro objetivo es distinto a aquél de las gramáticas generativas puesto que no estamos interesados en modelar todo el lenguaje, sino sólo un subconjunto relevante para la tarea del usuario en cuestión.

La gramática que sugerimos para traducir una oración en lenguaje natural a instrucciones de computadora es una gramática de reglas de reescritura con características adicionales para manejar el contexto y la referencia a objetos. A esta gramática la hemos denominado Gramática de Traducción Directa (GTD)

Dentro de la GTD se incluye tratamiento léxico y morfológico, y las categorías utilizadas hacen referencia a conceptos semánticos de las oraciones. Debido a esto, podemos considerar a la GTD como una gramática semántica [30]. En las gramáticas semánticas, la elección de las categorías se basa en la semántica del mundo y del dominio de aplicación deseado, así como en las regularidades del lenguaje. A pesar de que hoy en día no son muy utilizadas, las gramáticas semánticas tienen diversas ventajas como eficiencia, habitabilidad (en el sentido de Watt [188]), manejo de fenómenos del discurso, y el hecho de que son auto-explicativas. Permiten el uso de restricciones semánticas para reducir el número de interpretaciones alternativas que pueden ser consideradas en cierto momento, en contraste con sistemas altamente modulares, que fragmentan el proceso de interpretación.

2.2.2.4 Definición

Definimos una Gramática de Traducción Directa como una lista ordenada de reglas de reescritura que tienen la forma $\alpha \rightarrow \beta$, donde α y β son cadenas que consisten en los siguientes elementos (que explicamos a continuación) en cualquier orden:

1. Palabras de lenguaje natural
2. etiquetas con propiedades
3. comodines
4. nombres de procedimientos externos
5. referencias simbólicas a objetos, y
6. funciones incrustadas para control del contexto y manejo de referencia de objetos.

No se permiten dos o más reglas con el mismo lado α.

2.2.2.4.1 Orden de las reglas.

El procesamiento de las reglas es ordenado. Primero se consideran las reglas con un α que consisten sólo en palabras en lenguaje natural, comenzando con aquellas con un número mayor de palabras. Si no se puede aplicar ninguna regla, el resto de reglas se consideran según el número de elementos que componen α. Las más largas se consideran primero. Esto es debido a que elementos como *tabla roja* deben ser considerados antes de elementos que sólo mencionan *tabla*. De hecho, una cadena más larga de palabras implica una referencia más específica a un objeto.

Cada vez que se aplica una regla, el procesamiento de las reglas reinicia desde el comienzo de la lista en el orden explicado anteriormente.

El proceso termina cuando no se puede aplicar ninguna regla; la cadena resultante es la salida del programa. El proceso de traducción se considera exitoso si la cadena resultante consiste sólo en referencias simbólicas a objetos y nombres de procedimientos externos. Para evitar ciclos infinitos, el proceso se aborta si una regla se aplica más de una sola vez y/o su aplicación resulta en una cadena previamente obtenida. En este caso se considera que la traducción es no exitosa.

2.2.2.4.2 Componentes de las reglas

En esta sección explicamos cada elemento utilizado en las reglas según se listaron en la sección 2.2.2.4.

2.2.2.4.2.1 Palabras de lenguaje natural

Inicialmente, una oración de entrada consiste sólo en palabras. El ejemplo: *pon la silla junto a la mesa*, es una oración compuesta por 7 palabras que será traducida en una secuencia de procedimientos externos. Las palabras son cadenas de letras y no tienen ninguna propiedad.

2.2.2.4.2.2 Etiquetas con propiedades

Las etiquetas con propiedades tienen la forma:

$\delta\{p_1, p_2, ..., p_n\}$,

donde δ es el nombre de la etiqueta y $p_1, p_2, ..., p_n$ son sus propiedades en la forma `nombre:valor`, por ejemplo: `put{C:V, T:IMP}`. En la Tabla 1, presentamos las propiedades más comunes y sus posibles valores.

Tabla 1. Algunas propiedades y sus valores utilizados en los ejemplos.

Nombre	Propiedad	Valores posibles
C	categoría	N (sustantivo), V (verbo), ADJ (adjetivo), ADV (adverbio), PRO (pronombre), DEFART (artículo definido), INDART (artículo indefinido), OBJ (objeto), POS (posición)
G	género	M (masculino), F (femenino), N (neutral)
N	número	S (singular), P (plural)
T	tiempo verbal	PRES (presente), INF (infinitivo), IMP (imperativo), SUBJ (subjuntivo)
S	forma del sujeto	para verbos: el género y número del sujeto
O	forma del objeto directo	para verbos: el género y número del objeto directo
A	forma del objeto indirecto	para verbos: el género y número del objeto indirecto.
Q	cantidad	L, P, R, M, B (muy poco, poco, regular, mucho, bastante)

Esta construcción es similar a las Estructuras con Características (Feature Structures) tradicionales; sin embargo, las Estructuras con Características, según las definió Kay en [105], experimentan

mecanismos de herencia y unificación. Nuestras etiquetas no están relacionadas con dichos mecanismos.

La siguiente regla convierte la palabra *pon* en una etiqueta:

```
pon --> poner{C:V, T:IMP, S:2S, A:1S}
```

Esta regla sustituye todas las ocurrencias de *pon* en la cadena de entrada con la etiqueta `poner{C:V, T:IMP, S:2S, A:1S}` que expresa las siguientes propiedades: categoría es verbo, tiempo imperativo, sujeto segunda persona singular y objeto indirecto (implícito) primera persona singular.

2.2.2.4.2.3 ***Comodines***

Los comodines se definen por una etiqueta seguida opcionalmente de un conjunto de propiedades (según se definen en la Sección 2.2.2.4.2.2) contenida en paréntesis cuadrados:

$$\varphi[p_1, p_2, \ldots, p_n].$$

Los comodines proveen un mecanismo para generalizar una regla para evitar repeticiones redundantes de reglas. Un comodín hace posible aplicar una regla sobre un conjunto de etiquetas que comparten una o más propiedades. El alcance de un comodín se limita a la regla donde aparece.

Un comodín φ empata con una etiqueta δ, si δ tiene todas las propiedades listadas para φ con los mismos valores. Por ejemplo, los siguientes dos comodines: `A[C:V]` y `B[T:IMP, S2S]` empatan con la etiqueta `poner{C:V, T:IMP, S:2S, O:1S}`, pero el comodín `C[C:V, T:PRES]` no empata porque esta etiqueta no tiene la propiedad *Tiempo* con el valor *Presente*.

Cuando se utilizan del lado derecho de una regla, los comodines pueden ser utilizados para modificar propiedades al especificar otro valor para la propiedad que originalmente se empató. Por ejemplo, considere la frase *podrías juntarlo*, que es una forma amable del imperativo *júntalo*. Para transformar la frase a imperativo aplicamos primero las siguientes reglas:

```
podrías --> poder{C:V, T:SUBJ, S:2S}                    (1)
juntarlo --> juntar{C:V, T:INF, O:3SM}                  (2)
```

y luego utilizamos un comodín para transformar todas las construcciones similares a un imperativo. Note el uso del comodín para cambiar la propiedad T de INF a IMP.

```
poder{C:V,T:SUBJ,S:2S} A[C:V, T:INF] --> A[T:IMP]       (3)
```

Al aplicar (3) resulta la siguiente cadena de salida:

```
juntar{C:V,T:IMP,O:3SM}                                 (4)
```

Gracias a los comodines, la regla (3) funciona para cualquier expresión de amabilidad en la forma *podrías* + verbo_en_infinitivo.

Usualmente las propiedades que se encuentran entre paréntesis son utilizadas por el objeto cuyo nombre aparece inmediatamente a la izquierda de dichos paréntesis. En ocasiones es necesario accesar las propiedades de otros objetos fuera de los paréntesis. Esto es posible a través de la notación de punto definida a continuación. Considere la siguiente cadena,

```
juntar{C:V,T:IMP,O:3SM} un poco más
```

la frase *un poco más* puede ser transformada en un adverbio de cantidad por la regla:

un poco más --> x{C:ADV, Q:L}, (5)

que puede ser transformada en la propiedad del verbo por la regla:

A[C:V] B[C:ADV,Q] --> A[Q:B.Q] (6)

que significa: "si un verbo A es seguido por un adverbio B con alguna cantidad, añada a este verbo la propiedad Cantidad con el mismo valor que tiene en B". Esta última construcción se expresa en (6) como `B.Q`, que quiere decir el valor de Q en B.

Si se especifica una propiedad a un comodín sin asignarle valor, esto indica que para encontrar una coincidencia para el comodín, la propiedad debe estar presente, sin importar su valor.

Note que debido a esta capacidad de reemplazamiento, los comodines no cumplen con las propiedades de unificación [108].

2.2.2.4.2.4 ***Procedimientos externos***

Los procedimientos externos con argumentos se forman por el nombre de un procedimiento seguido de una lista de argumentos:

nombre_del_procedimiento (arg_1, arg_2, ... , arg_n),

donde n es un número natural (que puede ser 0; en este caso el procedimiento no tiene argumentos). A diferencia de las funciones, los procedimientos no regresan ningún valor. Se ejecutan por el motor del sistema CEO después de la aplicación exitosa de las reglas sobre una expresión en lenguaje natural. Por ejemplo,

```
move(A,B)
```

es un procedimiento externo que coloca al objeto `A` en posición `B`.

2.2.2.4.2.5 ***Referencias simbólicas a objetos***

Una escena es un objeto compuesto por otros objetos. A su vez, estos objetos pueden estar compuestos de otros objetos. Los catálogos también son objetos, compuestos por elementos que son también objetos.

Esta composicionalidad nos permite establecer contextos anidados para resolver la referencia a un objeto dependiendo de la escena que tiene el foco de atención del usuario en un momento dado. Cada uno de los objetos dentro de la escena tiene propiedades que pueden ser accesadas por nuestras reglas de conversión usando etiquetas.

En contraste con las propiedades gramaticales que son descritas exclusivamente dentro de nuestras reglas de conversión, las propiedades del objeto pertenecen al sistema CEO y pueden variar. Estas propiedades pueden ser, por ejemplo, posición, tamaño, componentes, color, material, densidad, alterabilidad, forma, y un conjunto de acciones que puede ser aplicado al objeto dado.

Las etiquetas que comienzan con `obj_` denotan referencias simbólicas a objetos. Por ejemplo, `obj_box231` hace referencia a una caja en particular en una escena en particular.

2.2.2.4.2.6 ***Funciones incrustadas para manejo de contexto y referencia de objetos.***

Las funciones incrustadas, que son un medio para manejar las referencias a objetos, serán discutidas en la siguiente sección.

2.2.2.5 Referencias a objetos y manejo del contexto

Para cada sustantivo, pronombre, o frase nominal, necesitamos hallar una referencia simbólica única a un objeto particular referido por el usuario. Sin embargo, la misma expresión puede ser usada para hacer referencia a distintos objetos particulares, dependiendo del contexto. Para transformar una expresión en una referencia simbólica, primero debemos determinar su contexto [147].

Para manejar el contexto consideramos el contexto como un objeto (llamado objeto escena) que contiene otros objetos. Similarmente a SQUIRREL [8] en nuestro modelo el contexto y la referencia a objetos son manejados por pilas. Nosotros empleamos tres pilas en lugar de una sola: pila de sujeto-objeto (pso), pila de objeto-objeto (poo) y pila de contexto (escena) (ss)

Un cambio de contexto ocurre cuando el usuario cambia su atención del objeto en sí mismo a sus componentes, o viceversa. Por ejemplo, el usuario puede considerar un catálogo, u objetos de este catálogo, o partes de objetos específicos del catálogo. Aquí podemos ver que los distintos catálogos corresponden a un contexto, en tanto que los objetos en él pertenecen a otro contexto. Cada uno de estos contextos es llamado *escena*.

2.2.2.5.1 Funciones incrustadas para manejo de contexto y referencia a objetos.

Además de las operaciones estándar sobre pilas (meter y sacar: *push* y *pop*), podemos buscar los objetos por propiedad en una pila dada (pso, poo o ss). Las funciones incrustadas para el manejo de objetos y manejo de contexto se listan más abajo. Estas funciones se ejecutan en línea, esto es, se evalúan inmediatamente después de la aplicación de la regla que las generó en la cadena, y antes de aplicar otra regla.

Sintácticamente, las funciones incrustadas se denotan por el nombre de la función seguido de la lista de argumentos, la cual puede ser vacía.

objeto nombre_de_la_función (arg$_1$, arg$_2$, ... , arg$_n$),

donde *n* pertenece a los números naturales y puede ser cero. Una función puede regresar un objeto. La Tabla 2 muestra las funciones y procedimientos incrustados usados en nuestro formalismo

Tabla 2. Funciones y procedimientos incrustados.

Función	Descripción
`push (s, x)`	Mete el objeto *x* en la pila *s*
`objeto pop (s)`	Saca y regresa el objeto que se encuentra en la parte superior de la pila *s*
`objeto last (s)`	Regresa el objeto de la parte superior de la pila *s* sin sacarlo
`objeto last (s, p = v)`	Busca el primer objeto con el valor *v* de la propiedad *p*, comenzando desde la parte superior de la pila *s*. Si no se encuentra un objeto, regresa `NIL` (valor nulo).

Usando tres pilas podemos definir el procedimiento para buscar el objeto referenciado por el usuario como sigue:

```
P1:Buscar el objeto en pso
    si no lo encuentra: buscar objeto en poo,
        si no lo encuentra, ir a BuscaSS
BuscaSS:
    Buscar objeto en ss,
    si no lo encuentra: hasta que lo encuentre:
                ss.pop();
                repite BuscaSS.
```

2.2.2.5.2 Marcadores condicionales

Un marcador condicional es una función que se utiliza para tomar decisiones durante el procesamiento de las reglas. Su formato es:

```
if <condición> then <objeto1> else <objeto2> end.
```

Esta función en-línea regresa $objeto_1$ si la condición se cumple y $objeto_2$ si no. Por ejemplo, el procedimiento BuscaSS anterior puede implementarse como sigue:

```
A[C:ARTDET] B[C:SUST] ->
if pso.last (nombre = A.nombre) then pso.last (n = A.nombre) else
if poo.last (nombre = A.nombre) then poo.last (name = A.nombre) else
    if  ss.last (nombre = A.nombre) then ss.last (nombre = A.nombre)
           else ss.pop () A B end
```

como podemos ver en esta regla, la recursividad se expresa al copiar el lado izquierdo de la regla como el lado derecho, que en esta regla se expresa como `A B` en la última línea.

2.2.2.6 Ejemplos de procesamiento de peticiones

En esta sección presentamos un conjunto de reglas que puede procesar diversas peticiones en español. Estas peticiones (abreviadas utt, por *utterance*) están inspiradas en diálogos que fueron presentados en *El proyecto DIME* [146].

utt.1: *¿Me puedes mostrar el catálogo?*

utt.2: *¿Me puedes mostrar el catálogo de objetos formados por esferas?*

utt.3: *A ver, ¿cuál es la diferencia entre el tercero y el cuarto?*

utt.4: *¿Me puedes poner el tercero junto al toroide?*

2.2.2.6.1 Conjunto de reglas

Aquí presentamos el conjunto de reglas usado para el fragmento de peticiones presentadas anteriormente. La regla 1 sintetiza el procedimiento P1 para búsqueda de referencia de objetos.

```
1   A[C:ARTDET] B[C:SUST] ->
    if(sos.last(name = A.name),sos.last(name = A.name),
    if(oos.last(name = A.name),oos.last(name = A.name),
    if( ss.last(name = A.name), ss.last(name = A.name),
    ss.pop();A B))))
2   [C:ARTDET] cuarto{C:ADJ} -> 4{C:SUST}
3   [C:ARTDET] tercero{C:ADJ} -> 3{C:SUST}
4   B diferencia entre D y F -> diferencia D F
5   B[C:SUST,Q] -> if (ss.last(name = B.name),ss.last(name = B.name),
ss.last(prop = B.Q))
6   el -> el{C:ARTDET,S:SM}
7   nextto A[C:OBJ] -> getpos(A); oos.push(A)
8   cuarto -> cuarto{C:ADJ,S:SM}
9   diferencia A[C:OBJ] B[C:OBJ] -> diff(A,B);sos.push(A);sos.push(B);
10  el catálogo de objetos formados por esferas -> cat021
11  junto a -> nextto
12  me poder{C:V,T:PRES,S:2S} A[C:V,T:INF] -> A[T:IMP,S:2S,O:1S]
13  mostrar -> mostrar{C:V, T:INF}
14  mostrar{C:V,T:IMP,S:2s} A[C:OBJ] -> show(A); ss.push(A);
15  poner{C:V,T:IMP,S:1S} B[C:OBJ] D[C:POS] -> F=move(B,D);sos.push(F);
16  puedes -> poder{C:V,T:PRES,S:2S}
17  tercero -> tercero{C:ADJ,S:SM,Q:3}
18  catálogo -> catálogo{C:SUST,S:SM,Q:4}
19  poner -> poner{C:V,T:INF,S:1S}
20  al -> a el
```

2.2.2.6.2 Aplicación de las reglas

Ahora podemos procesar las peticiones presentadas al comienzo de esta sección. El número de la izquierda indica el número de la regla usada entre un paso y el siguiente.

utt1: ¿Me puedes mostrar el catálogo

```
16. me poder{C:V,T:PRES,S:2S} mostrar el catálogo
13. me poder{C:V,T:PRES,S:2S} mostrar{C:V,T:INF} el catálogo
12. mostrar{C:V,T:IMP,S:2S,O:1S} el catálogo
18. mostrar{C:V,T:IMP,S:2S,O:1S} el catálogo{c:SUST,S:SM}
 6. mostrar{C:V,T:IMP,S:2S,O:1S} el{C:ARTDET,S:SM} catálogo{c:SUST,S:SM}
 1. mostrar{C:V,T:IMP,S:2S,O:1S} objcat01
14. show(objcat01);ss.push(objcat01);
```

utt2: ¿Me puedes mostrar el catálogo de objetos formados por esferas?

16. me poder{C:V,T:PRES,S:2S} mostrar el catálogo de objetos formados por esferas
13. me poder{C:V,T:PRES,S:2S} mostrar{C:V,T:INF} el catálogo de objetos formados por esferas
12. mostrar{C:V,T:IMP,S:2S,O:1S} el catálogo de objetos formados por esferas
10. mostrar{C:V,T:IMP,S:2S,O:1S} cat021
14. show(cat021);ss.push(cat021);

utt3: A ver, ¿Cuál es la diferencia entre el tercero y el cuarto?

4. diferencia el tercero el cuarto
6. diferencia el{C:ARTDET,S:SM} tercero el cuarto
6. diferencia el{C:ARTDET,S:SM} tercero el{C:ARTDET,S:SM} cuarto
2. diferencia el{C:ARTDET,S:SM} tercero 4{C:SUST}
3. diferencia 3{C:SUST} 4{C:SUST}
5. diferencia objcat021_03 4{C:SUST}
5. diferencia objcat021_03 objcat021_04
9. diff(objcat021_03,objcat021_04); sos.push(objcat021_03); sos.push(objcat021_04);

utt4: ¿Me puedes poner el tercer junto al toroide?

16. me poder{C:V,T:PRES,S:2S} poner el tercero junto al toroide
19. me poder{C:V,T:PRES,S:2S} poner{C:V,T:INF,S:1S} el tercero junto al toroide
6. me poder{C:V,T:PRES,S:2S} poner{C:V,T:INF,S:1S} el{C:ARTDET,S:SM} tercero junto al toroide
20. me poder{C:V,T:PRES,S:2S} poner{C:V,T:INF,S:1S} el{C:ARTDET,S:SM} tercero junto a el toroide
6. me poder{C:V,T:PRES,S:2S} poner{C:V,T:INF,S:1S} el{C:ARTDET,S:SM} tercero junto a el{C:ARTDET,S:SM} toroide
11. me poder{C:V,T:PRES,S:2S} poner{C:V,T:INF,S:1S} el{C:ARTDET,S:SM} tercero nextto el{C:ARTDET,S:SM} toroide
17. me poder{C:V,T:PRES,S:2S} poner{C:V,T:INF,S:1S} el{C:ARTDET,S:SM} tercero{C:ADJ,S:SM} nextto el{C:ARTDET,S:SM} toroide
12. poner{C:V,T:IMP,S2s,O1S} el{C:ARTDET,S:SM} tercero{C:ADJ,S:SM} nextto el{C:ARTDET,S:SM} toroide
3. poner{C:V,T:IMP,S2s,O1S} 3{C:SUST} nextto el{C:ARTDET,S:SM} toroide
5. poner{C:V,T:IMP,S2s,O1S} objcat021_03 nextto el{C:ARTDET,S:SM} toroide
1. poner{C:V,T:IMP,S2s,O1S} objcat021_03 nextto obj002; oos.push(obj002);
7. poner{C:V,T:IMP,S2s,O1S} objcat021_03 getpos(obj002); oos.push(obj002);

```
15. move(objcat021_03,getpos(obj002));sos.push(obj003); oos.push(obj002);
```

En la última línea el objeto se copia cuando se mueve de un contexto al otro.

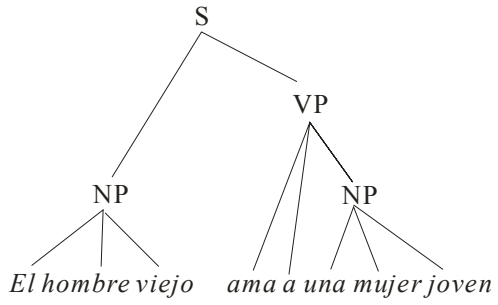
2.2.2.7 Resumen

En este capítulo, acerca del estudio del estado del arte de reglas de reescritura, presentamos un sistema que puede derivar una o más instrucciones específicas de computación a partir de una petición del usuario final. Los objetos referenciados dentro de esta petición se traducen en símbolos por una gramática de reglas de reescritura con modificación de propiedades, sustitución de comodines, funciones en línea, y el uso de objetos especiales para el manejo del contexto llamados Escenas. Las instrucciones se ejecutarán posteriormente por un sistema externo.

El sistema presentado en este capítulo puede usarse para tareas computacionales que cumplen con las siguientes condiciones: el usuario y la computadora comparten contextos comunes que son visualizados, el lenguaje usado es imperativo, y el dominio de aplicación es limitado. Hemos presentado ejemplos para la tarea de colocar objetos en un lienzo tridimensional. Sin embargo, el uso de reglas de reescritura no nos permite obtener una representación con roles semánticos como la que nos hemos planteado en el objetivo. Continuamos con nuestro estudio del estado del arte centrándonos ahora en otro formalismo: las gramáticas de dependencias.

2.2.3 Gramáticas de dependencias

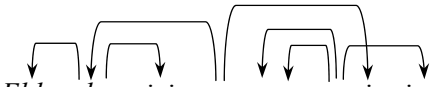
En un árbol de dependencias, los nodos del árbol son palabras simples, de tal forma que una dependencia se establece entre un par de palabras: una de las palabras es la principal o gobernante, y la otra es una subordinada (o dependiente) de la primera:.



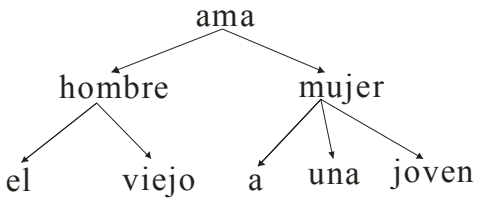
donde los nodos son partes de texto (constituyentes) y los arcos son relaciones de “consiste en”.

En el enfoque de dependencias, las palabras se consideran “dependientes” de, o que modifican otras palabras [124]. Una palabra modifica otra palabra (gobernante) en la oración si añade detalles a esta última, mientras que la combinación completa hereda las propiedades

sintácticas (y semánticas) del gobernante: *hombre viejo* es un tipo de *hombre* (y no un tipo de *viejo*); *hombre ama a mujer* es un tipo de (situación de) *amor* (y no, digamos, un tipo de *mujer*). Dicha dependencia se representa por una flecha del gobernador a la palabra gobernada:



o, en forma gráfica:



Todas excepto una palabra (*ama*) en la oración tienen una palabra gobernante. Una relación de dependencias entre un gobernante *G* y un dependiente *D* significa, en términos generales, que una combinación de palabras *G D* (o *D G*) es significativa y hereda las propiedades sintácticas y semánticas de *G* (y no *D*): *G D* es una *G* (y no es una *D*); se dice que *D* modifica a *G*. En nuestro ejemplo, la combinación *hombre viejo* es un tipo de *hombre* (y no un tipo de *viejo*); *hombre ama a mujer* es un tipo de (situación de) *amar* (y no, digamos, un tipo de *mujer*). A diferencia de un árbol estructurado por sintagmas, en el árbol de dependencias los arcos están (o pueden ser) etiquetados: *viejo* y *joven* son modificadores de atributos de *hombre* y *mujer*, respectivamente; *hombre* es el sujeto de *ama*.

La representación de dependencias simplifica grandemente ciertas tareas en comparación con el enfoque de constituyentes. Por ejemplo:

– En lexicografía, reunir estadísticas de combinabilidad sintáctica de palabras individuales (*leer un libro* y *clavar un clavo* vs. ?*leer un clavo* y ?*clavar un libro*) es trivial en la representación de dependencias: únicamente se cuentan las frecuencias de arcos que conectan a las instancias de dos palabras dadas en el corpus. Una de las numerosas aplicaciones de dichas estadísticas [14, 21,22] es la desambiguación sintáctica: se prefiere el árbol con pares de palabras frecuentemente [199, 80]. En cambio, en el enfoque de estructuras de sintagmas, esto es muy difícil, si no es que imposible.

– En recuperación de información y minería de texto, empatar una frase o una consulta compleja con las oraciones en el corpus es, de nuevo, casi trivial con un árbol de dependencias: una consulta *camiseta de mangas largas y rayas rojas* concuerda fácilmente con la descripción: *una playera de seda de buena calidad con rayas amplias rojas verticales y mangas azules largas* en

una base de datos de comercio electrónico, pero no con *una playera roja con rayas largas azules en las mangas*.

– En análisis semántico, transformar el árbol de dependencias en prácticamente cualquier representación semántica, como grafos conceptuales [176] o redes semánticas [124] es mucho más sencillo. De hecho, HPSG construye un tipo de árbol de dependencias para construir su representación de semántica de recursión mínima (MRS) [165].

Sin embargo, la mayoría de las herramientas existentes y recursos están orientados a la representación de estructura por sintagmas, para la cual, según se dice (discutiblemente) es más fácil construir un analizador.

A pesar de las aparentes diferencias, ambas representaciones comparten el grueso de información acerca de la estructura sintáctica, a tal grado que pueden combinarse [165]. Una puede ser automáticamente derivada de la otra, siempre que cierta información se añada que está presente en la segunda representación pero ausente en la primera. Básicamente, la estructura por sintagmas tiene más información acerca de una palabra dentro de una unidad estructura, en tanto que la estructura de dependencias tiene más información acerca de la herencia de propiedades sintácticas dentro de dicha unidad.

En este trabajo nos centraremos en esta representación por ser la que nos es más útil para nuestros propósitos, puesto que prácticamente todos los enfoques semánticos conocidos (como grafos conceptuales [176], Recursión de Semántica Mínima (MRS) [57], o redes semánticas [124]) se parecen a grandes rasgos a un conjunto de predicados, donde las palabras individuales representan predicados de sus argumentos (quienes a su vez pueden ser también predicados). Las estructuras resultantes están en una correspondencia mucho más directa con un árbol de dependencias que con un árbol de constituyentes de la oración en cuestión, de tal forma que la sintaxis de dependencias parece ser más apropiada para su traducción directa en estructuras semánticas. Específicamente, la estructura de dependencias hace que sea mucho más fácil hacer que coincidan (por ejemplo, en recuperación de información) paráfrasis del mismo significado (como la transformación de voz pasiva en activa y viceversa) o transformar de una estructura equivalente a otra.

Adicionalmente, la estructura producida por un analizador de dependencias puede obtenerse fácilmente de una manera más robusta que un analizador de constituyentes. Los enfoques conocidos del análisis de dependencia tratan mucho más fácilmente tanto con gramáticas incompletas y oraciones no gramaticales, que los enfoques estándar del análisis libre de contexto.

2.3 Acerca de la extracción automática de roles semánticos y preferencias de selección

2.3.1 Roles Semánticos: Un estudio de diversos enfoques

En esta sección comparamos los dos extremos en cuanto a la consideración de roles semánticos según diversos autores: desde el conjunto reducido de roles semánticos generalizados para todos los verbos [175], hasta la consideración de roles semánticos particulares a cada verbo, como se hace en FrameNet. Finalizamos con las preguntas que motivan esta investigación.

Los grafos conceptuales son estructuras en las que se representan entidades y relaciones entre ellas. Para representar una oración como un grafo conceptual, se considera que el verbo es una entidad que se relaciona con sus argumentos. El nombre de la relación entre el verbo y estos argumentos se conoce como **rol semántico**. Por ejemplo, según Sowa [175], para *el perro rompió la ventana,* el grafo conceptual es el siguiente:

[Perro]←(Fuente)←[Romper]→(Paciente)→[Ventana]

donde el rol semántico de *el perro* es fuente (el efector de la acción), y el rol semántico de *la ventana*, es paciente (quien recibe la acción).

Sin embargo, la construcción automática de grafos conceptuales a partir de oraciones enfrenta diversos problemas, principalmente el de determinar los roles semánticos de forma automática. A grandes rasgos, para representar una oración como un grafo conceptual es necesario seguir los siguientes pasos:

paso 1. identificar las categorías gramaticales de la oración

paso 2. segmentar adecuadamente las entidades. Por ejemplo, en la oración *voy a la casa de mi abuela*, aunque *la casa de mi abuela* son 5 palabras, forman una sola entidad. En cambio, en la oración *ponme el libro en el estante*, *el libro* es una entidad, en tanto que *el estante* es otra.

paso 3. clasificar las entidades semánticamente (*la casa de mi abuela* es un lugar)

paso 4. identificar las relaciones que tienen estas entidades dentro de la oración. (*la casa de mi abuela* tiene el rol de *meta* o *destino* en la oración *voy a casa de mi abuela*). Estas relaciones son los roles semánticos.

Tabla 3. Ejemplos de Ocurrencias para algunos verbos en español.

Combinación de categorías	Ocurrencias
ir a {actividad}	711
ir a {tiempo}	112
ir hasta {comida}	1
beber {sustancia}	242
beber de {sustancia}	106
beber con {comida}	1
amar a {agente_causal}	70
amar a {lugar}	12
amar a {sustancia}	2

{comida}: desayuno, banquete, cereal, frijoles, leche, etc.
{actividad}: abuso, educación, lección pesca, apuro, prueba
{tiempo}: oscuridad, historia, jueves, edad media, niñez
{sustancia}: alcohol, carbón, chocolate, leche, morfina
{nombre}: Juan, Pedro, América, China
{agente_causal}: abogado, capitán, director, intermediario, nieto
{lugar}: aeródromo, bosque, fosa, valle, traspatio, rancho

Figura 6. Ejemplos de palabras que pertenecen a las categorías mostradas en la Tabla 3.

Sin embargo no existe un conjunto estándar de roles semánticos. Existen diversas teorías, desde aquellas que generalizan los roles semánticos existentes (como los roles temáticos de Sowa [175]) hasta aquellas que consideran conjuntos particulares de roles para cada verbo, como FrameNet [6].

¿Cómo podemos elegir algún conjunto de roles semánticos de tal manera que estos puedan ser determinados automáticamente?

Nuestra hipótesis es que es posible deducir los roles semánticos a partir de estadísticas de las preferencias seleccionales de los verbos. Las preferencias seleccionales para verbos indican la tendencia de cierto tipo de argumentos a pertenecer al verbo. Por ejemplo, para el verbo *beber* es más probable un argumento del tipo {líquido} como su objeto directo. Las preferencias seleccionales, además, nos sirven para segmentar adecuadamente las entidades.

En la Tabla 3 se muestran ejemplos de ocurrencias de complementos para algunos verbos en español. A partir de esta tabla puede verse, por ejemplo, que el verbo *ir* es más utilizado con el complemento *a {actividad}*. Combinaciones menos utilizadas tienen casi cero ocurrencias, como por ejemplo *ir hasta {comida}*.

El verbo *amar* es a menudo utilizado con la preposición *a*. Esta información ha sido extraída con el método descrito en el Capítulo 4.

2.3.1.1 Relaciones semánticas

Las relaciones semánticas fueron introducidas en la gramática generativa a mediados de los 1960's e inicios de los 1970s [73, 101, 90] como una manera de clasificar los argumentos del lenguaje natural en un conjunto cerrado de tipos de participación que se pensaba tenían un status especial en la gramática. A continuación se presenta una lista de los roles más usados y las propiedades usualmente asociadas con ellos:

agente: un participante que según el verbo especifica realización o causa de algo, posiblemente de manera intencional. Ejemplos: los sujetos de *matar, comer, golpear, patear, ver*.

paciente: un participante al cual le sucede algo, caracterizado por el verbo y afectado por lo que le sucede. Ejemplos: los objetos directos de *matar, comer,* pero no aquellos de *ver, oír, amar*.

experimentador: un participante que se caracteriza como consciente de algo. Ejemplos: sujeto de *amar*.

tema: un participante que se caracteriza por cambiar su posición o condición, o por estar en un estado o posición. Ejemplos: el objeto directo de *dar*, el sujeto de *caminar, morir*.

locación: el rol temático asociado con el sintagma nominal que expresa la ubicación en una oración con un verbo de ubicación. Ejemplos: sujetos de *retener, saber*, complementos circunstanciales de lugar.

fuente: objeto del cual procede el movimiento. Ejemplos: sujetos de *comprar, prometer*, objetos de *privar, librar, curar*.

meta: objeto al cual se dirige el movimiento. Ejemplos: sujeto de *recibir, comprar*, objetos indirectos de *contar, dar*.

En la teoría lingüística, los roles temáticos han sido tradicionalmente considerados como determinantes para expresar generalizaciones acerca de la realización sintáctica de los argumentos del predicado.

El aspecto teórico de los roles semánticos en la teoría lingüística es aún un problema no resuelto. Por ejemplo, existen dudas considerables acerca de si los roles semánticos deberían ser considerados como entidades sintácticas, léxicas o semántico-conceptuales. Otro problema, conectado con el anterior, es si los roles semánticos deberían ser considerados como una parte primitiva del conocimiento lingüístico [73,72,70,193,48,11], o como una noción derivativa de cierto aspecto específico del mapeo forma-significado [101,102,152]. Sin embargo la idea más generalizada es que los roles semánticos son elementos semántico-conceptuales. La mayoría de las

caracterizaciones de los roles semánticos han sido expresadas en términos de propiedades semánticas primitivas de los predicados. Por ejemplo, Jackendoff [101] sugirió que las relaciones temáticas deberían estar definidas en términos de tres subfunciones semánticas *causa, cambio* y *ser* que constituyen algunos bloques de construcción primitivos para las representaciones conceptuales. De acuerdo con este tratamiento, la representación léxico-conceptual de un verbo transitivo como *abrir* sería como se muestra a continuación, donde el SN_1 se interpreta como un agente y el SN_2 como tema.

$$\text{CAUSAR}\left(\text{SN}_1, \left[\begin{matrix}\text{CAMBIO}\\ \textit{físico}\end{matrix}\right]\left(\text{SN}_2, \text{NO ABIERTO}, \text{ABIERTO}\right)\right)$$

2.3.1.2 Grafos Conceptuales[6]

En los grafos conceptuales los roles semánticos se representan por relaciones conceptuales que unen el concepto de un verbo a los conceptos de los participantes en el sentido expresado por el verbo. En los sistemas de marcos (*frame systems*), se representan por *slots* en el marco del verbo correspondiente. Estas notaciones son formas equivalentes de representar los vínculos entre un proceso y sus participantes.

En la ontología de Representación del Conocimiento (*Knowledge Representation, KR*), los roles temáticos se clasifican como subtipos de participante, que se subdivide más por dos pares de distinciones: *determinante* o *imanente* y *fuente* o *producto*. Esta subdivisión genera los cuatro tipos básicos de participantes mostrados en la Figura 7

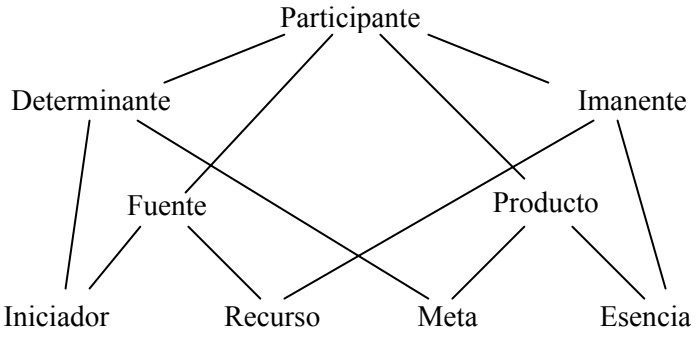


Figura 7: Representación gráfica de los subtipos de participante

En la Figura 7, el tipo participante está subdividido por dos pares de distinciones para generar cuatro subtipos mostrados en el nivel inferior del grafo. Cada participante es una entidad que juega

[6] Sección basada en la sección Roles Temáticos de [Sowa, 2000]

un papel en un proceso. La subdivisión mostrada en la Figura 7 distingue a los participantes según el tipo de papel que juegan. En los lenguajes naturales, estas distinciones se expresan por marcadores gramaticales como preposiciones y marcadores de caso, que vinculan los verbos que expresan los procesos a los sustantivos que expresan los participantes. En lógica estas distinciones se expresan por relaciones o predicados que unen a los símbolos que identifican los procesos de los símbolos que identifican los participantes.

- Un participante *determinante* determina la dirección del proceso, ya sea desde el principio como el iniciador, o al final, como el objetivo

- Un participante *imanente* está presente durante todo el proceso, pero no controla activamente lo que sucede

- Una *fuente* debe estar presente al inicio del proceso, pero no necesita participar a través del proceso

- Un *producto* debe estar presente al final del proceso, pero no necesita participar a través de él.

Por ejemplo, considere la oración *Susana envió el regalo a Roberto por Estafeta*. El regalo y Estafeta son participantes imanentes, puesto que el regalo (esencia) y Estafeta (recurso) están presentes desde el inicio hasta el final. Susana y Roberto, sin embargo, son participantes determinantes, puesto que determinan el curso del proceso desde el iniciador (Susana) hasta el objetivo (Roberto). A diferencia de los participantes imanentes, los participantes determinantes están involucrados principalmente en los extremos. Si Susana hubiera escrito la dirección erróneamente, el posible receptor, Roberto, podría no involucrarse para nada.

Después de analizar y resumir diversos sistemas de relaciones de casos o roles temáticos, Somers [172] los organizó en una red con cuatro tipos de participantes en el nivel superior y seis categorías de verbos al lado. En las 24 cajas de la matriz, Somers tenía algunas cajas con nombres de roles duplicados y algunas cajas con dos roles que se distinguían por otras propiedades: ±animado, ±físico, ±dinámico, o ±volicional. Al usar la clasificación de Somers, Dick [69] aplicó los roles a los grafos conceptuales como representación del conocimiento para argumentos legales.

Estimulado por el trabajo de Moravcsik [135] y Pustejovsky [151], Sowa [173] relacionó las cuatro columnas de la matriz de Somers-Dick a las cuatro causas de Aristóteles, o *aitia*, como se describen en *Metafísica*:

- *Iniciador* corresponde a la causa eficiente de Aristóteles, "dondequiera que un cambio o estado se inicia"

- *Recurso* corresponde a la causa material, que es "la materia del sustrato (*hypokeimenon*)"

- *Meta* corresponde a la causa final, que es "el propósito o el beneficio; pues esta es la meta (*telos*) de cualquier generación de movimiento"

- *Esencia* corresponde a la causa formal, que es "la esencia (*ousia*) o lo que es (*to ti ên einai*)"

Los cuatro términos, *iniciador*, *recurso*, *meta* y *esencia* describen mejor los participantes de una acción que las cuatro causas de Aristóteles. En la Tabla 4 hay una versión de la matriz de Somers-Dick de roles temáticos con la terminología adaptada a la ontología presentada en el libro de Sowa [175].

Tabla 4. Roles temáticos como subtipos de los cuatro tipos de participantes

	Iniciador	**Recurso**	**Meta**	**Esencia**
Acción	Agente, Efector	Instrumento	Resultado, Receptor	Paciente, Tema
Proceso	Agente, Origen	Materia	Resultado, Receptor	Paciente, Tema
Transferencia	Agente, Origen	Instrumento, Medio	Experimentador, Receptor	Tema
Espacial	Origen	Trayectoria	Destino	Localización
Temporal	Inicio	Duración	Término	Punto en el tiempo
Ambiente	Origen	Instrumento, Materia	Resultado	Tema

Las opciones y los duplicados en las cajas de la tabla indican que pueden hacerse distinciones más profundas. La opción de Agente o Efector como iniciador de una acción está determinada por la distinción de un iniciador voluntario (Agente) o un iniciador involuntario (Efector). La duplicación del rol de Agente para acciones, procesos y transferencias, inicia interacciones implícitas entre los

tipos de verbos y los tipos de participantes. La Tabla 4 es un inicio importante, pero debería ser extendida con análisis posterior siguiendo las líneas de la clasificación de los verbos de Beth Levin.

En caso de ambigüedad, la jerarquía mostrada en la Figura 7 permite un tipo más especializado de participante en la parte inferior para ser generalizado a cualquier supertipo sobre él en la gráfica. En la oración *Tomás horneó un pastel*, el pastel puede ser un resultado (producto determinante) que está siendo creado, o un paciente (producto imanente) que está siendo calentado. Estas dos interpretaciones se expresarían por dos grafos conceptuales distintos:

```
[Persona: Tomás]←(Agnt)←[Hornear]→(Rslt)→[Pastel: #].
[Persona: Tomás]←(Agnt)←[Hornear]→(Pcnt)→[Pastel: #].
```

Pero de acuerdo con la jerarquía de participantes, Resultado < Meta < Producto, y Paciente < Esencia < Producto. Puesto que Producto es un supertipo común, la interpretación inicial podría haber sido etiquetada Prod. La representación resultante sería un solo grafo conceptual que expresara exactamente la misma información que la oración original sin hacer ninguna suposición sobre el estado imanente o determinante del pastel.

```
[Persona: Tomás]←(Agnt)←[Hornear]→(Prod)→[Pastel: #].
```

En la oración *El perro rompió la ventana*, el perro podría ser un agente que la rompió deliberadamente, un efector que la rompió accidentalmente, o un instrumento que fue aventado a través de la ventana por el agente real. Cada interpretación sería expresada por un grafo conceptual diferente.

```
[Perro: #]←(Agnt)←[Romper]→(Pcnt)→[Ventana: #].
[Perro: #]←(Efct)←[Romper]→(Pcnt)→[Ventana: #].
[Perro: #]←(Inst)←[Romper]→(Pcnt)→[Ventana: #].
```

Pero Agente < Iniciador < Fuente, Efector < Iniciador < Fuente e Instrumento < Recurso < Fuente. Puesto que todos los tipos de participantes son subtipos especializados de Fuente, un solo Grafo Conceptual con la relación Fte expresaría la información equivalente en la oración original.

```
[Perro: #]←(Fte)←[Romper]→(Pcnt)→[Ventana: #].
```

Cuando se disponga de más información del rol del perro, el tipo de relación Fte puede especializarse a uno de los tres subtipos usados en los grafos anteriores.

La siguiente lista da una breve descripción y un ejemplo para cada uno de los roles temáticos que aparecen en la Tabla 4. El primer término en cada entrada es el nombre del rol, p. ej. Agente.

Siguiendo el símbolo < se encuentra el supertipo, p. ej. Iniciador. Después aparece una abreviación como Agnt seguido de restricciones de categoría sobre el tipo de concepto del verbo (Act) y el tipo de concepto del participante (Animado). Cada relación se define en términos del rol correspondiente y la relación diádica correspondiente Tiene. Agnt, por ejemplo, se define como TieneAgnt, Benf es TieneBeneficiario, y Term es TieneTérmino.

Agente < Iniciador; Agnt(Act,Animado).

Una entidad animada activa que voluntariamente inicia una acción.

Ejemplo: *Eva mordió un manzana.*

`[Persona: Eva]←(Agnt)←[Morder]→(Ptnt)→[Manzana].`

Beneficiario < Receptor; Benf (Act, Animado)

Un receptor que obtiene un beneficio a partir del término exitoso del evento.

Ejemplo: *Los diamantes le fueron dados a Rubí*.

`Diamante: {*}]←(Tema)←[Dar]→(Benf)→[Persona: Rubí].`

Término < Meta; Term(ProcesoTemporal, Físico).

Una meta de un proceso temporal

Ejemplo: *María esperó hasta el amanecer*.

`[Persona: María]←(Tema)←[Esperar]→(Term)→[Amanecer].`

Destino < Meta; Dest (ProcesoEspacial, Físico).

Una meta de un proceso espacial

Ejemplo: *Roberto fue a Cuernavaca.*.

`[Persona: Roberto]←(Agnt)←[Ir]→(Dest)→[Ciudad: Cuernavaca].`

Duración < Recurso; Dur (Estado, Intervalo)

Un recurso de un proceso temporal.

Ejemplo: El camión fue reparado por 5 horas.

`[Camión: #]←(Tema)←[Reparar]→(Dur)→[Intervalo: @5hrs].`

Efector < Iniciador, Efct(Entidad, Entidad).

Una fuente activa determinante, ya sea animada o inanimada, que inicia una acción pero sin intención voluntaria.

Ejemplo: *El árbol dio hojas nuevas*.

`[Arbol: #]←(Efct)←[Dar]→(Rslt)→[Hoja: {*}]→(Atr)→[Nuevo].`

Experimentador < Meta; Expr(Estado, Animado).

Una meta activa animada de una experiencia.

Ejemplo: *Marlín ve al pez*.

```
[Pez: Marlín]←(Expr)←[Ver]→(Tema)→[Pez: #].
```

Instrumento < Recurso; Inst(Act, Entidad).

Un recurso que no es cambiado por un evento.

Ejemplo: *La llave abrió la puerta*

```
[Llave: #]←(Inst)←[Abrir]→(Tema)→[Puerta: #].
```

Locación < Esencia; Loc (Físico, Físico).

Un participante esencial de un nexo espacial.

Ejemplo: *Los vehículos llegan a la estación*.

```
[Vehículo: {*}]←(Tema)←[Llegar]→(Loc)→[Estación].
```

Materia < Recurso; Matr(Act, Sustancia).

Un recurso que es cambiado por el evento.

Ejemplo: *El arma fue tallada en madera*.

```
[Arma]←(Rslt)←[Tallar]→(Matr)→[Madera].
```

Medio < Recurso; Med (Transferencia, Físico)

Un recurso físico para transmitir información, como el sonido de la voz o las señales electromagnéticas que transmiten datos.

Ejemplo: *Bill le dijo a Boris por Teléfono*

```
[Persona: Bill]←(Agnt)←[Decir]-
   (Expr)→[Persona: Boris]
   (Med)→[Teléfono].
```

Origen < Iniciador; Orgn(Proceso, Físico).

Una fuente pasiva determinante de un nexo espacial o ambiental

Ejemplo: *El capítulo comienza en la página 20*.

```
[Capítulo: #]←(Tema)←[Comenzar]→(Orgn)→[Página: 20].
```

Trayectoria < Recurso; Tray(Proceso, Lugar)

Un recurso de un nexo espacial.

Ejemplo: *La pizza fue enviada vía Tlalnepantla y Naucalpan*.

```
[Pizza: #]←(Tema)←[Enviar]
                  →(Tray)→[Ciudad: {Tlalnepantla, Naucalpan}].
```

Paciente < Esencia; Pcnt (Proceso, Físico).

Un participante esencial que padece algún cambio estructural como resultado del evento.

Ejemplo: *Silvestre se tragó a Piolín.*

```
[Gato: Silvestre]←(Agnt)←[Tragar]→(Pcnt)→[Canario: Piolín].
```

PuntoenelTiempo < Esencia; PTie(Físico, Tiempo).

Un participante esencial de un nexo terminal.

Ejemplo: *A las 5:25 PM partió Erin.*

```
[Tiempo: 5:25pm]←(PTie)
                 ←[Situación: [Persona: Erin]←(Agnt)←[Partir]].
```

Receptor < Meta; Rcpt(Act, Animate).

Una meta animada de un acto

Ejemplo: *Susana envió el regalo a Roberto*.

```
[Persona: Susana]←(Agnt)←[Enviar]-
     (Tema)→[Regalo: #]
     (Rcpt)→[Persona: Roberto].
```

Resultado < Meta; Rslt (Proceso, Entidad)

Una meta inanimada de un acto

Ejemplo: *Eric construyó una casa*

```
[Persona: Eric]←(Agnt)←[Construir]→(Rslt)→[Casa].
```

Inicio < Iniciador; Ini(Entidad, Tiempo)

Una fuente determinante de un nexo temporal

Ejemplo: Juan esperó desde la mañana hasta las tres

```
[Persona: Juan]←(Tema)←[Esperar]-
     (Ini)→[Mañana]
     (Term)→[Tiempo: 3pm].
```

Tema < Esencia; Tema (Situación, Entidad)

Un participante esencial que puede ser movido, mencionado (dicho) o experimentado, pero no cambia estructuralmente.

Ejemplo: *A Torcuato le gusta la cerveza*.

```
[Persona: Torcuato]←(Expr)←[Gustar]→(Tema)→[Cerveza: #].
```

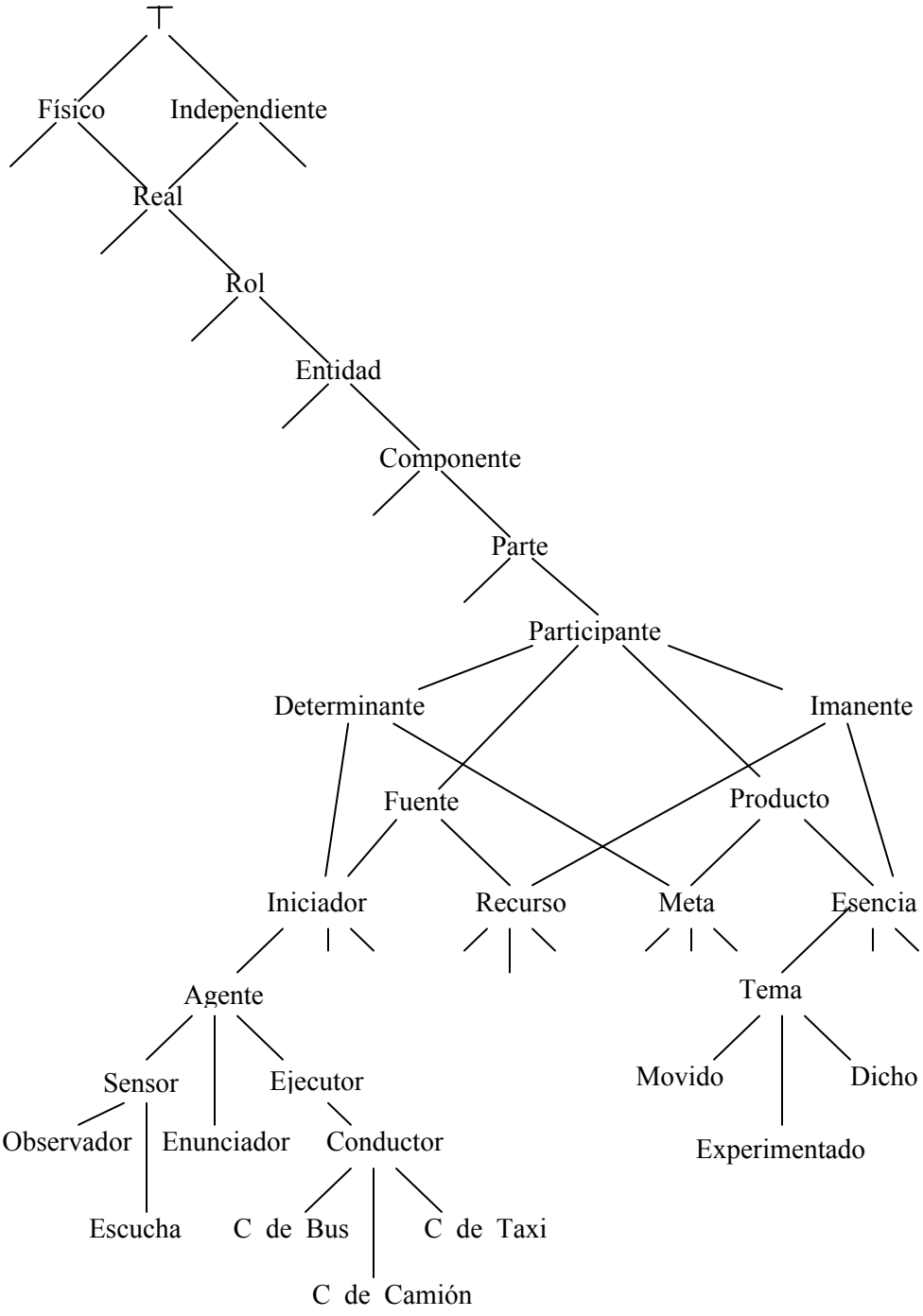


Figura 8. Ubicación de los roles temáticos en la ontología.

Como subtipos de Participante, los roles temáticos ocupan un nivel intermedio en la ontología. La Figura 8 muestra una trayectoria a través de la jerarquía desde los niveles superiores de la ontología a los subtipos de Participante mostrados en la Figura 7. Cada uno de los roles temáticos de la Tabla 4 podrían arreglarse dentro de los cuatro subtipos de Participante: Iniciador, Recurso, Meta y Esencia. Las líneas incompletas en la Figura 8 sugieren otras ramas de la ontología que se han omitido con el objeto de evitar el amontonamiento en el diagrama.

En la parte inferior de la Figura 8 hay ramas de muestra de la ontología bajo Agente y Tema. Agente, por ejemplo, tiene como subtipo Conductor, que tiene subtipos más específicos como Conductor_de_Autobús, Conductor_de_Camión y Conductor_de_Taxi. En principio, cualquiera de los roles temáticos podría subdividirse más para mostrar distinciones que podrían ser significativas en algún dominio de interés. Otros roles temáticos podrían también subdividirse más para representar a los participantes de tipos específicos de conceptos: Hablante < Agente; Sensor < Agente; Destinatario < Receptor; Experimentado < Tema; Movido < Tema; Dicho < Tema.

Aunque los roles temáticos representan una clase importante de categorías ontológicas, su supertipo común Participante está varios niveles más allá de la categoría general Rol. Por tanto, Rol incluiría muchos tipos que no están directamente asociados con verbos. Por ejemplo, el rol Conductor de la Figura 8 representa solamente una persona que está conduciendo activamente un vehículo; este rol sería incompatible con el rol Peatón. La categoría Conductor_con_Licencia, sin embargo, incluye a personas que están autorizadas legalmente para conducir, estén o no haciéndolo. En Nueva York, los conductores con licencia probablemente pasan más tiempo como peatones que como conductores. Como otro ejemplo, una persona podría tener un período continuo de empleo como chofer, pero no ser un conductor activo continuamente. Por lo tanto, el tipo Chofer sería un subtipo de Empleado y Conductor_con_Licencia, pero no un subtipo de Conductor.

2.3.1.3 FrameNet

FrameNet es un proyecto lexicográfico computacional de gran escala, independiente del dominio, organizado según los principios que motivan la semántica léxica, a saber: que pueden hallarse relaciones sistemáticas entre los componentes de significado de palabras, principalmente los roles semánticos asociados con eventos y sus propiedades combinatorias en la sintaxis. Este principio ha sido instanciado a diversos niveles de granularidad en diversas tradiciones de investigación lingüística; los investigadores de FrameNet trabajan con un nivel determinado de granularidad llamado *marco (frame)*. Ejemplos de marcos son: MOVIMIENTO DIRECCIONAL, CONVERSACIÓN, JUICIO y TRANSPORTACIÓN. Los marcos consisten de múltiples unidades léxicas, que son elementos que corresponden al sentido de una palabra. Asociado a cada marco existe un conjunto de roles

semánticos. Ejemplos para el marco de MOVIMIENTO direccional incluyen el objeto en movimiento, llamado TEMA, el destino final: la META; la FUENTE y la TRAYECTORIA.

Además de los marcos y las definiciones de roles, FrameNet ha producido un gran número de oraciones anotadas con roles. Estas oraciones están tomadas principalmente del Corpus Nacional Británico. Finalmente, el corpus contiene también información de categorías sintácticas para cada rol. A continuación mostramos algunos ejemplos en inglés de FrameNet. El marco aparece en llaves al inicio, el predicador en negrita, y cada constituyente relevante etiquetado con su rol y el tipo de sintagma. Note que el último ejemplo tiene un rol de DRIVER que tiene una instancia nula.

{MOTION DIRECTIONAL} Mortars lob heavy shells high into the sky so that
[NP THEME they] **drop** [PP PATHdown] [PP GOALon the target] [PP SOURCE from the sky].

{ARRIVING} He heard the sound of liquid slurping in a metal container as
[NP THEMEFarrell] **approached** [NP GOALhim] [PP SOURCE from behind]

{TRANSPORTATION} [NULL DRIVER] [NP CARGO The ore] was **boated** [PP GOAL down the river].

3 Estructura general del sistema propuesto (DILUCT)

3.1 Introducción

Después de explorar diversos enfoques y distintas estructuras representacionales descritos en los dos capítulos anteriores, encontramos que el formalismo que nos permite realizar nuestro objetivo es la representación con árboles de dependencias. En este capítulo describimos la construcción y diseño del sistema que nos permitirá obtener una estructura de dependencias con roles semánticos a partir de una oración en español. Hemos llamado DILUCT a este sistema, por ser las letras más frecuentes en el español, después de e, a, o, s y n[7].

Aunque algunas reglas específicas, así como los recursos léxicos y las herramientas de preprocesamiento utilizadas son específicas para el español, el entorno en general es independiente del lenguaje. Una interfaz en línea y el código fuente del sistema están disponibles en Internet[8]

3.2 Enfoque de dependencias

El enfoque de dependencias en relación a la sintaxis fue introducido inicialmente por Tesnière [182] y fue desarrollado posteriormente por Mel'čuk [124], quien lo utilizó extensamente en su Teoría Texto ⇔ Significado [123, 178], en conexión con la representación semántica, así como con diversas propiedades léxicas de las palabras, incluyendo funciones léxicas [122, 20].

Uno de los primeros intentos serios de los que tenemos conocimiento, para construir un analizador de dependencias fue el módulo sintáctico del sistema de traducción inglés-ruso ETAP [4]. El algoritmo de análisis consiste en dos pasos principales:

1. Se identifican todos los pares individuales de palabras con una relación potencialmente plausible de dependencia.
2. Los así llamados filtros remueven los enlaces incompatibles con otros enlaces identificados.
3. De los enlaces potenciales que quedan, se elige un subconjunto que forma un árbol (a saber, un árbol proyectivo, excepto por ciertas situaciones específicas).

---

[7] Siguiendo la idea de Winograd de llamar a su sistema SHRDLU.

[8] http://likufanele.com/diluct.

En ETAP, la gramática (un compendio de situaciones donde es plausible una relación de dependencias) está descrita en un lenguaje de especificación especialmente desarrollado que describe los patrones a ser buscados en la oración, y las acciones para construir el árbol que serán realizadas cuando dicho patrón se encuentre. Tanto los patrones como las acciones se desarrollan en una forma semi-procedural, usando numerosas funciones incorporadas (algunas de las cuales son específicas al lenguaje) usadas por el intérprete de la gramática. Una regla de patrón-acción en promedio consiste de 10 a 20 líneas de código denso. Hasta donde sabemos, no se utiliza información estadística en el analizador de ETAP.

Nuestro trabajo está inspirado por este enfoque; sin embargo, tomamos las siguientes decisiones de diseño distintas a aquellas de ETAP. Primero, nuestro analizador está diseñado para ser mucho más simple, incluso a pesar del costo inevitable de pérdida de exactitud. Segundo, no nos basamos en recursos léxicos complejos y detallados. Tercero, nosotros sí utilizamos estadísticas de co-ocurrencia de palabras, las cuales creemos que compensan la falta de completitud de la gramática.

De hecho, Yuret [199] ha mostrado que las estadísticas de co-ocurrencia (más precisamente, una medida similar a la que él le llama *atracción léxica*), puede por sí misma proveer suficiente información para el análisis exacto de dependencia, sin utilizar para nada una gramática hecha a mano. En su algoritmo, de todos los árboles proyectivos, se elige el que provee el valor más alto de atracción léxica de todos los pares conectados de palabras. Sin embargo, su enfoque se basa en cantidades enormes de datos de entrenamiento (aunque el entrenamiento es no supervisado). En adición, sólo puede construir árboles proyectivos (un árbol es llamado proyectivo si no tiene arcos que crucen según la representación gráfica mostrada en la sección 2.2.3).

Creemos que un enfoque combinado usando tanto una gramática simple hecha a mano y estadísticas de co-ocurrencia aprendidas de manera no supervisada de un corpus más pequeño, provee un compromiso razonable entre exactitud y factibilidad práctica.

Por otra parte, la corriente dominante de la investigación actual en análisis de dependencias está orientada a gramáticas formales [64]. De hecho, la gramática HPSG [148] fue quizá uno de los principales intentos exitosos para lograr una estructura de dependencias (necesaria tanto para usar la información léxica en el analizador en sí mismo, como para construir la representación semántica) usando una combinación de maquinaria de constituyentes y dependencias. Como hemos mencionado, una desventaja de los enfoques no basados en heurísticas (como aquellos basados en gramáticas formales) es su baja robustez.

El analizador usa un conjunto ordenado de reglas heurísticas simples para determinar iterativamente las relaciones entre palabras a las cuales no se les ha asignado aún un gobernante. En

el caso de ambigüedades de ciertos tipos, se utilizan estadísticas de co-ocurrencia de palabras reunidas previamente de una manera no supervisada a partir de un corpus grande, o a partir de la Web (a través de un buscador como Google). De esta manera se selecciona la variante más probable. No se utiliza un treebank preparado manualmente para el entrenamiento.

Siguiendo el enfoque estándar, pre-procesamos primero el texto de entrada (separación de partículas (tokenización), división de oraciones, etiquetado y lematizado). Posteriormente aplicamos reglas heurísticas para la obtención de una estructura que posiblemente contiene ambigüedades. Estas ambigüedades serán resultas en los módulos de Preferencias de Selección y Desambiguación de unión de sintagma preposicional descritos en los capítulos 4 y 5, respectivamente.

3.3 Preprocesamiento

3.3.1 Tokenización y división de oraciones.

El texto se tokeniza en palabras y símbolos de puntuación y se divide en oraciones. Actualmente no distinguimos signos de puntuación, de tal forma que cada signo de puntuación se sustituye con una coma, a excepción del punto final entre cada frase. En el futuro puede considerarse un tratamiento distinto para distintos signos de puntuación.

Se separan los compuestos de preposición y artículo: *del* → *de el*, *al* → *a el*.

Las preposiciones compuestas representadas en la escritura como diversas palabras, se unen como una sola palabra, como *con la intención de*, *a lo largo de*, etc. De manera similar se tratan unas cuantas frases adverbiales como *a pesar de*, *de otra manera*, y diversas frases pronominales como *sí mismo*. La lista de dichas combinaciones es pequeña (actualmente contiene 62 elementos) y es cerrada.

Actualmente no realizamos reconocimiento de entidades nombradas, si bien lo hemos considerado para el futuro.

3.3.2 Etiquetado

El texto es etiquetado con categorías gramaticales obteniendo las posibles categorías gramaticales a partir del analizador morfológico AGME [79]. Para elegir sólo una de las posibles categorías gramaticales, usamos el etiquetador estadístico TnT [24] entrenado con el corpus en español CLiC-

TALP[9]. Este etiquetador tiene un desempeño de más de 94% [134]. Además corregimos algunos errores frecuentes del etiquetador. Por ejemplo (Det es determinante, Adj es adjetivo, V es verbo, N es sustantivo y Prep es preposición):

Regla	Ejemplo
Det Adj V → Det N V	*el inglés vino*
Det Adj Prep → Det N Prep	*el inglés con*

3.3.3 Lematización

Usamos un analizador morfológico basado en diccionario [83] [10]. En caso de ambigüedades, se selecciona la variante de la categoría gramatical reportada por el etiquetador, con las siguientes excepciones:

El etiquetador predice	El analizador encuentra:	Ejemplo:
Adjetivo	Pasado participio	*dado*
Adverbio	Presente participio	*dando*
Sustantivo	Infinitivo	*dar*

Si el analizador no da una opción en la primer columna, pero se da una en la segunda columna, esta última se acepta.

Si un sustantivo, adjetivo, o participio esperado no se reconoce por el analizador, tratamos de remover el sufijo. Por ejemplo: *flaquito* → *flaco*. Para esto, tratamos de remover el sufijo sospechoso y verificar si la palabra es reconocida por el analizador morfológico. Ejemplos de reglas de eliminación de sufijo son:

Regla	Ejemplo
-cita → *-za*	*tacita* → *taza*
-quilla → *-ca*	*chiquilla* → *chica*

3.4 Reglas

Las reglas de análisis se aplican al texto lematizado. Siguiendo un enfoque similar al de [4 y 31], representamos una regla como un subgrafo, como: N ← V. La aplicación de una regla consiste en los siguientes pasos:

1. Se busca en la oración una subcadena que empate con la secuencia de palabras en la regla.

[9] http://clic.fil.ub.es.

[10] http://Gelbukh.com/agme.

2. Se establecen relaciones sintácticas entre las palabras que empataron de acuerdo a las relaciones que se especifican en la regla.

3. Todas las palabras a las cuales se les ha asignado un gobernante por la regla, se retiran de la oración en el sentido de que ya no participan en comparaciones posteriores en el paso 1.

Por ejemplo, para la oración *Un perro grande ladra*:

Oración	Regla
Un(Det) *perro*(N) *grande*(Adj) *ladra* (V)	Det ← N
perro(N) *grande*(Adj) *ladra* (V) ↓ *Un*(Det)	N → Adj
perro(N) *ladra* (V) ↙ ↘ *Un*(Det) *grande*(Adj)	N ← V
ladra (V) ↓ *perro*(N) ↙ ↘ *Un*(Det) *grande*(Adj)	Listo

Como puede verse del ejemplo, el orden de la aplicación de las reglas es importante. Las reglas son ordenadas; a cada iteración del algoritmo, la primer regla aplicable se aplica, y entonces el algoritmo repite la búsqueda de una regla aplicable a partir de la primera. El procesamiento se detiene cuando no se puede aplicar ninguna regla.

Note que una de las consecuencias de dicho algoritmo es su tratamiento natural de modificadores repetidos. Por ejemplo, en las frases *el otro día* o *libro nuevo interesante*, los dos determinantes (dos adjetivos, respectivamente) serán conectados como modificadores al sustantivo por la misma regla Det ← N (N → Adj, respectivamente) en dos iteraciones sucesivas del algoritmo.

Nuestras reglas no se encuentran aún completamente formalizadas (es por ello que le llamamos a nuestro enfoque "semi-heurístico", de tal forma que en las siguientes reglas añadiremos comentarios a algunas de ellas. Actualmente nuestra gramática incluye las siguientes reglas[11]:

Regla	**Ejemplo**
Sistema auxiliar de verbos y cadenas de verbos	
estar | *andar* ← Ger	*estar comiendo*
haber | *ser* ← Part	*haber comido*
haber ← *estado* ← Ger	*haber estado comiendo*

[11] La barra | quiere decir variantes: *estar* | *andar* ← Ger son dos reglas: *estar* ← Ger y *andar* ← Ger.

ir_{pres} *a* ← Inf	*ir a comer*
ir_{pres} ← Ger ← Inf	*ir queriendo comer*
V → *que* → Inf	*tener que comer*
V → V	*querer comer*
Construcciones estándar	
Adv ← Adj	*muy alegre*
Det ← N	*un hombre*
N → Adj	*hombre alto*
Adj ← N	*gran hombre*
V → Adv	*venir tarde*
Adv ← V	*perfectamente entender*
Conjunciones (vea la explicación más abajo)	
N Conj N V(pl) ⇒ [N N] V(pl)	*Juan y María hablan*
X Conj X ⇒ [X X] (X significa cualquiera)	(*libro*) *nuevo e interesante*
Otras reglas	
N → *que* V	*hombre que habla*
que → V	*que habla*
N X *que* (X significa cualquiera)	*hombre tal que*; *hombre , que*
Det ← Pron	*otro yo*
V → Adj	*sentir triste*
N , Adj	*hombre , alto*
N , N	*hombre , mujer*
N → Prep → V	*obligación de hablar*
V , V	*comer , dormir*
V Det ← V	*aborrecer el hacer*

Las conjunciones coordinantes siempre han sido un incordio para los formalismos de dependencias y un argumento a favor de los enfoques de constituyentes. Siguiendo la idea de Gladki [184], representamos las palabras coordinadas de una forma similar a los constituyentes, uniéndolos en una cuasi-palabra compuesta. En el "árbol" resultante efectivamente duplicamos (o multiplicamos) cada arco que entra o que sale de dicho nodo especial. Por ejemplo, en el fragmento [*Juan María*] ← *hablar* (*Juan y María hablan*) se interpreta como representar dos relaciones: *Juan* ← *habla* y *María* ← *habla*. El fragmento *merry* ← [*John Mary*] ← *marry* (*Merry John and Mary marry*) genera los pares de dependencias: *merry* ← *John* ← *marry* y *merry* ← *Mary* ← *marry*.

En correspondencia con esto, nuestras reglas para manejar las conjunciones son reglas de reescritura, no reglas de construcción de árboles. La primer regla forma un compuesto cuasi-palabra a partir de dos sustantivos coordinados si preceden a un verbo en plural. Esta regla elimina la conjunción, y la conjunción no participa en la estructura de árbol. Básicamente lo que hace la regla es asegurarse de que todos los verbos que tienen ese sujeto compuesto están en plural, para evitar la

interpretación de *Juan ama a María y Pedro ama a Rosa* como *Juan ama a* [*María y Pedro*] *ama a Rosa*.

3.5 Heurísticas

Las heurísticas se aplican después de las etapas descritas en las secciones anteriores. El propósito de las heurísticas es unir las palabras para las cuales no se encontró ningún gobernante en la etapa de aplicación de reglas.

El sistema actualmente usa las siguientes heurísticas, las cuales se aplican iterativamente en este orden, de una manera similar a como se aplican las reglas:

1. Un *que* suelto se une al verbo más cercano (a la izquierda o a la derecha del *que*) que no tiene otro *que* como su gobernante inmediato o indirecto.
2. Un pronombre suelto se une al verbo más cercano que no tiene un *que* como su gobernante inmediato o indirecto.
3. Un N suelto se une al verbo más probable que no tiene un *que* como su gobernante intermedio o indirecto. Para estimar la probabilidad, se usa un algoritmo similar a aquél descrito en la sección anterior. Las estadísticas usadas se describen a detalle en el módulo de Preferencias de Selección, Vea el Capítulo 4.
4. Para un verbo *v* suelto, el verbo *w* más cercano se busca hacia la izquierda; si no hay verbo a la izquierda, entonces se busca el más cercano a la derecha. Si *w* tiene un *que* como gobernante directo o indirecto, entonces *v* se une a este *que*. De otra forma, se une a *w*.
5. Un verbo suelto o una conjunción subordinada (excepto *que*) se une al verbo más cercano (a la izquierda o a la derecha del *que*) que no tenga otro *que* como su gobernante inmediato o indirecto.

Note que si la oración contiene más de un verbo, en el paso 4, cada verbo se adjunta a otro verbo, lo cual puede resultar en una dependencia circular. Sin embargo, esto no daña, puesto que dicha dependencia circular se romperá en la última etapa de procesamiento.

3.6 Selección de la raíz

La estructura construida en los pasos descritos del algoritmo descrito en las secciones anteriores puede ser redundante. En particular, puede contener dependencias circulares entre verbos. El paso final del análisis es seleccionar la raíz más apropiada.

Usamos la siguiente heurística simple para seleccionar la raíz. Para cada nodo en el digrafo obtenido, contamos el número de nodos alcanzables desde éste a través de un camino dirigido a lo largo de las flechas. La palabra que maximiza este número es seleccionada como raíz. En particular, todos los arcos entrantes a la raíz se eliminan de la estructura final

4 Preferencias de selección

En este capítulo presentamos diversos métodos para extraer preferencias seleccionales de verbos a partir de texto no anotado. En particular profundizamos en un método en el cual las preferencias seleccionales se vinculan a una ontología (por ejemplo, las relaciones de hiperónimos que se encuentran en WordNet). Vincular las preferencias seleccionales a una ontología permiten extender la cobertura en el caso de sustantivos que llenan las valencias y no han sido vistos previamente. Por ejemplo si *beber vodka* se encuentra en el corpus de entrenamiento, toda la jerarquía de WordNet que se encuentra por encima de *vodka* se vincula a este verbo: *beber licor*, *beber alcohol*, *beber bebida*, *beber sustancia*, etc.), de tal manera que cuando se busca *beber ginebra* es posible relacionar la preferencia de selección de *beber vodka* con *beber ginebra*, pues *ginebra* es un co-hipónimo de *vodka* y es también un *licor*. Este tipo de información puede ser utilizada para hacer desambiguación de frases de palabra, desambiguación de adjunción de frase preposicional y también de otros fenómenos sintácticos, y otras aplicaciones dentro del enfoque de métodos estadísticos combinados con conocimiento.

4.1 Introducción

Las preferencias de selección son patrones que miden el grado de acoplamiento de un argumento (objeto directo, objeto indirecto o complementos preposicionales) con un verbo. Por ejemplo, para el verbo *beber*, los objetos directos *agua*, *jugo*, y *leche* son más probables que *pan*, *ideas* o *pasto*.

Con objeto de tener una cobertura amplia de complementos posibles para un verbo, es necesario tener un corpus de consulta muy grande, de tal forma que prácticamente cualquier combinación de un verbo y un complemento pueda encontrarse en dicho corpus de consulta. Sin embargo, incluso para un corpus de cientos de millones de palabras, existen combinaciones de palabras que no ocurren en él. Algunas veces estas combinaciones de palabras no son usadas muy frecuentemente, o algunas son usadas a menudo pero no se encuentran en ciertos corpus de consulta.

Tabla 5. Usos no comunes (valores de ocurrencia más bajos) y usos comunes (valores de ocurrencia más altos) de combinaciones de palabras de verbo + synset de Wordnet

Verbo	Synset	Ocurrencias ponderadas
leer	*fauna*	0.17
leer	*comida*	0.20
leer	*mensaje*	27.13
leer	*escrito*	28.03
leer	*objeto_inanimado*	29.52
leer	*texto*	29.75
leer	*artículo*	37.20
leer	*libro*	41.00
leer	*comunicación*	46.17
leer	*periódico*	48.00
leer	*línea*	51.50
beber	*superficie*	0.20
beber	*vertebrado*	0.20
beber	*lectura*	0.20
beber	*sustancia*	11.93
beber	*alcohol*	12.50
beber	*líquido*	22.33
tomar	*artrópodo*	0.20
tomar	*clase_alta*	0.20
tomar	*conformidad*	0.20
tomar	*postura*	49.83
tomar	*resolución*	89.50
tomar	*control*	114.75
tomar	*acción*	190.18

Una solución para este problema es usar clases de palabras. En este caso, *agua*, *jugo*, *vodka* y *leche* pertenecen a la clase de *líquido* y pueden ser asociados con el verbo *beber*. Sin embargo, algunos verbos tienen más de una clase asociada con ellos. Por ejemplo, el verbo *tomar* puede tener argumentos de muchas clases distintas: *tomar asiento*, *tomar lugar*, *tomar tiempo*, *tomar agua*, etc. Por otra parte, cada palabra puede pertenecer a más de una clase. Esto no depende solamente del sentido de la palabra, sino de la característica principal que sea tomada en cuenta cuando se le asigne una clase. Por ejemplo, si consideramos el color de los objetos, *leche* pertenecería a la clase de los objetos blancos. Si consideramos las propiedades físicas, podría pertenecer a la clase de los fluidos o líquidos. *Leche* puede ser también un *alimento_básico* por ejemplo. Podemos decir entonces, que la clasificación relevante para una palabra depende tanto en su uso como en el sistema de clasificación que está siendo utilizado.

Para establecer una correlación entre el uso de un sustantivo, su sentido, y su preferencia de selección con respecto a un verbo, se requiere la siguiente información: 1) **Información ontológica**: una palabra no está vinculada a una sola clase, sino a una jerarquía completa, y 2) **Ejemplos del uso de la palabra** en oraciones, dado un verbo, y su posición específica en la ontología.

En esta sección de la tesis, proponemos un método para extraer preferencias de selección que están vinculadas a una ontología. El método presentado aquí forma parte de las principales aportaciones de este trabajo. La información obtenida es útil para resolver diversos problemas que requieren de una solución basada en métodos estadísticos combinados con conocimiento [156, 157].

La Tabla 5 presenta un ejemplo del tipo de información que obtenemos con nuestro método. La tabla muestra los valores de co-ocurrencia de los argumentos para tres distintos verbos en español usando la jerarquía de WordNet. Los números en dicha tabla fueron obtenidos siguiendo la metodología que se describirá en detalle en la sección 4.3. Note que los synsets que tienen una probabilidad más alta de ser un argumento para un verbo, tienen un valor más grande, como *beber líquido*. En contraste, los valores más bajos indican que es menos probable que un synset sea un argumento para el verbo correspondiente (por ejemplo *beber lectura, leer comida* o *beber superficie*). Estas combinaciones fueron encontradas debido a errores en el corpus de entrenamiento o debido a diversos sentidos no relacionados entre sí de una palabra. Por ejemplo, en *leer libro*, *libro* tiene varios sentidos, uno de ellos es el nombre de una de las partes en que se divide el estómago de los rumiantes. Como puede comerse, produce una entrada errónea, aunque muy baja, como **leer comida*. Cuando se usan corpus grandes para entrenamiento, este ruido se reduce sustancialmente en contraste con los patrones correctos. Es así como puede desambiguarse el sentido de un sustantivo basándonos en su uso junto con el verbo principal.

La Tabla 5 también muestra que los synsets que se encuentran más alto en la jerarquía de WordNet tienen valores más altos, puesto que acumulan el impacto de los hipónimos que se encuentran debajo de ellas (vea por ejemplo *comunicación*, *líquido* o *acción*). Una estrategia *ad-hoc* simple para ponderar los valores en la jerarquía de WordNet será descrita en la sección 4.3.

En las siguientes secciones mostraremos cómo obtenemos información como la mostrada en la Tabla 5, y luego evaluaremos nuestro método aplicando esta información a la desambiguación de sentidos de palabra (WSD por sus siglas en inglés).

4.2 Trabajo relacionado

Uno de los primeros trabajos sobre extracción de preferencias seleccionales vinculado a los sentidos de WordNet fue el trabajo de Resnik [158]. Su trabajo se enfocó principalmente a la desambiguación de sentidos de palabras en inglés. Resnik supuso que un texto anotado con sentidos de palabras era difícil de obtener, por lo cual basó su trabajo en texto etiquetado sólo morfológicamente. Posteriormente, Agirre y Martínez [1, 2] trabajaron vinculando el uso de los

verbos con sus argumentos. A diferencia de Resnik, Agirre y Martínez supusieron la existencia de texto anotado con sentidos de palabras, específicamente Sem-Cor, en inglés.

Otros sistemas supervisados para WSD incluyen a JHU [198], el cual ganó la competencia de Senseval-2, y a un sistema basado en máxima entropía por Suárez y Palomar [180]. El primer sistema combina, por medio de un clasificador basado en votación, diversos subsistemas de WSD basado en distintos métodos: listas de decisión [197], modelos vectoriales basados en cosenos, y clasificadores bayesianos. El segundo sistema hace una selección de la mejor característica para clasificar los sentidos de palabras y también usa un sistema de votación. Ambos sistemas tienen una puntuación de alrededor de 0.70 en las pruebas de Senseval-2.

Debemos tomar en cuenta que un recurso como Sen-Cor actualmente no está disponible para muchos lenguajes (en particular, español), y el costo de construirlo es alto. Es por esto que nosotros seguimos el enfoque de Resnik, en el sentido de asumir que no hay suficiente cantidad de texto anotado con sentidos de palabras. Inclusive, consideramos que el proceso de desambiguación de sentidos de palabra debe ser completamente automático, de tal forma que todo el texto que usamos se anote automáticamente con etiquetas morfológicas y etiquetas de partes gramaticales (POS-tags). Es por esto que nuestro sistema es completamente no-supervisado.

Trabajos anteriores realizados en sistemas no supervisados no ha alcanzado el mismo desempeño que los sistemas supervisados: Carroll y McCarthy [45] presentan un sistema que utiliza preferencias seleccionales para hacer desambiguación de sentidos de palabras y obtienen un 69.1% de precisión y un *recall* de 20.5%. Agirre y Martínez [3] presentan otro método, en esta ocasión no supervisado. Ellos utilizan la medida de *recall* como la única medida de desempeño, y reportan 49.8%; Resnik [158] logra un 40% de desambiguación correcta.

En las siguientes secciones describiremos nuestro método y mediremos su desempeño.

Tabla 6. Combinaciones seleccionadas extraídas del corpus CVV

	verb	relation	noun
1	*contar*	*con*	*permiso*
2	*pintar*	<	*pintor*
3	*golpear*	>	*balón*
4	*solucionar*	>	*problema*
5	*Dar*	>	*señal*
6	*haber*	>	*incógnita*
7	*poner*	*en*	*cacerola*
8	*beber*	*de*	*fuente*
9	*beber*	>	*vodka*

4.3 Vinculación a ontologías existentes

Con objeto de obtener las preferencias seleccionales vinculadas a una ontología, usamos las relaciones de hiperónimos del EuroWordNet[12] 1.0.7 (EWN-ES) como una ontología, y el corpus descrito en [84] como un corpus de entrenamiento (CVV). Este corpus de 38 millones de palabras fue creado con objeto de combinar los beneficios de un corpus virtual (como la Web como corpus) con los beneficios de un corpus local.

El texto se etiquetó morfológicamente usando el etiquetador estadístico TnT por Thorsten Brants [24] entrenado con el corpus CLiC-TALP. Este etiquetador tiene un desempeño de más de 92%, según se reporta en [134].

Después de que el texto se etiquetó morfológicamente, se extrajeron diversas combinaciones para cada oración:

(1) verbo + sustantivo a la izquierda (sujeto),

(2) verbo + sustantivo a la derecha (objeto), y

(3) verbo *cerca_de* preposición + sustantivo.

Aquí, + denota adyacencia, en tanto que *cerca_de* denota co-ocurrencia dentro de una oración. La Tabla 6 muestra un ejemplo de información obtenida de esta forma. El símbolo > significa que el sustantivo está a la derecha del verbo; el símbolo < significa que el sustantivo aparece a la izquierda del verbo.

[12] EWN-ES fue desarrollado conjuntamente por la Universidad de Barcelona (UB), la Universidad Nasional de Educación a Distancia (UNED), y la Universidad Politécnica de Cataluña, España.

Una vez que las combinaciones han sido extraídas, se busca el sustantivo para cada combinación en WordNet y se registra una ocurrencia para los synsets correspondientes (es decir, considerando todos los posibles sentidos para cada palabra). También se registra la ocurrencia para los hiperónimos de cada uno de estos synsets, con objeto de *propagar* el efecto de la combinación *hacia arriba*, es decir, hacia los términos más generales dentro de la red. Como puede el lector advertir, los nodos superiores en la ontología tendrán un impacto mayor, puesto que recogen los valores de cada uno de sus hijos. Es por esto que utilizamos un factor de ponderación de tal manera que los nodos que se encuentran más arriba en la jerarquía (hasta el nodo *entidad*) tienen un menor impacto que las palabras en la parte inferior de la jerarquía. Utilizamos un factor de ponderación simple $\frac{1}{level}$. Por ejemplo, suponga que *beber vodka* fue encontrado en el texto. Entonces se registra una ocurrencia para la combinación *beber vodka* con peso 1. También las ocurrencias para *beber licor* se registran, pero con peso 0.5, *beber alcohol* con 0.33, etc. Para cada combinación, los pesos de sus ocurrencias se acumulan, es decir, se suman.

Actualmente hemos obtenido 1.5 millones de patrones de preferencias seleccionales vinculados a los synsets de WordNet. Cada patrón consiste en un verbo, una preposición (en algunos casos), y un synset. Un ejemplo de la información obtenida puede ser visto en la Figura 9. *Canal* tiene 6 sentidos según se listan en WordNet: *camino*, *conducto* (camino), *conducto* (anatómico), *transmisión*, *depresión* y *agua*. Para *atravesar*, el sentido marcado con el número mayor de ocurrencias es *conducto*, en tanto que aquél marcado con un número menor de ocurrencias es *transmisión*, en el sentido de *canal de transmisión* o *canal de Televisión.* Por ejemplo, uno no *atraviesa* normalmente un canal de Televisión, pero sí *atraviesa* un *conducto*. Ahora considere *libro.* Esta palabra tiene cinco sentidos: *estómago*, *producto*, *sección*, *publicación* y *trabajo*/*juego*. El primer sentido hace referencia al nombre en español de una parte del estómago de los rumiantes. Podemos ver que este es el sentido con un número menor de ocurrencias junto con *leer*: uno no puede *leer* un *órgano interno*. El sentido con un número mayor de ocurrencias es aquél que se relaciona en el *lenguaje_escrito*. Esta información puede ser utilizada para desambiguar el sentido de la palabra, dado el verbo con el cual se utiliza. En la siguiente sección describiremos un experimento para medir el desempeño de las preferencias seleccionales extraídas con este método, en la tarea de desambiguación de sentidos de palabra.

4.4 Aplicación a desambiguación de sentidos de palabra

Senseval es una serie de competencias dirigido a evaluar los programas de desambiguación de sentidos de palabra, organizado por la ACL (Asociación de Lingüística Computacional) y SIGLEX

(grupo de interés especial en lexicografía de la ACL). Ha habido tres competiciones: Senseval-1, Senseval-2 y Senseval-3. La penúltima competición tuvo lugar en 2001 y la última en 2004. Los datos de las competencias están disponibles en línea. Para este experimento nos basaremos en los datos de Senseval-2. Esta competencia incluyó, entre 10 idiomas, al español, que es el que usaremos para aplicar nuestro método. El conjunto de evaluación comprende aproximadamente más de 1,000 oraciones. Cada oración contiene una palabra, para la cual se indica el sentido correcto entre aquellos listados para ella en WordNet.

Nuestra evaluación mostró que pudimos resolver 577 de 931 casos (un *recall* de ~62%). De estos, 223 corresponden finamente (fine-grained) con el sentido manualmente anotado (la precisión es de cerca de 38.5%). Estos resultados son similares a aquellos obtenidos por Resnik [158] para el inglés. Él obtuvo un promedio de 42.55% para las relaciones verbo-sujeto y verbo-objeto solamente. Note que en ambos casos los resultados son considerablemente mejores que una selección aleatoria de sentidos (alrededor de 28%, según reporta el mismo Resnik en [158]).

4.5 Discusión

Existen sistemas de desambiguación de sentidos de palabras (supervisados) que obtienen resultados superiores. Por ejemplo, Suarez y Palomar [180] reportan una puntuación de 0.702 para desambiguación de sustantivos para el mismo conjunto de evaluación de Senseval-2. Sin embargo, este sistema es supervisado, en tanto que el nuestro es no-supervisado. En comparación con los sistemas de desambiguación de sentidos de palabras (por ejemplo [158, 45, 3]) nuestro método tiene un *recall* superior, aunque precisión inferior en algunos casos. Esto último se debe a la estrategia de nuestro método que considera sólo relaciones de verbo-sustantivo, siendo que en ocasiones el sentido de la palabra está fuertemente vinculado al sustantivo precedente. Esto es particularmente cierto para pares de sustantivos que forman una sola frase preposicional. Por ejemplo, en el texto de entrenamiento aparece la siguiente oración:

La prevalencia del principio de libertad frente al principio de autoridad es la clave de Belle Epoque

En este caso, el sentido de *autoridad* está restringido de forma más fuerte por el sustantivo que la precede, *principio*, en contraste con el verbo principal *es*. El intentar desambiguar el sentido de *autoridad* usando las combinaciones *es* < *autoridad* y *es de autoridad*, no es la mejor estrategia para desambiguar el sentido de esta palabra en este caso.

atravesar canal:

02342911n → **camino** 3.00 → a_través 8.83 → artefacto 20.12 → objeto_inanimado 37.10 → entidad 37.63

02233055n → **conducto** 6.00 → camino 3.00 → a_través 8.83 → artefacto 20.12 → objeto_inanimado 37.10 → entidad 37.63

03623897n → **conducto** 5.00 → estructura_anatómica 5.00 → parte_del_cuerpo 8.90 → parte 7.22 → entidad 37.63

04143847n → **transmisión** 1.67 → comunicación 3.95 → acción 6.29

05680706n → **depresión** 2.33 → formación_geológica 2.83 → objeto_natural 14.50 → objeto_inanimado 37.10 → entidad 37.63

05729203n → **agua** 4.17 → objeto_inanimado 37.10 → entidad 37.63

leer libro:

01712031n → **estómago** 3.50 → órgano_interno 3.00 → órgano 3.08 → parte_del_cuerpo 3.75 → parte 4.35 → entidad 41.51

02174965n → **product** 14.90 → creación 13.46 → artefacto 34.19 → objeto_inanimado 36.87 → entidad 41.51

04214018n → **sección** 23.33 → escritura 33.78 → lenguaje_escrito 25.40 → comunicación 55.28 → relación_social 43.86 → relación 42.38 → abstracción 44.18

04222100n → **publicación** 16.58 → obra 7.95 → producto 14.90 → creación 13.46 → artefacto 34.19 →objeto_inanimado 36.87 → entidad 41.51

04545280n → **obra_dramática** 4.50 → escritura 33.78 → lenguaje_escrito 25.40 → communicación 55.28 →relación_social 43.86 → relación 42.38 → abstracción 44.18

Figura 9. Ontología con valores de uso para las combinaciones *atravesar canal* y *leer libro*.

4.6 Otras aplicaciones

Además de desambiguación de sentidos de palabras, la información de las preferencias seleccionales obtenida por este método puede ser utilizada para resolver otros problemas importantes, como desambiguación sintáctica. Por ejemplo, considere la frase en español *Pintó un pintor un cuadro*. En español es posible poner el sujeto a la derecha del verbo. Existe ambigüedad, puesto que no es posible decidir cuál sustantivo es el sujeto de la oración. Puesto que el español es un idioma con prácticamente un orden libre de palabras, incluso *Pintó un cuadro un pintor* tiene el mismo significado.

Para decidir cuál palabra es el sujeto (*cuadro* o *pintor*) es posible consultar a la ontología vinculada con preferencias seleccionales construida con el método presentado en este capítulo. Primero nos basamos en la base estadística de que el sujeto aparece a la izquierda del verbo en 72.6% de las ocasiones [131]. Posteriormente, la búsqueda de *un pintor pintó* regresa la siguiente cadena de hiperónimos con sus respectivos valores de ocurrencia: *pintor* → *artista* 1.00 → *creador* 0.67 → *ser_humano* 2.48 → *causa* 1.98. Finalmente, la búsqueda de *un cuadro pintó* regresa *escena*→ *situación* 0.42 → *estado* 0.34. Esto quiere decir, que *pintor* (1.00) es más probable como sujeto que

cuadro (0.42) para esta oración. Presentaremos una implementación a más grande escala de este método en los capítulos siguientes de esta tesis.

4.7 Resumen

En este capítulo hemos presentado un método para extraer preferencias seleccionales de verbos vinculados a una ontología. Es útil para resolver problemas de procesamiento de texto en lenguaje natural que requieren información acerca de la utilización de las palabras con un verbo en particular en una oración. Específicamente, hemos presentado un experimento que aplica este método para desambiguar los sentidos de palabras. Los resultados de este experimento muestran que aún existe camino por recorrer para mejorar los sistemas actuales de desambiguación de sentidos de palabras no supervisados usando preferencias seleccionales; sin embargo, hemos identificado puntos específicos para mejorar nuestro método bajo la misma línea de métodos estadísticos basados en patrones combinados con conocimiento.

5 Desambiguación de unión de frase preposicional

5.1 Introducción

En muchos lenguajes, las frases preposicionales (FP) como *en el jardín* pueden ser unidas a sintagmas nominales (SN): *un grillo en el jardín* o sintagmas verbales (SV): *juega en el jardín*. Algunas veces en una oración hay más de una posibilidad para la unión de FP. Por ejemplo, en *La policía acusó al hombre de robar* podemos considerar dos posibilidades: 1) el objeto del verbo es *al hombre de robar*, o 2) el objeto es *el hombre* y la acusación es *de robar*. Un ser humano sabe que la segunda opción es la correcta, sin embargo para que una máquina lo determine necesitamos un método que le permita hacerlo.

Existen diversos métodos para encontrar el lugar correcto de unión de FP, que se basan en estadísticas de *treebanks*. Estos métodos han reportado tener una precisión de hasta un 84.5% según [153, 28, 54, 125, 200, 74]. Sin embargo, el recurso de un treebank no está disponible para muchos lenguajes y puede ser difícil de obtener, de tal manera que un método que requiera menos recursos es deseable. Ratnaparkhi muestra en [154] un método que requiere sólo un etiquetador de categorías gramaticales e información morfológica. Su método usa texto simple para ser entrenado.

La calidad del corpus de entrenamiento determina significativamente la calidad de los resultados. Particularmente, para reducir los efectos del ruido en un corpus y para considerar una gran cantidad de fenómenos, se requiere un corpus muy grande. Eric Brill sustenta en [27] que es posible lograr precisión del estado del arte con métodos relativamente simples cuyo poder viene de la plétora de texto disponible para estos sistemas. Su artículo también da ejemplos de diversas aplicaciones de procesamiento de lenguaje natural que se benefician del uso de corpus muy grandes.

5.2 Unión de frases preposicionales usando Internet.

Volk ha propuesto dos variantes de un método que requiere un buscador en Internet para encontrar la adjunción de frase preposicional más probable. En esta sección describimos cómo aplicamos la última variante del método de Volk al español con mejoras que nos permiten lograr un mejor desempeño cercano al de los métodos estadísticos usando treebanks.

Hoy en día, corpus grandes comprenden más de 100 millones de palabras. También la Web puede verse como el corpus más grande con más de un billón de documentos. Particularmente para el español, Bolshakov y Galicia-Haro reportan aproximadamente 12,400,000 páginas que pueden encontrarse a través de Google [15]. Podemos considerar a la Web como un corpus que es grande y lo suficientemente diverso como para obtener mejores resultados con métodos estadísticos para procesamiento de lenguaje natural.

Usar la Web como corpus es una tendencia reciente que ha crecido. Un conteo de la investigación reciente que trata de aprovechar el potencial de la Web para el procesamiento de lenguaje natural puede ser encontrado en [106]. En particular, para el problema de encontrar la unión correcta de FP, Volk [186, 187] propone variantes de un método que consulta un buscador de Internet para encontrar la unión más probable de FP.

En este capítulo mostraremos los resultados de aplicar al español la última variante del método de Volk con modificaciones. En la sección 5.2.1 explicamos las distintas variantes del método de Volk. En la sección 5.2.2 presentamos las diferencias del método que usamos con respecto a su método. En la sección 5.2.3 explicamos los detalles de nuestro experimento y los resultados que obtuvimos.

5.2.1 El método de Volk

Volk propone dos variantes de un método para decidir la unión de una FP a un SN o a un verbo. En esta sección explicaremos ambas variantes y sus resultados.

5.2.1.1 Primer variante del método de Volk

Volk propone en [186] desambiguar las uniones de FP usando la Web como corpus considerando las frecuencias de co-ocurrencia (freq) de verbo+preposición contra aquellas de sustantivo+preposición. La fórmula utilizada para calcular la co-ocurrencia es:

```
cooc(X,P) = freq(X,P) / freq (X)
```

donde X puede ser ya sea un sustantivo o un verbo. Por ejemplo, para *Él llena el cuarto con libros,* N=*cuarto* P=*con* y V=*llena*. `cooc(X,P)` es un valor entre 0 (no se encontraron coocurrencias) y 1 (siempre ocurren juntos).

`freq (X,P)` se calcula consultando al buscador Altavista usando el operador `NEAR`: `freq(X,P) = query("X NEAR P")`.

Para decidir una unión, se calculan `cooc(N+P)` y `cooc(V+P)`. El valor más alto decide la unión. Si alguno de los valores de `cooc` es menos que un *umbral de co-ocurrencia mínima*, la unión no puede decidirse, y por tanto, no está cubierta. Ajustando el *umbral de co-ocurrencia mínima*, el

algoritmo de Volk 2000 puede lograr muy buena cobertura pero exactitud baja; o buena exactitud con una cobertura baja. La Tabla 7. muestra los valores de cobertura/exactitud de los experimentos de Volk.

Volk también concluye en [186] que usar formas completas es mejor que usar lemas.

Tabla 7. Cobertura y exactitud para el algoritmo de Volk.

umbral	cobertura	exactitud
0.1	99.0%	68%
0.3	36.7%	75%
0.5	7.7%	82%

El mismo experimento fue realizado para el holandés por Vandeghinste [185], logrando una cobertura de 100% y una exactitud de 58.4. Para lograr una exactitud de 75%, Vandeghinste usó un umbral de 0.606 lo cual dio una cobertura de sólo 21.6%.

5.2.1.2 Segunda variante

En un artículo subsecuente [187], Volk usa una fórmula diferente para calcular co-ocurrencias. Ahora se incluye el núcleo de la FP dentro de las consultas. La fórmula usada es:

$\texttt{cooc(X,P, N}_2\texttt{)} = \texttt{freq(X, P, N}_2\texttt{) / freq(X)}$

$\texttt{freq (X,P,N}_2\texttt{)}$ se calcula consultando el buscador de Altavista utilizando el operador NEAR: $\texttt{freq(X,P,N}_2\texttt{) = query(“X NEAR P NEAR N}_2\texttt{”)}$. X puede ser N_1 o V. Por ejemplo, para *Él llena el cuarto con libros*, N_1 = *cuarto*, P = *con*, N_2 = *libros* y V = *llena*.

Volk experimenta primero requiriendo que tanto $\texttt{cooc(N}_1\texttt{,P,N}_2\texttt{)}$ como $\texttt{cooc(V,P,N}_2\texttt{)}$ sean calculados para determinar un resultado. Posteriormente, considera usar un umbral para determinar la unión de FP cuando $\texttt{cooc(N}_1\texttt{,P,N}_2\texttt{)}$ o $\texttt{cooc(V,P,N}_2\texttt{)}$ no son conocidos. Esto es, si $\texttt{cooc(N}_1\texttt{,P,N}_2\texttt{)}$ no es conocido, $\texttt{cooc(V,P,N}_2\texttt{)}$ debe ser más grande que el umbral para decidir que la FP se une al verbo, y *vice versa*. Posteriormente, al incluir tanto lemas como formas completas, Volk logra un mejor desempeño, y al hacer que por omisión la unión sea al sustantivo para uniones no cubiertas, logra una cobertura del 100%. Los resultados que encontró se muestran en la Tabla 8.

Tabla 8. Resultados del método de Volk 2001

cobertura	exactitud	requiere tanto cooc(N1,P,N2) **como** cooc(V,P,N_2)	umbral cuando alguno de cooc(N1,P,N2) **ó** cooc(V,P,N2) no se conoce	incluye tanto lemas como formas completas en las consultas	por omisión la unión es al sustantivo
55%	74.32%	✓	NA		
63%	75.04%		0.001		
71%	75.59%		0.001	✓	
85%	74.23%		0	✓	
100%	73.08%		0	✓	✓

Para el holandés, requiriendo tanto cooc(N_1,P,N_2) como cooc(V,P,N_2), Vandeghinste obtiene una cobertura de 50.2% con una precisión de 68.92. Usando un umbral e incluyendo tanto lemas como formas completas en las consultas, logra una cobertura de 27% con una exactitud de 75%. Con una cobertura del 100%, uniendo los casos no cubiertos por omisión al sustantivo, se logra una exactitud de 73.08%.

5.2.2 Mejora del desempeño

Los métodos que resuelven la unión de FP basados en estadísticas de treebanks tienen un desempeño notablemente mayor que los experimentos descritos anteriormente. No obstante, creemos que existen varios elementos que pueden ser cambiados para mejorar los métodos basados en las consultas a Web. Uno de los elementos a considerar es el tamaño de la base de datos de documentos de los buscadores. Esto es relevante para encontrar frecuencias de co-ocurrencia representativas para cierto lenguaje. Es sabido que no todos los buscadores proporcionan los mismos resultados. Por ejemplo, la Tabla 9 muestra el número de co-ocurrencias encontradas en diversos buscadores para las mismas palabras:

Tabla 9. Número de co-ocurrencias encontradas en diversos buscadores.

	leer en el metro	*read in the subway*
Google	104	30
All-the-Web	56	23
Altavista	34	16
Teoma	15	19

Google está clasificado como el buscador con la base de datos más grande por "the search engine showdown"[13]. Debido a su mayor base de datos, hemos determinado que usar Google para obtener frecuencias de co-ocurrencia de palabras puede dar mejores resultados.

[13] Información tomada de www.searchengineshowdown.com, actualizado el of December 31st, 2002.

Otro elemento a considerar es el uso del operador `NEAR`. Decidimos no utilizarlo puesto que no garantiza que las palabras de la consulta aparezcan en la misma oración. Por ejemplo, considere las siguientes búsquedas en Altavista:

lavar `NEAR` con `NEAR` puerta	6,395 resultados	(1)
lavar `NEAR` con `NEAR` cloro	6,252 resultados	(2)

(1) muestra 6,395 páginas, incluso cuando las puertas no están relacionadas a la operación de lavar. Comparando con (2), que muestra 6,252 páginas encontradas, podemos ver que no existe una distinción clara de cuándo una preposición+verbo está relacionada a un verbo. Por otra parte, usar una frase exacta muestra 0 resultados, lo cual muestra una distinción clara entre "lavar con puerta" y "lavar con cloro". Los resultados encontrados fueron:

búsqueda de frase exacta	resultados	buscador
"lavar con puerta"	0	Altavista
"lavar con cloro"	100	Altavista
"lavar con puerta"	0	Google
"lavar con cloro"	202	Google

Siguiendo el enfoque del segundo trabajo de Volk [187], usamos tanto formas completas como formas le matizadas de sustantivos y verbos para obtener un mejor desempeño. Sin embargo, puesto que no estamos utilizando el operador `NEAR`, debemos considerar a los determinantes que pueden ser colocados delante del sustantivo o verbo y la preposición. También debemos considerar que el núcleo de una FP puede aparecer en plural, sin que esto afecte notablemente su uso. Para ilustrar esto, considere la siguiente oración[14]:

Veo al gato con un telescopio

Las uniones son calculadas por las consultas mostradas en la Tabla 10.

puesto que `freq(veo,con,telescopio)` es más alta que

`freq(gato,con,telescopio)`, se decid la unión como *veo con telescopio*.

[14] ejemplo tomado de [77]

Tabla 10. Consultas para determinar la unión de FP de

***Veo al gato con un telescopio* y *I see the cat with a telescope* en inglés.**

Veo al gato con un telescopio	resultados	I see the cat with a telescope	resultados
ver	296,000	see	194,000,000
"ver con telescopio"	8	"see with telescope"	13
"ver con telescopios"	32	“see with telescopes”	76
"ver con un telescopio"	49	"see with a telescope"	403
"ver con el telescopio"	23	"see with the telescope"	148
"ver con unos telescopios"	0	“see with some telescopes”	0
"ver con los telescopios"	7	“see with the telescopes”	14
veo	642,000		
"veo con telescopio"	0		
"veo con telescopios"	0		
"veo con un telescopio"	0		
"veo con unos telescopios"	0		
"veo con el telescopio"	1		
"veo con los telescopios"	0		
freq(veo,con,telescopio) =	**1.279×10^{-4}**	**freq(see,with,telescope) =**	**3.371×10^{-6}**
gato	185,000	cat	24,100,000
"gato con telescopio"	0	"cat with telescope"	0
"gato con telescopios"	0	“cat with telescopes”	0
"gato con un telescopio"	3	“cat with a telescope"	9
"gato con unos telescopios”	0	“cat with some telescopes”	0
"gato con el telescopio"	6	"cat with the telescope"	2
"gato con los telescopios"	0	“cat with the telescopes”	0
freq(gato,con,telescopio) =	**0.486×10^{-4}**	**freq(cat,with,telescope) =**	**0.456×10^{-6}**

5.2.3 Resultados experimentales

Para nuestra evaluación extrajimos aleatoriamente 100 oraciones del corpus LEXESP para el español [61] y del periódico Milenio Diario[15]. Todas las búsquedas fueron restringidas a sólo páginas en español.

Inicialmente, consideramos no restringir las consultas a un lenguaje específico, dado el beneficio que podíamos obtener de diversas palabras entre distintos lenguajes, como el francés y el español. Por ejemplo, la frase *responsables de la debacle* es usada en ambos lenguajes variando únicamente en su acentuación (*débâcle* en francés, *debacle* en español). Puesto que Google no toma en cuenta la acentuación de palabras, la misma consulta regresa resultados para ambos lenguajes. Sin embargo,

[15] www.milenio.com

con una búsqueda irrestricta Google regresa distintos conteos en su API[16] y en su interfaz Web[17] respectivamente. Por ejemplo, para *ver*, su interfaz Web muestra sólo 270,000 resultados, mientras que su API regresa más de 20,000,000, incluso después de activar el filtro de "agrupar resultados similares". Esta enorme desviación se reduce al restringir la consulta a un lenguaje específico. Para el español, una búsqueda restringida de *ver* en la interfaz Web regresa 258,000 resultados, mientras que la API regresa 296,000. Actualmente desconocemos la razón de esta diferencia, aunque no tiene un impacto serio en nuestro experimento.

Las oraciones de nuestro experimento tienen 181 casos de ambigüedad de unión de frase preposicional. De éstas, 162 pudieron ser resueltas automáticamente. Después de verificarlas manualmente, se determinó que 149 de ellas fueron resueltas correctamente y 13 fueron incorrectas.

En los términos de cobertura y exactitud usados por Volk, obtenemos una cobertura de 89.5% con una exactitud de 91.97%. Considerando todos los casos (cubiertos y no cubiertos), el porcentaje global de ambigüedades resueltas correctamente es de 82.3%

Hemos encontrado un incremento en el desempeño usando el método de Volk con las siguientes modificaciones: usando búsquedas de frase exacta en lugar del operador `NEAR`, usando un buscador con una base de datos más grande; buscando combinaciones de palabras que incluyen artículos determinados e indeterminados; y buscando formas singulares y plurales de palabras siempre que es posible. Los resultados obtenidos con este método (cobertura de 89.5%, exactitud de 91.97% y eficiencia global de 82.3%) están cercanos a aquellos obtenidos por los métodos que usan estadísticas de *treebank*, sin la necesidad de dichos recursos. Nuestro método puede ser probado en http://likufanele.com/ppattach.

5.3 Unión de FP usando preferencias de selección

En este capítulo hablaremos de un método no supervisado para generalizar la información de corpus locales por medio de la clasificación semántica de sustantivos basada en los 25 conceptos iniciales únicos de WordNet. Después, proponemos un método para usar esta información para unión de FPs.

---

[16] Google API es un servicio de Web que usa los estándares de SOAP y WSDL para permitir que un programa consulte directamente el buscador de google. Más información en http://api.google.com.

[17] http://google.com

Tabla 11. Ejemplos de ocurrencia de algunos verbos en español.

Tripleta	Ocurrencias	% del total de ocurrencias de verbos
ir a {actividad}	711	2.41%
ir a {tiempo}	112	0.38%
ir hasta {comida}	1	0.00%
beber {sustancia}	242	8.12%
beber de {sustancia}	106	3.56%
beber con {comida}	1	0.03%
amar a {agente_causal}	70	2.77%
amar a {lugar}	12	0.47%
amar a {sustancia}	2	0.08%

5.3.1 Introducción

Como ya hemos mencionado, existen diversos métodos para encontrar la unión de una FP. Los primeros métodos [153, 28] mostraron que se podía lograr una exactitud de hasta 84.5% usando estadísticas sobre treebanks. Kudo and Matsumoto [109] reportan una exactitud de 95.77% con un algoritmo que requiere semanas para entrenamiento, y Lüdtke and Sato [116] logran una exactitud de 94.9% requiriendo sólo 3 horas para entrenamiento. Estos métodos requieren un corpus anotado sintácticamente con marcas de bloques (chuks). No están disponibles corpus anotados de esta manera para todos los lenguajes, y el costo para construirlos puede llegar a ser considerablemente alto, si tomamos en cuenta el número de personas-hora que se requieren. En el capítulo anterior (5.2) mostramos un método que usa texto no anotado. Este método tiene una exactitud de 82.3%, utiliza la Web como corpus y por tanto puede ser lento: hasta 18 consultas para resolver una sola ambigüedad de frase preposicional, y cada par preposición+sustantivo en una oración multiplica este número.

Dicho algoritmo se basa en la idea de que un corpus muy grande tendrá suficientes términos representativos que permitirán la desambiguación de unión de FP. Puesto que hoy en día es posible tener corpus muy grandes localmente, hicimos experimentos para explorar la posibilidad de aplicar dicho método sin la limitación de requerir una conexión a Internet. Probamos con un corpus muy grande de 151 millones de palabras en 61 millones de oraciones. Este corpus fue obtenido en línea a partir de 3 años de publicaciones de 4 periódicos mexicanos. Los resultados fueron desalentadores: el mismo algoritmo que usaba la Web como corpus que daba un *recall* de casi 90%, tenía un *recall* de sólo 36% con una precisión de casi 67% usando el corpus local de periódicos.

Por tanto, nuestra hipótesis es que necesitamos generalizar la información contenida en los periódicos locales para maximizar el *recall* y la precisión. Una forma de hacer esto, es usar preferencias de selección: una medida de la probabilidad de un complemento que puede ser usado con cierto verbo, basada en la clasificación semántica del complemento. De esta manera, el problema de analizar *Veo un gato con un telescopio* puede ser resuelto considerando *Veo {animal} con {instrumento}*.

Por ejemplo, para desambiguar la adjunción de frase preposicional para la oración *Bebe de la jarra de la cocina*, las preferencias de selección proveen la información de que *de {lugar}* es un complemento poco común para el verbo *bebe*, y por tanto, la probabilidad de unir este complemento al verbo *bebe*, es baja. Entonces, se une al sustantivo *jarra*, dando como resultado *Bebe de [la jarra de la cocina]*.

La Tabla 11 muestra algunos ejemplos adicionales de números de ocurrencias para algunos verbos en español. De esta tabla podemos ver que el verbo *ir* es principalmente usado con el complemento *a {actividad}*. Combinaciones menos usadas tienen casi cero ocurrencias, como *ir hasta {comida}*. El verbo *amar* es usado a menudo con la preposición *a*.

En este capítulo proponemos un método para obtener información de preferencias de selección como la que se muestra en la Tabla 11. En la sección 5.3.2 mencionaremos trabajos relacionados a la desambiguación usando preferencias de selección. Las secciones 5.3.3 a la 5.3.5 explican nuestro método. Finalmente, después de integrar este módulo al sistema completo según se describió en el Capítulo 3, estaremos en condiciones de evaluar su desempeño. En la sección 6.1 presentamos esta evaluación con respecto a la desambiguación de unión de SP.

{comida}:	desayuno, festín, cereal, frijoles, leche, etc.
{actividad}:	abuso, educación, lectura, pesca, apuración, prueba
{tiempo}:	atardecer, historia, jueves, edad media, niñez
{sustancia}:	alcohol, carbón, chocolate, leche, morfina
{nombre}:	Juan, Pedro, América, Japón
{agente_causal}:	abogado, capitán, director, intermediario, nieto
{lugar}:	aeropuerto, bosque, pozo, valle, patio, rancho

Figura 10. Ejemplos de palabras para las categorías mostradas en la Tabla 11.

Tabla 12. Ejemplos de clasificaciones semánticas de verbos.

Palabra	Clasificación
rapaz	actividad
rapidez	actividad
rapiña	forma
rancho	lugar
raqueta	cosa
raquitismo	actividad
rascacielos	actividad
rasgo	forma
rastreo	actividad
rastro	actividad
rata	animal
ratero	agente causal
rato	lugar
ratón	animal
raya	actividad
	animal
	forma
rayo	actividad
raza	agrupación
razón	atributo
raíz	parte
reacción	actividad
reactor	cosa
real	agrupación
realidad	atributo
realismo	forma
realización	actividad
realizador	agente causal

5.3.2 Trabajo relacionado

Los términos *restricciones seleccionales* y *preferencias de selección* son relativamente nuevos, aunque existen conceptos similares presentes en trabajos como *Exploraciones en la Teoría Semántica* [191] o *La teoría de la Gramática Funcional, Parte I: La estructura de la Oración* [70]. Uno de los trabajos iniciales que usó estos términos fue *Restricciones Seleccionales: Un modelo de teoría de la información y su realización computacional* [157] donde Resnik considera restricciones seleccionales para determinar las restricciones que un verbo impone en su objeto. Las restricciones seleccionales tienen valores binarios: únicamente si un objeto de cierto tipo puede ser usado con un verbo o no. Las preferencias de selección, en cambio, son graduadas y miden, por ejemplo, la probabilidad de que un objeto pueda ser usado con cierto verbo [158]. Dichos trabajos usan un corpus analizado llanamente y un lexicón de

clases semánticas para encontrar las preferencias de selección para la desambiguación de sentidos de palabra.

Otro trabajo que usa clases semánticas para la desambiguación sintáctica es *Utilización de un léxico probabilística basado en clases para resolución de ambigüedad léxica* [149]. En este trabajo, Presecher *et al.* usan un algoritmo de clusterización-EM para obtener un lexicón probabilístico basado en clases. Este lexicón es usado para desambiguar palabras en traducción automática.

Un trabajo que usa particularmente clases de WordNet para resolver la unión de FP es *Un enfoque basado en reglas para la desambiguación de unión de frase preposicional* [28]. En este trabajo Brill y Resnik aplican el modelo basado en transformaciones, manejado por errores, para desambiguar la unión de FP. Ellos obtienen una exactitud de 81.8%. Este es un algoritmo supervisado.

Hasta donde sabemos, las preferencias de selección no han sido utilizadas en modelos no supervisados para desambiguación de unión de FP.

5.3.3 Fuentes para clasificación semántica de sustantivos

Una clasificación semántica para sustantivos puede ser obtenida a partir de WordNets existentes, usando un conjunto reducido de clases que corresponden a los conceptos iniciales únicos para sustantivos de WordNet descritos en *WordNet: Una base de datos léxica en-línea* [127]. Estas clases son: actividad, animal, forma_de_vida, fenómeno, cosa, agente_causal, lugar, flora, cognición, proceso, evento, sentimiento, forma, comida, estado, agrupación, sustancia, atributo, tiempo, parte, posesión y motivación. A estos conceptos únicos iniciales añadimos nombre y cantidad. Nombre corresponde a sustantivos propios no encontrados en el diccionario semántico y cantidad corresponde a números.

Puesto que no todas las palabras están cubiertas por WordNet y puesto que no existe una WordNet para cada lenguaje, las clases semánticas pueden ser obtenidas alternativamente de manera automática a partir de Diccionarios Explicativos Orientados al lector Humano (DEOH). Un método para hacer esto se explica en detalle en *Extracción de categorías semánticas para sustantivos para desambiguación sintáctica a partir de diccionarios explicativos orientados al humano* [34]. En la Tabla 13 mostramos ejemplos de clasificaciones semánticas de sustantivos extraídas del DEOH [110] usando este método.

5.3.4 Preparación de las fuentes para extraer preferencias de selección

Los periódicos y revistas son fuentes comunes de texto con calidad de media a buena. Sin embargo, usualmente estos medios exhiben una tendencia a expresar diversas ideas en poco espacio.

Esto causa que las oraciones sean largas y estén llenas de oraciones subordinadas, especialmente para lenguajes en los cuales un número ilimitado de oraciones pueden ser anidadas. Es por esto que uno de los primeros problemas a ser resueltos es dividir una oración en diversas sub-oraciones. Considere por ejemplo las oraciones mostradas en la Figura 11: es una sola oración extraída de un periódico mexicano.

Usamos dos tipos de delimitadores para separar las oraciones subordinadas: palabras delimitadores y patrones delimitadores. Ejemplos de palabras delimitantes son: pues, ya que, porque, cuando, como, por eso, y luego, con lo cual, mientras, con la cual, mientras que, etc. Ejemplos de patrones delimitadores se muestran en la Figura 12. Estos patrones están basados en categorías gramaticales. Esto significa que el texto fue analizado llanamente (shallow-parsed) antes de aplicarlos.

La oración en la Figura 11 fue separada usando esta técnica simple de tal forma que cada sub-oración se encuentra en un renglón distinto.

Y ahora, cuando
(el mundo) **está** gobernado por (las leyes del mercado), cuando
(lo determinante en la vida) **es**
comprar o
vender, sin
fijarse en <los que
carecen de todo>,
son fácilmente **comprensibles** <las razones de
<la ola de publicidad global que
convenció <a los posibles compradores de servicios y regalos > de que
había (grandes razones) para
celebrar> y
como les **pareciese** poco (el fin de año)
se **lanzaron** a
propagar (el fin del siglo y del milenio)

Figura 11. Ejemplo de una oración muy larga en un estilo típicamente encontrado en publicaciones.
( ) señalan SN simples; < > señalan SN subordinados, los verbos están en negritas

PREP V ,	CONJ PRON V	CONJ N V
V ADV que	PREP DET que N	PREP DET V
, PRON V	N que V	, N V
V PREP N , N V	, donde	N , que V
V PREP N , N PRON V	N , N	N , CONJ que
V PREP N V	CONJ N N V	N que N PRON V
V de que	CONJ N PRON V	CONJ PRON que V V

Figura 12. Patrones delimitantes:
V: verbo, PREP: preposición, CONJ: conjunción, DET: determinante,
N: sustantivo, las minúsculas son cadenas de palabras, ADV: adverbio, PRON:pronombre

5.3.5 Extracción de información de preferencias de selección

Una vez que las oraciones son etiquetadas y separadas, nuestro propósito es encontrar los siguientes patrones sintácticos:

1. Verbo $_{\text{CERCA\_DE}}$ Preposición $_{\text{JUNTO\_A}}$ Sustantivo
2. Verbo $_{\text{CERCA\_DE}}$ Sustantivo
3. Sustantivo $_{\text{CERCA\_DE}}$ Verbo
4. Sustantivo $_{\text{JUNTO\_A}}$ Preposición $_{\text{JUNTO\_A}}$ Sustantivo

Los patrones 1 a 3 serán llamados de aquí en adelante *patrones de verbos*. El patrón 4 será conocido como patrón de clasificación de sustantivos. El operador $_{\text{CERCA\_DE}}$ implica que podría haber otras palabras en medio. El operador $_{\text{JUNTO\_A}}$ implica que no hay palabras entre las palabras. Note que se preserva el orden de palabras, de tal manera que el patrón 2 es diferente del patrón 3. Los resultados de estos patrones se almacenan en una base de datos. Para verbos, se almacena el lema. Para sustantivos, se almacena su clasificación semántica, cuando está disponible a través del WordNet en español. Puesto que un sustantivo puede tener diversas clasificaciones, principalmente debido a sus diversos sentidos, se almacena un patrón distinto para cada clasificación semántica. Por ejemplo, vea la Tabla 13. Esta tabla muestra la información extraída para la oración de la Figura 11.

Tabla 13. Información de patrones semánticos extraída de la **Figura 11**

Palabras	Patrón
gobernado, *por*, *ley*	*gobernar*, *por*, cognición
gobernado, *de*, *mercado*	*gobernar*, *de*, actividad / cosa
es, *en*, *vida*	*ser*, *en*, estado / forma_de_vida / agente_causal / atributo
convenció, *a*, *comprador*	*convencer*, *a* agente_causal
convenció, *de*, *servicio*	*convencer*, *de*, actividad / proceso / posesión / cosa / agrupación
pareciese, *de*, *año*	*parecer*, *de*, cognición / tiempo
lanzaron, *de*, *año*	*lanzar*, *de*, cognición / tiempo
propagar, *de*, *siglo*	*propagar*, *de*, cognición / tiempo
propagar, *de*, *milenio*	*propagar*, *de*, cognición / tiempo
ley, *de*, *mercado*	cognición, *de*, actividad / cosa
ola, *de*, *publicidad*	evento, *de*, actividad / cognición
comprador, *de*, *servicio*	agente_causal *de*, actividad / proceso / posesión / cosa / agrupación
fin, *de*, *año*	lugar / cognición / evento / tiempo, *de*, cognición / tiempo
fin, de, *siglo*	lugar / cognición / evento / tiempo, *de*, cognición / tiempo

Una vez que se recolecta esta información, se cuenta la ocurrencia de los patrones. Por ejemplo, los dos últimos renglones en la Tabla 13, *fin, de, año* y *fin, de, siglo* añaden 2 de las siguientes ocurrencias: lugar de cognición, cognición de cognición, evento de cognición, tiempo de cognición, lugar de tiempo, cognición de tiempo, evento de tiempo, y tiempo de tiempo. Esta información se usa entonces para determinar la medida de preferencia de selección que un sustantivo tiene para un verbo o para otro sustantivo.

5.3.6 Aplicación de distintos métodos de suavizado

La unión de Frase Preposicional (FP) puede ser tratada tomando en cuenta el conteo de frecuencia de tripletas de dependencia vistas en un corpus no anotado. Sin embargo, no todas las tripletas buscadas se encuentran siempre incluso en un corpus muy grande. Para resolver este problema, existen diversas técnicas. Evaluamos dos métodos diferentes de suavizado, uno basado en WordNet y otro en un diccionario de sinónimos, antónimos e ideas afines distribucional (creado automáticamente). El diccionario de ideas afines distribucional es creado usando las tripletas de dependencia encontradas en el mismo corpus usado para contar la frecuencia de tripletas no ambiguas. El corpus de entrenamiento usado para ambos métodos es una enciclopedia. El método basado en un diccionario de ideas afines (DIA) distribucional tiene una cobertura más alta pero una precisión menor que el método basado en WordNet.

Para encontrar suficientes ocurrencias de dichas tripletas se necesita un corpus muy grande. Dichos corpus ahora están disponibles. También la Web puede ser utilizada [32, 187], sin embargo, aún con dichos corpus algunas combinaciones de palabras no ocurren. Este es un efecto conocido de la ley de Zipf: unas cuantas palabras son muy comunes pero existen muchas palabras que ocurren con una baja frecuencia [120]. Lo mismo ocurre con las combinaciones de palabras.

Para tratar el problema, se han explorado diversas técnicas de suavizado. En general, 'suavizar' consiste en buscar estadísticas para un conjunto de palabras, cuando hay datos insuficientes para la palabra en particular. De esta forma, *gato con telescopio* se convierte en {animal} *con* {instrumento} y *ver con telescopio* se convierte en *ver con* {instrumento}. Las palabras entre corchetes denotan conjuntos de palabras: {animal} = palabras de animales, {instrumento} = palabras para instrumento, etc.

Una forma de identificar el conjunto de palabras asociadas a una palabra dada es usando WordNet, y otra es usar un diccionario de ideas afines (DIA) distribucional. Un DIA distribucional es un DIA generado automáticamente a partir de un corpus buscando palabras que ocurren en contextos similares [88, 177, 190]. Ambos enfoques han sido explorados para el inglés y han mostrado resultados cercanos a la desambiguación humana. Vea la Tabla 14.

Experimentos que usan diversas técnicas han sido llevados a cabo independientemente, y a la fecha, no hay evaluaciones que comparen a WordNet con DIAs. En este capítulo comparamos estos dos enfoques según la metodología propuesta en *Diccionarios de ideas afines para procesamiento de lenguaje natural* [107]. Usamos un mismo corpus para ambos casos para permitirnos comparar los

resultados. El mismo corpus es usado para generar el DIA y las generalizaciones de WordNet. Este corpus es también utilizado para contar las tripletas de dependencia.

Nuestro trabajo es en español. Este, hasta donde sabemos, es el primer trabajo que explora métodos de suavizado para unión de FP para un lenguaje distinto al inglés.

5.3.6.1 Unión de FP sin suavizado

5.3.6.1.1 Construcción de los recursos

El recurso principal es el conteo de tripletas de dependencia. Para aumentar la cobertura y la eficiencia, en lugar de considerar palabras estrictamente adyacentes, contamos relaciones de dependencia entre lemas de palabras. Sólo se consideran relaciones no ambiguas de dependencia. Por ejemplo, las siguientes dos oraciones: *Veo con un telescopio. Un gato con tres patas está caminando*, darán las tripletas de dependencias: *ver, con, telescopio* y *gato, con, patas*, respectivamente. Sin embargo, la oración *Veo un gato con un telescopio* no dará ninguna tripleta de dependencias, pues es un caso ambiguo.

Extraemos todas las tripletas de dependencia siguiendo el proceso descrito en la sección 5.3.5: Primero etiquetamos el texto morfológicamente y después agrupamos adjetivos con sustantivos, y adverbios con verbos. Después, buscamos los patrones *verbo preposición sustantivo, sustantivo preposición sustantivo, sustantivo verbo,* y *verbo sustantivo*. Los determinantes, pronombres y otras palabras se ignoran.

Siguiendo el concepto de Lin [115], las tripletas de dependencias consisten en dos palabras y la relación gramatical (incluyendo preposiciones) entre estas dos palabras en la oración de entrada. Para ilustrar el tipo de tripletas de dependencias, considere un micro-corpus (μC) que consiste en dos oraciones: *Una mujer ve con un telescopio* y *La mujer con un sombrero ve un gato*. Las tripletas que corresponden a este μC se muestran en la Figura 14. Denotamos el número de ocurrencias de una tripleta <*w,r,w'*> como |*w,r,w'*|. Contando en μC, |*mujer,SUJ,ver*|=2 y |*mujer*,*con*,*sombrero*|=1. |*,*,*| denota el número total de tripletas (10 en μC), un asterisco * representa cualquier palabra o relación. En μC, |*ver,*,*| = 4, |*,*con,*| = 2, |*,*,mujer*| = 3.

Tabla 14. Estado del arte para desambiguación de Frase Preposicional.

Humanos (sin contexto)			Usan suavizado con WordNet			Usan suavizado con DIA		
Ratnaparkhi	[153]	88.2	Stetina and Nagao	[179]	88.1	Pantel and Lin	[142]	84.3
Mitchell	[130]	78.3	Li and Abe 1998	[114]	85.2	McLauchlan	[121]	85.0

$$x = |x,*,*| \quad p = |*,p,*| \quad n = |*,*,n_2| \quad t = |*,*,*| \quad \bar{x} = t - x\,, \quad \bar{p} = t - p\,, \quad \bar{n} = t - n$$

$$xpn = |x,p,n_2|\,, \quad \bar{x}pn = |*,p,n_2| - xpn\,, \; x\bar{p}n = |x,*,n_2| - xpn\,, \quad xp\bar{n} = |x,p,*| - xpn$$

$$\bar{x}\bar{p}n = n - xpn - \bar{x}pn - x\bar{p}n\,, \quad \bar{x}p\bar{n} = p - xpn - \bar{x}pn - xp\bar{n}$$

$$x\bar{p}\bar{n} = x - xpn - x\bar{p}n - xp\bar{n}\,, \; \bar{x}\bar{p}\bar{n} = t - \left(xpn + \bar{x}pn + x\bar{p}n + xp\bar{n} + \bar{x}\bar{p}n + \bar{x}p\bar{n} + x\bar{p}\bar{n}\right)$$

$$\begin{aligned} score = {} & xpn \cdot \log[xpn/(x \cdot p \cdot n/t^2)] + \bar{x}pn \cdot \log[\bar{x}pn/(\bar{x} \cdot p \cdot n/t^2)] + \\ & x\bar{p}n \cdot \log[x\bar{p}n/(x \cdot \bar{p} \cdot n/t^2)] + xp\bar{n} \cdot \log[xp\bar{n}/(x \cdot p \cdot \bar{n}/t^2)] + \\ & \bar{x}p\bar{n} \cdot \log[\bar{x}p\bar{n}/(\bar{x} \cdot p \cdot \bar{n}/t^2)] + \bar{x}\bar{p}n \cdot \log[\bar{x}\bar{p}n/(\bar{x} \cdot \bar{p} \cdot n/t^2)] + \\ & x\bar{p}\bar{n} \cdot \log[x\bar{p}\bar{n}/(x \cdot \bar{p} \cdot \bar{n}/t^2)] + \bar{x}\bar{p}\bar{n} \cdot \log[\bar{x}\bar{p}\bar{n}/(\bar{x} \cdot \bar{p} \cdot \bar{n}/\mathrm{t}^2)] \end{aligned}$$

para VScore, x es v, para NScore, x es n_1

Figura 13. Fórmulas para calcular similitud logarítmica de tres puntos.

ver, SUJ, mujer	ver, SUJ, mujer	ver, OBJ, gato
mujer, SUJ-DE, ver	mujer, SUJ-DE, ver	gato, OBJ-DE, ver
ver, con, telescopio	mujer, con, sombrero	
telescopio, con_r, ver	sombrero, con_r, mujer	

Figura 14. Tripletas de dependencias extraídas del micro-corpus (μC)

Las relaciones gramaticales sin preposiciones serán útiles después para construir un diccionario de ideas afines, donde la similitud de palabras será calculada basándose en contextos compartidos entre dos palabras. Por ahora, usaremos este recurso (DTC) sólo para contar tripletas de (*verbo*, *preposición*, *sustantivo*$_2$) y (*sustantivo*$_1$, *preposición*, *sustantivo*$_2$) para decidir una unión de FP. Esto lo explicaremos a detalle en la siguiente sección.

5.3.6.1.2 Aplicación de los recursos

La tarea es decidir la unión correcta de p,n_2 dada una 4-tupla de verbo, sustantivo$_1$, preposición, sustantivo$_2$: (v,n_1,p,n_2). La unión de p,n_2 puede ser ya sea al verbo v o al sustantivo n_1. El algoritmo no supervisado más simple une según el valor más alto entre VScore = $|v,p,n_2|$ y NScore = $|n_1,p,n_2|$. Cuando ambos valores son iguales, decimos que esta unión no es decidible por este método.

El corpus usado para contar las tripletas de dependencia (DTC) en este experimento fue la enciclopedia Encarta 2004 en español [126]. Tiene 18.59 millones de palabras, 117,928 lemas en 73MB de texto, 747,239 oraciones, y 39,685 definiciones. El corpus fue anotado con categorías gramaticales usando el etiquetador estadístico entrenado con el corpus etiquetado manualmente

Tabla 15. Diferentes fórmulas para calcular VScore y NScore

descripción		VScore	NScore
S	la más simple	$\lvert v,p,n_2\rvert$	$\lvert n_1,p,n_2\rvert$
S2	considerando bipletas también	$\lvert v,p,n_2\rvert \cdot \lvert v,p,*\rvert$	$\lvert n_1,p,n_2\rvert \cdot \lvert n_1,p,*\rvert$
LL3	Razón de similtud logartítmica	Vea la Figura 13	
Feat	Características simplificadas de Roth [142 y 164]	$\log(\lvert *,p,*\rvert/\lvert *,*,*\rvert) +$ $\log(\lvert v,p,n_2\rvert/\lvert *,*,*\rvert) +$ $\log(\lvert v,p,*\rvert/\lvert v,*,*\rvert) +$ $\log(\lvert *,p,n_2\rvert/\lvert *,*,n_2\rvert)$	$\log(\lvert *,p,*\rvert/\lvert *,*,*\rvert) +$ $\log(\lvert n_1,p,n_2\rvert/\lvert *,*,*\rvert) +$ $\log(\lvert n_1,p,*\rvert/\lvert v,*,*\rvert) +$ $\log(\lvert *,p,n_2\rvert/\lvert *,*,n_2\rvert)$

(con categorías gramaticales) CLiC-TALP[18] y lematizado usando el diccionario Anaya en español [111].

Una vez que el corpus se etiqueta y lematiza morfológicamente, se extraen las tripletas de dependencia. Encarta produjo 7 millones de tripletas de dependencias (algunas de ellas repetidas) entre las cuales hubo 3 millones de tripletas diferentes. 0.7 millones de tripletas (0.43 millones diferentes) involucraron preposiciones.

Usamos cuatro diferentes fórmulas para calcular VScore y NScore, las cuales se listan en la Tabla 15. Las primeras dos fórmulas (S y S2) pueden ser vistas como el cálculo de la probabilidad de cada tripleta, es decir, $p(v,p,n_2)=\lvert v,p,n_2\rvert/\lvert *,*,*\rvert$. Puesto que tanto VScore y NScore se dividen por el mismo número, $\lvert *,*,*\rvert$, para compararlas éste puede ser omitido sin ninguna diferencia. Para las fórmulas de similitud logarítmica[19], vea la Figura 13.

Siguiendo el método de evaluación de unión de FP propuesto por Ratnaparkhi *et al.* [153], la tarea será determinar la unión correcta dada una 4-tupla (v,n_1,p,n_2). Extrajimos 1,137 4-tuplas, junto con su unión correcta (sustantivo o verbo) del corpus etiquetado manualmente Cast-3LB[20] [136]. Puede ver 4-tuplas de muestra en la Tabla 16.

Tabla 16. Ejemplo de 4-tuples (v,n_1,p,n_2) usadas para evaluación

4-tuplas
informar comunicado del Banco_Central N
producir beneficio durante periodo V
defender resultado de elección N
recibir contenido por Internet V
planchar camisa de puño N

[18] http://clic.fil.ub.es. El etiquetador TnT entrenado con el corpus CLiC-TALP tiene un desempeño de más de 92%

[19] La similitud logarítmica fue calculada usando el paquete estadístico Ngram, vea [7].

[20] Cast-3LB es parte del proyecto 3LB, financiado por el Ministerio de Ciencia y Tecnología de España. 3LB, (FIT-150500-2002-244 and FIT 150500-2003-411)

Los valores de referencia (punto de partida) pueden definirse de dos maneras: La primera es asignando todas las uniones a *sustantivo*$_1$. Esto da una precisión de 0.736. La segunda se basa en el hecho de que la preposición *de* se une a un sustantivo en 96.9% de las 1,137 4-tuplas.[21] Esto da una precisión de 0.855, un valor alto para un punto de partida, considerando que el nivel de acuerdo humano es 0.883. Para evitar estos valores de referencia iniciales altamente tendenciosos, optamos por excluir todas las 4-tuplas con preposición *de*—ninguna otra preposición presenta dicha predisposición tan grande. Entonces todas nuestras observaciones están hechas usando sólo 419 de las 1,137 4-tuples extraídas. La línea de referencia en este caso consiste en asignar todas las uniones al verbo, lo cual da una precisión de 66.1%. El acuerdo entre etiquetadores humanos para 4-tuples excluyendo la preposición *de* es de 78.7%, sustancialmente menor que el acuerdo humano para todas las 4-tuples. Los resultados se muestran en la Tabla 17.

La precisión más alta está dada por la fórmula S2, por lo cual a partir de ahora usaremos esta fórmula para comparar los resultados entre métodos de suavizado.

5.3.6.2 Suavizado con WordNet

5.3.6.2.1 Construcción del diccionario

Estamos buscando una mayor cobertura de relaciones de dependencia para decidir una unión correcta de FP. Para lograr esto, construimos un diccionario que usa WordNet para encontrar una generalización de las relaciones de dependencia. Por ejemplo, buscamos la generalización de *comer*

Tabla 17. Comparación entre formulas para calcular VScore y NScore

Método	Cobertura	Precisión
Valor de referencia inicial (*baseline*)	1.000	0.661
S	0.127	0.750
S2	0.127	**0.773**
LL3	0.127	0.736
Feat	0.127	0.717

con tenedor, *comer con cuchara*, y *comer con cuchillo* en *comer con* {artículos de mesa}. Note que {artículos de mesa} no es una palabra, sino un concepto en WordNet. WordNet provee el conocimiento de que *tenedor*, *cuchara*, y *cuchillo* son {artículos de mesa}. De esta forma, si se encuentra una tripleta que no ha sido vista anteriormente, como *comer con palillos*, WordNet puede

[21] Esto es válido también para el ingles. Del conjunto de entrenamiento provisto por Ratnaparkhi, la preposición *of* 'de', se une al sustantivo en 99.5% de las 20,801 4-tuplas.

ayudar diciendo que *palillos* son {artículos de mesa} también, de tal forma que podemos aplicar nuestro conocimiento acerca de *comer con* {artículos de mesa}.

Antes de describir nuestro método, permítasenos introducir un poco de notación. Cada palabra *w* está unida a uno o más synsets en WordNet en correspondencia con sus diferentes sentidos. W_n denota al synset que corresponde al *n*-avo sentido de *w,* y *N* el número total de sentidos. Cada uno de estos synsets tiene diversos caminos a la raíz siguiendo sus hiperónimos. W_n^m denota el *m*-avo hiperónimo del sentido *n*-avo de *w,* y M_n la profundidad, es decir, el número de hiperónimos atravesados hasta la raíz para el sentido número *n*.

Por ejemplo, *banco* en WordNet en español tiene 8 sentidos. El cuarto hiperónimo del tercer sentido de *banco* se denota como W_3^4 = *utillaje_5*. Para ilustrar esto, vea a continuación un extracto para *banco* de WordNet.

sentido 3: *banco* (asiento) → mueble_3 → mueblaje_4 → utillaje_5 → artefacto_6 → objeto_inanimado_7 → entidad_8

sentido 4: banco (grupo biológico) → grupo_biológico_2 → grupo_3

sentido 5: *banco* (institución financiera) → institución_3 → organización_4 → grupo_social_5 → grupo_ 6

Nuestro método de suavizado en WordNet está basado en *Aprendizaje no supervisado de preferencias de selección vinculadas a una ontología* [35] y *Estimación de probabilidad basada en clases usando una jerarquía semántica* [52]. Para propagar un *score* (NScore or VScore) a través de WordNet, debemos considerar todas las tripletas que involucran la misma *w* y *r*, variando *w'* (como en el caso de aprender *comer con* {artículos de mesa} a partir de diversos ejemplos de *comer con* *). Este conjunto de tripletas se denota por <*w,r,**>. Para cada *w*' involucrada, distribuimos uniformemente[22] cada *score* s(*w,r,w'*) entre cada uno de los sentidos de *w'* (como en [158]). Después este resultado se propaga a todos los hiperónimos W_n^m. Este valor es acumulativo: nodos superiores en WordNet recolectan información de todos sus hijos. De esta manera, los conceptos más generales totalizan el uso (frecuencia de las tripletas) de sus conceptos específicos (hipónimos).

[22] Asumimos una distribución equiprobable, lo cual puede ser problemático. Sin embargo, no existen textos etiquetados extensamente para el español a partir de los cuales podamos extraer distribución de sentidos.

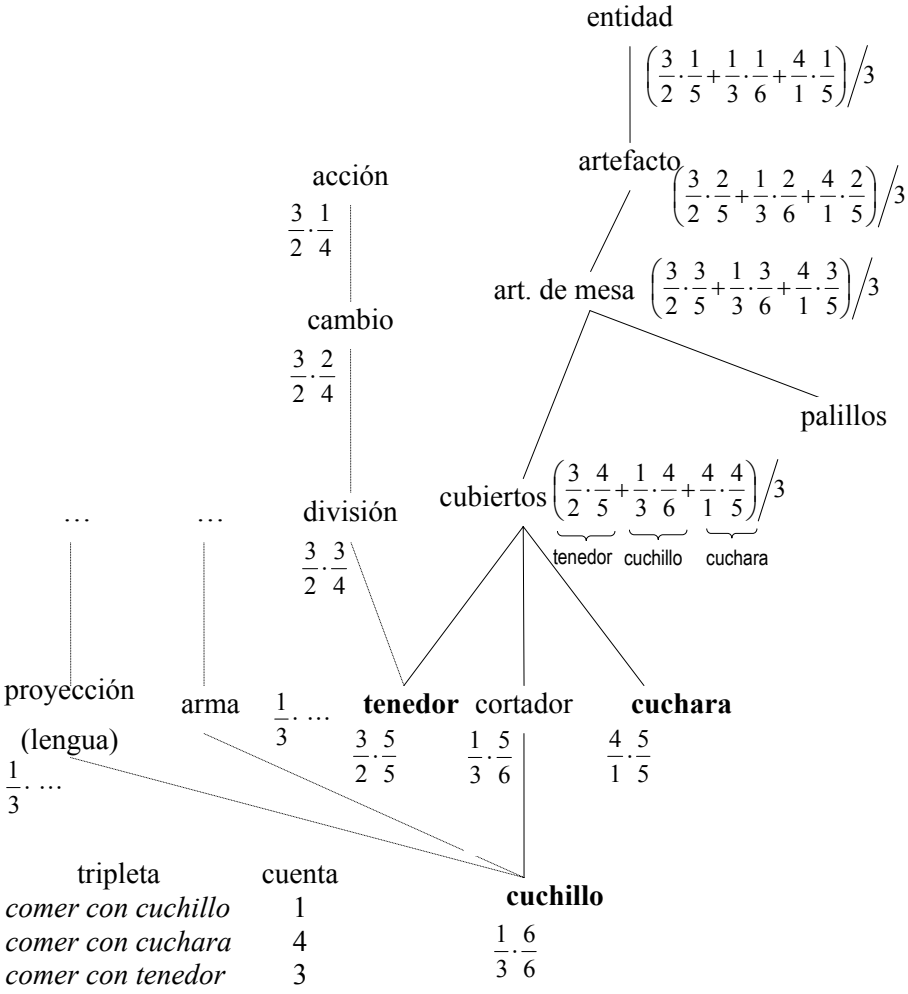


Figura 15. Ejemplo de propagación de cuentas de tripletas en WordNet

Para evitar la sobre-generalización (esto es, la acumulación excesiva en niveles superiores), la profundidad debe considerarse. Algunas veces la profundidad de la cadena de hiperónimos es muy larga (como en el sentido 3 ó 5 de *banco*) y algunas veces más pequeña (sentido 4 de *banco*). Una fórmula útil de propagación que permite la generalización y considera la profundidad de las cadenas de hiperónimos es:

$$s(w,r,W_n^m) = [s(w,r,w')/\mathrm{N}] \times [1-(m-1/M_n)] \quad (1)$$

Adicionalmente, el número de tripletas que contribuyen a cierto nodo de WordNet se cuenta para promediar niveles superiores. Esto es, después de considerar las *k* tripletas <*w,r,**>, contamos el número de tipos de tripletas que contribuyen a cada nodo. Luego, el valor de cada nodo se divide por ese número.

Tabla 18. Ejemplos de tipos de tripletas (*w,r,w'*) con suavizado con WordNet

w	*r*	*w'*	English	score
comer	*con*	*mano*	hand	3.49
'eat'	'with'	*cubiertos*	cutlery	1.31
		tenedor	fork	1.19
matar	*con*	*arma*	weapon	0.27
'kill'	'with'	*armamento*	armaments	0.23
		utillaje	utensil	0.18

contamos el número de tipos de tripletas que contribuyen a cada nodo. Posteriormente, el valor de cada nodo se divide entre dicho número.

Para ilustrar nuestro algoritmo, vea la Figura 15. Para este ejemplo, suponga que sólo tenemos tres tripletas, cada una de las cuales se lista con su cuenta en la Figura 15. El conteo de frecuencias de cada tripleta se añade a la palabra correspondiente en WordNet. Para *comer con tenedor*, el nodo para la palabra *tenedor* es etiquetado con 3 palabras para *comer con*. *tenedor* puede ser utilizado con otras combinaciones de palabras, pero aquí mostramos sólo valores para *comer con*, es decir, <*w,r,**>.

De acuerdo con la Fórmula (1), este valor se divide por el número de sentidos de *tenedor*. En este ejemplo, asumimos dos diferentes sentidos para *tenedor*, con distintos hiperónimos cada uno: {*división*} y {*cubiertos*}. Al enfocarnos en la rama de {*cubiertos*}, podemos ver cómo se propaga este valor hacia {*entidad*}. Para esta rama hay 5 niveles de profundidad desde {*entidad*} a *tenedor* (M_2=5). La otra rama tiene 4 niveles (M_1=4). Siguiendo la propagación de *tenedor* hacia arriba en el árbol, puede verse cómo cada nivel tiene un factor de peso menor: para {*artículos_de_mesa*} es 3/5 y para {*entidad*} sólo 1/5. Cada nodo es acumulativo; por esto, {*cubiertos*} acumula los valores para *tenedor*, *cuchillo* y *cuchara.* El valor para {*cubiertos*} se divide entre 3 porque el número de tipos que contribuyen a este nodo es 3. Si tuviéramos otra tripleta: *comer con palillos*, entonces {*cubiertos*} permanecería sin cambios, pero {*artículos_de_mesa*} se dividiría por 4.

Para este experimento usamos EuroWordNet en español[23] 1.0.7 (S-EWN) [71]. Tiene 93,627 synsets (62,545 sustantivos, 18,517 adjetivos, 12,565 verbos), 51,593 relaciones de hipónimos/hiperónimos, 10,692 relaciones de merónimos y 952 entradas de información de rol (sustantivo agente, instrumental, de locación o paciente). Propagamos todas las tripletas de dependencias en el DTC (corpus de cuenta de tripletas de dependencias) usando la Fórmula (1) (la creación de DTC se explicó en la sección 5.3.6.1.1.)

[23] S-EWN fue desarrollado conjuntamente por la Universidad de Barcelona (UB), la Universidad Nacional de Educación Abierta (UNED), y la Universidad Politécnica de Cataluña (UPC), España.

El algoritmo de suavizado presentado en esta sección produce subjetivamente buenos resultados. En la Tabla 18 se listan las primeras tripletas que califican con *con* como una relación para dos verbos comunes del español.

5.3.6.2.2 Utilización del diccionario

Para decidir una unión a FP dada una 4-tupla (v,n_1,p,n_2), calculamos NScore para (n_1,p,n_2) y VScore para (v,p,n_2) según se explicó en la sección 5.3.6.1.2. El valor más alto determina la unión. El suavizado de WordNet se aplica cuando una tripleta no se encuentra. En este caso, n_2 se sustituye por sus hiperónimos hasta que el *score* para la nueva tripleta (x,p, W_n^m) se encuentre a partir de *scores* previamente calculados en WordNet. Cuando se calcula NScore, x es n_1, y cuando se calcula VScore, x es v. El *score* más alto determina la unión. Note que estamos suavizando n_2 únicamente. Decidimos no suavizar v porque la estructura de verbos en S-EWN tiene muy pocas relaciones de hiperónimos para verbos (7,172) y la definición del hiperónimo de un verbo no es clara en muchos casos. Puesto que no suavizamos v, tampoco podemos suavizar n_1, puesto que esto introduciría una desviación de los NScores contra los VScores. También note que W_n^m es un synset específico en la jerarquía de WordNet, y por tanto tiene un sentido específico. El problema de desambiguar el sentido de n_2 se resuelve eligiendo el valor más alto de cada conjunto de sentidos en cada capa de hiperónimo (vea [35] y [179] para desambiguación de sentidos de palabras usando información de unión de Frase Preposicional). Los resultados para este método serán presentados en el capítulo de evaluación, en la sección 6.2.

Siguiendo el ejemplo de la Figura 15, suponga que queremos calcular el VScore para *comer con palillos*. Puesto que esta tripleta no se encuentra en nuestro corpus de conteo de frecuencias, buscamos los hiperónimos para *palillos*, en este caso {artículos_de_mesa}. Luego, el valor de este nodo se usa para calcular VScore.

5.3.6.3 Suavizado con diccionario de ideas afines (DIA)

5.3.6.3.1 Construcción del diccionario

Aquí describimos la construcción automática de un diccionario de ideas afines (DIA) de tal forma que las palabras no encontradas en las tripletas de dependencia puedan ser sustituidas por palabras similares. Esta medida de similitud está basada en el trabajo de Lin [142]. Este DIA está basado en la medida de similitud descrita en *Una medida de similitud según teoría de la información* [115]. La similitud entre dos palabras w_1 y w_2 según la define Lin es:

$$sim_{lin}(w_1, w_2) = \frac{\sum_{(r,w)\in T(w_1)\cap T(w_2)} (I(w_1, r, w) + I(w_2, r, w))}{\sum_{(r,w)\in T(w_1)} I(w_1, r, w) + \sum_{(r,w)\in T(w_2)} I(w_2, r, w)}$$

$$I(w, r, w') = \log \frac{|w, r, w| \times |*, r, *|}{|w, r, *| \times |*, r, w'|}$$

$T(w)$ es el conjunto de pares (r,w') tales que $I(w,r,w')$ es positivo. El algoritmo para construir el DIA es el siguiente:

```
para cada palabra-tipo w1 en el corpus
|    para cada palabra-tipo w2 en el corpus
|    |      sims(w1) ← {simlin(w1,w2), w2}
|    ordenar sims(w1) por similitud en orden descendiente
```

Como en caso del suavizado usando WordNet, este método da resultados subjetivos satisfactorios: la Tabla 19 muestra las 3 palabras más similares a *guitarrista*, *devoción* y *leer*.

5.3.6.3.2 Uso del diccionario

Para decidir una unión a FP en una 4-tupla (v,n_1,p,n_2), nuestro algoritmo calcula NScore para (n_1,p,n_2), y VScore para (v,p,n_2) según se explicó en la sección 5.3.6.1.2. El *score* más alto determina la unión. Cuando no se encuentra una tripleta, el algoritmo de suavizado se aplica. En este caso, n_2 se sustituye por su palabra más similar n'_2 calculada usando $sim_{lin}(n_2, n'_2)$. Si la nueva tripleta (x,p,n'_2) se encuentra en la cuenta de tripletas de dependencias (DTC), entonces se usa para calcular el *score*. Si no se encuentra, entonces la siguiente palabra más similar se prueba como sustitución de la palabra anterior, hasta que la nueva tripleta (x,p,n'_2) se encuentre. Cuando se calcula NScore, x es n_1; cuando se calcula VScore, x es v. El valor más alto determina la unión. El algoritmo se muestra a continuación. Cuando n=1, la palabra n-ésima más similar es la primera palabra más similar. Por ejemplo, *pianista* para *guitarrista*. Para n=2, sería *fisiólogo*, etc.

Tabla 19. Ejemplo de palabras similares usando el método de similitud de Lin

palabra w	palabra similar w'	$sim_{lin}(w,w')$
guitarrista	*pianista*	0.141
	fisiólogo	0.139
	educador	0.129
devoción	*afecto*	0.095
	respeto	0.091
	admiración	0.078
leer	*editar*	0.078
	traducir	0.076
	publicar	0.072

```
Para decidir la unión en (v,n1,p,n2):

     VSCore = cuenta(v,p,n2)
     NScore = cuenta(n1,p,n2)
     n, m ← 1
     si NScore = 0
     |  mientras NScore = 0 & existe una palabra n-ésima más similar a n2
     |  |       simn2  ← n-ésima palabra más similar a n2
     |  |       factor ← sim(n2,simn2)
     |  |       NScore ← count(n1,p,simn2) × factor
     |  └       n      ← n + 1
     si VScore = 0
     |  mientras VScore = 0 & existe una palabra n-ésima más similar a n2
     |  |       simn2  ← m-ésima palabra más similar a n2
     |  |       factor ← sim(n2,simn2)
     |  |       VScore ← cuenta(n1,p,simn2) × factor
     |  └       m      ← m + 1

     si NScore = VScore entonces no se puede decidir
     si NScore > Vscore entonces la unión es a n1
     si NScore < Vscore entonces la unión es a v
```

6 Evaluación del sistema

En esta sección mostraremos los resultados de evaluar nuestro sistema a diversos niveles. Primeramente evaluaremos el módulo de desambiguación sintáctica, usando los módulos presentados en el capítulo 5, incluyendo el resultado de la comparación entre diversos métodos de suavizado, según nuestra investigación presentada en la sección 5.3.6. Posteriormente presentamos una evaluación global de nuestro sistema, con todos sus módulos en funcionamiento en la sección 6.3. Para realizar esta evaluación global, requerimos de un estándar de referencia, para lo cual utilizamos un corpus anotado sintácticamente con algunas anotaciones semánticas. Este corpus se encuentra en el formato de constituyentes. Los detalles de su conversión a formato de dependencias se presentan en la sección 6.3.2.

6.1 Evaluación del módulo de desambiguación sintáctica

El procedimiento explicado en la sección 5.3 anterior se aplicó a un corpus de 161 millones de palabras que comprenden más de 3 años de artículos de cuatro distintos periódicos mexicanos. Tomó aproximadamente tres días en una Pentium IV obtener 893,278 distintas preferencias de selección para los patrones de verbos (patrones 1 al 3) para 5,387 raíces de verbos, y 55,369 distintas preferencias de selección para patrones de clasificación de sustantivos (patrón 4).

6.1.1.1 Desambiguación de unión de FP

Para evaluar la calidad de las preferencias de selección obtenidas, las probamos en la tarea de desambiguación de FP. Considere los primeros dos renglones de la Tabla 13, correspondientes al fragmento de texto *gobernado por las leyes del mercado*. Este fragmento reportó dos patrones de preferencias de selección: *gobernar por* {cognición} y *gobernar de* {actividad/cosa}. Con las preferencias de selección obtenidas, es posible determinar automáticamente la unión correcta de FP: se comparan los valores de co-ocurrencia para *gobernar por* {cognición} y *gobernar de* {actividad/cosa}. El valor más alto define la unión.

Formalmente, para decidir si el sustantivo N_2 se une a su sustantivo precedente N_1 o se une al verbo V de la sub-oración local, se comparan los valores de frecuencia para las uniones usando la siguiente fórmula [187]:

$$freq(X, P, C_2) = \frac{occ(X, P, C_2)}{occ(X) + occ(C_2)}$$

donde X puede ser V, un verbo, o C_1, la clasificación del primer sustantivo N_1. P es una preposición, y C_2 es la clasificación del segundo sustantivo N_2. Si $freq(C_1, P, C_2) > freq(V, P, C_2)$, entonces la unión se decide al sustantivo N_1. De otra forma, se decide que la unión sea al verbo V. Los valores de $occ(X, P, C_2)$ son el número de ocurrencias del patrón correspondiente en el corpus. Vea la Tabla 11 para ejemplos de ocurrencias de verbos. Ejemplos de ocurrencias de clasificación tomadas de un periódico en español son: {lugar} *de* {cognición}: 354,213, {lugar} *con* {comida}: 206, {lugar} *sin* {flora}: 21. Los valores de *occ(X)* son el número de ocurrencias del verbo o de la clasificación del sustantivo en el corpus. Por ejemplo, para {lugar} el número de ocurrencias es 2,858,150.

6.1.1.2 Evaluación

La evaluación fue llevada a cabo en 3 diferentes archivos del corpus LEXESP [169], el cual contiene 10,926 palabras en 546 oraciones. En promedio, este método logró una precisión de 78.19% y un recall de 76.04%. Los detalles de cada archivo procesado se muestran en la Tabla 20.

6.1.1.3 Resumen

Usar preferencias de selección para desambiguar la unión de FP tuvo una precisión of 78.19% y un recall de 76.04%. Estos resultados no son tan buenos como aquellos obtenidos con otros métodos, los cuales pueden llegar a lograr una exactitud de hasta 95%. Sin embargo, nuestro método no requiere ningún recurso costoso como un corpus anotado, ni una conexión a Internet (para usar la Web como corpus); ni siquiera se requiere el uso de una jerarquía semántica (como WordNet), puesto que las clases semánticas pueden ser obtenidas a parir de Diccionarios Explicativos Orientados al lector Humano, como se expuso en 5.3.3.

Tabla 20. Resultados de unión de Frase Preposicional usando preferencias de selección.

archivo	#oraciones	palabras	promedio de pals./ oración	tipo de texto	precisión	recall
n1	252	4,384	17.40	noticias	80.76%	75.94%
t1	74	1,885	25.47	narrativo	73.01%	71.12%
d1	220	4,657	21.17	deportes	80.80%	81.08%
total:	546	10,926		**promedio**:	78.19%	76.04%

Tabla 21. Resultados de nuestros experimentos para desambiguación de unión de FP.

Método	Cobertura	Precisión	Promedio
Acuerdo manual (humano)	1.000	0.787	0.894
Unión a verbo por omisión (valor de referencia inicial)	1.000	0.661	0.831
Sin suavizado	0.127	**0.773**	0.450
Suavizado con WordNet	0.661	0.693	0.677
Suavizado con diccionario de ideas afines (DIA) distribucionales	**0.740**	0.677	**0.707**

Encontramos también que, al menos para esta tarea, aplicar técnicas que usan el Web como corpus a corpus locales reduce el desempeño de estas técnicas en más del 50%, incluso si los corpus locales son muy grandes.

Para mejorar los resultados de desambiguación de unión de FP usando preferencias de selección, nuestra hipótesis es que en lugar de usar únicamente las 25 clases semánticas superiores, pueden obtenerse clases intermedias usando una jeararquía completa. De esta forma, sería posible tener una particularización flexible para términos comúnmente usados juntos, es decir, colocaciones, como 'fin de año', en tanto que se mantiene el poder de la generalización. Otro punto de desarrollo posterior es añadir un módulo de desambiguación de sentidos de palabras, de tal manera que no se tengan que considerar todas las clasificaciones semánticas para una sola palabra, como se mostró en la sección, según se describió en la sección 5.3.5.

6.2 Evaluación de métodos de suavizado

En esta sección comparamos los resultados de los tres métodos: sin suavizado, suavizado con WordNet y suavizado con DIA (diccionario de ideas afines). Los resultados se listan en la Tabla 21, junto con los valores de referencia iniciales (baseline) y el acuerdo manual humano. La tercera columna muestra el promedio entre la cobertura y la precisión. Note que los valores de referencia mostrados en la Tabla 21 involucran cierto conocimiento supervisado: la mayoría de las uniones, después de excluir los casos *de*, son a verbo. Los valores de precisión, cobertura y promedio más altos están mostrados en negritas. Después de excluir los casos *de*, tenemos 419 instancias. Para 12.7% de ellas los tres algoritmos hacen prácticamente lo mismo, así que las diferencias entre el suavizado de WordNet y el tesauro distribucional están basadas en los restantes 366 casos.

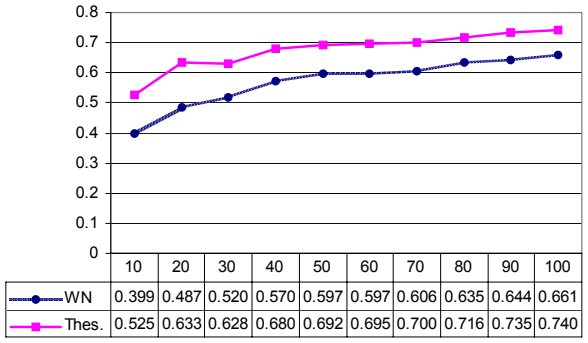
No todos los casos están cubiertos por estos métodos de suavizados ya sea porque no se puede encontrar sustituto para una palabra (como diversos acrónimos o nombres propios) o porque después de probar todas las posibles sustituciones la tripleta no se encontró en el DTC (Corpus de cuenta de tripletas de dependencia). En general, la cobertura es baja debido al tamaño del corpus para contar las frecuencias de unión. Si bien una enciclopedia provee un texto con muchas palabras

diferentes, el número de uniones de FP extraídas es relativamente bajo. Creemos que usando un corpus más grande conducirá a medidas de cobertura más altas, aunque se mantendrá la misma relación entre los métodos de suavizado estudiados.

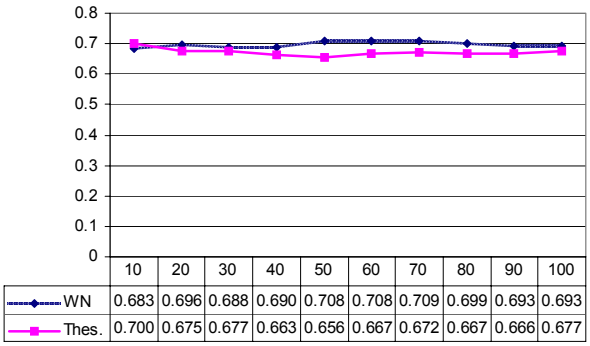
Para confirmar esto, realizamos porcentajes parciales elegidos al azar del corpus DTC. Esto se muestra en la Figura 16. Note que estamos usando un modelo totalmente no supervisado. Esto es, en ambos algoritmos no utilizamos ninguna otra técnica de suavizado para los casos no cubiertos.

6.2.1 Resumen

Entre los tres métodos evaluados para la unión de FP, la mejor medida promedio fue 0.707 usando suavizado con DIA, debido a su cobertura más grande comparada con otros métodos. Sin embargo, tiene una precisión menor que el suavizado usando WordNet. El método sin suavizado tiene una cobertura muy baja (0.127) pero para las uniones cubiertas los resultados fueron los mejores: 0.773, lo cual es cercano al acuerdo humano (recuerde que este acuerdo se calcula excluyendo una preposición que causa mucha desviación: *de*, la cual prácticamente siempre se une a los sustantivos). El desempeño del suavizado con WordNet podría incrementarse añadiendo información de la distribución de sentidos para cada palabra, en lugar de asumir una distribución



	10	20	30	40	50	60	70	80	90	100
WN	0.399	0.487	0.520	0.570	0.597	0.597	0.606	0.635	0.644	0.661
Thes.	0.525	0.633	0.628	0.680	0.692	0.695	0.700	0.716	0.735	0.740



	10	20	30	40	50	60	70	80	90	100
WN	0.683	0.696	0.688	0.690	0.708	0.708	0.709	0.699	0.693	0.693
Thes.	0.700	0.675	0.677	0.663	0.656	0.667	0.672	0.667	0.666	0.677

Figura 16. Precisión y cobertura usando distintos porcentajes de cuentas de tripletas (0–100%)

equiprobable, aunque esto acercaría a nuestro método a los enfoques supervisados, además de que no existe un recurso actualmente que provea las distribuciones de sentidos para el español.

Nuestros resultados indican que un recurso construido automáticamente (en este caso, un diccionario de ideas afines) pueden ser usados en lugar de uno construido manualmente, y aún así obtener resultados similares.

En trabajos posteriores podría explorarse usando corpus mucho más grandes para reunir cuentas de tripletas, así como experimentar con algoritmos más sofisticados que usen diccionarios de ideas afines para determinar uniones.

6.3 Evaluación global del extractor de estructura de dependencias con roles semánticos

Para realizar esta evaluación global, requerimos de un estándar de referencia, para lo cual utilizamos un corpus anotado sintácticamente con algunas anotaciones semánticas. Este corpus se encuentra en el formato de constituyentes. Los detalles de su conversión a formato de dependencias se presentan en las secciones 6.3.1 y 6.3.2. Posteriormente en la sección 6.3.3 mostraremos los resultados de nuestra evaluación.

6.3.1 Construcción de árboles de dependencias

Hemos seguido el esquema de evaluación propuesto por Briscoe *et al*. [29], el cual sugiere evaluar la exactitud de los analizadores sintácticos, basándose en las relaciones gramaticales entre núcleos léxicos lematizados. Este esquema es adecuado para evaluar los analizadores de dependencias y los analizadores de constituyentes también, porque considera las relaciones en un árbol que están presentes en ambos formalismos. Por ejemplo [Det *coche el*] y [ObjetoDirecto *tirar lo*]. Para evaluar, extrajimos tripletas de los árboles de dependencias encontrados por nuestro método, y lo comparamos con las tripletas extraídas manualmente del treebank 3LB.

Una tripleta es una relación de dependencias entre un nodo padre con un nodo hijo y el tipo de su relación. Por ejemplo, las tripletas de dependencias extraídas de la frase *El hombre viejo ama a la mujer joven* son:

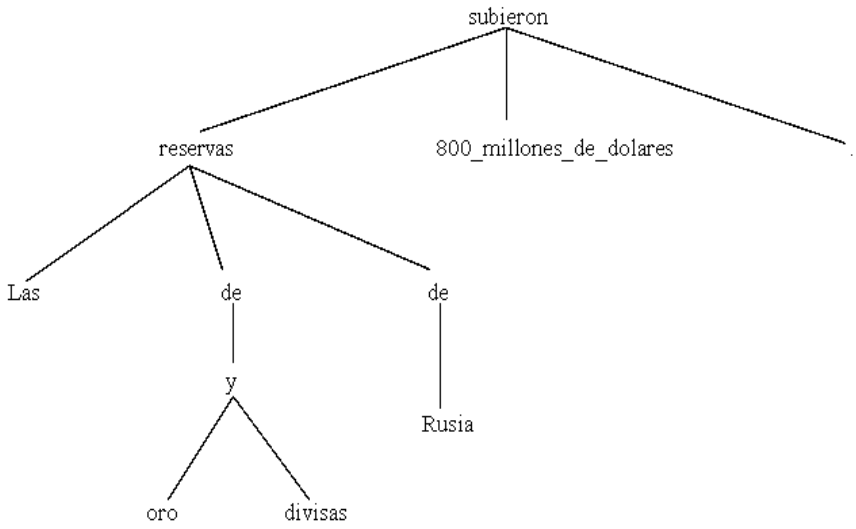


Figura 17. Árbol de dependencias resultante sin etiquetas, de la oración "Las reservas de oro y divisas de Rusia subieron 800 millones de dólares"

ama SUJ hombre
hombre DET el
hombre ADJ viejo
ama OBJ mujer
mujer DET la
mujer ADJ joven

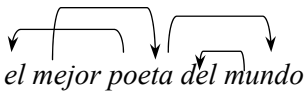
6.3.2 Conversión automática entre estructuras de constituyentes y estructuras de dependencias

En esta sección comentaremos sobre nuestra experiencia en convertir el corpus Cast3LB del español. El método general de evaluación consiste en extracción de una gramática libre de contexto del corpus etiquetado, identificación automática del elemento rector en cada regla, y usando esta información para la construcción del árbol de dependencias. Nuestras heurísticas identifican el elemento rector de las reglas con precisión de 99% y cobertura de 80%, con lo que el algoritmo identifica correctamente 92% de las relaciones de dependencias entre las palabras.

Marcadas automáticamente	Marcada manualmente
infinitiu <-- van0000 vmp00sm sps00 @infinitiu	infinitiu <-- van0000 @vmp00sm sps00 infinitiu
S.F.C.co-CD <-- conj.subord S.F.C @coord S.F.C	S.F.C.co-CD <-- @conj.subord S.F.C coord S.F.C

Figura 18. **Reglas que no coincidieron.**

En este trabajo mostramos cómo la información faltante puede añadirse automáticamente a un árbol de estructura por sintagmas para convertirlo en un árbol de dependencias. Obviamente, dicha conversión no puede ser completamente precisa porque las dos representaciones son (discutiblemente) no equivalentes cuando se trata de construcciones gramaticales más extrañas como construcciones no proyectivas:



Esta sección está organizada de la siguiente manera: La sección 6.3.2.1 describe brevemente el corpus que fue la base de nuestros experimentos. La sección 6.3.2.2 presenta en detalle nuestro procedimiento de transformación y las heurísticas que usamos en la conversión de este corpus específico. La sección 6.3.2.4 trata sobre los resultados experimentales de esta conversión.

6.3.2.1 El treebank 3LB en español

Cast3LB es un corpus de 100 mil palabras, aproximadamente 3,700 oraciones creado a partir de dos corpus: el corpus CLiCTALP (75 mil palabras), un corpus balanceado anotado morfológicamente que contiene lenguaje literario, periodístico, científico, etc., y el corpus de la agencia noticiosa española EFE (25 mil palabras) correspondiente al año 2000.

El proceso de anotación fue llevado en dos etapas, en la primera se seleccionó un subconjunto del corpus y se anotó dos veces por dos diferentes anotadores. Los resultados de este doble proceso de anotación se compararon y se asignó una tipología de desacuerdos en asignaciones de sentidos. Después de un proceso de análisis y discusión, se produjo un manual de anotación, donde se describen los criterios a seguir en caso de ambigüedad. En el segundo paso, el resto del corpus se anotó siguiendo la estrategia de todas las palabras. Los elementos léxicos anotados son aquellos con significado léxico: sustantivos, verbos y adjetivos [136].

6.3.2.2 Procedimiento de transformación

El proceso de transformación puede ser descrito brevemente como sigue:

1. Extraer las reglas gramaticales de constituyentes a partir del Treebank 3LB.
2. Marcar los núcleos: usar heurísticas para encontrar el componente nuclear de cada regla.
3. Usar esta información de las cabezas para encontrar cuál componente subirá en el árbol.

En las siguientes secciones describiremos estos pasos en detalle.

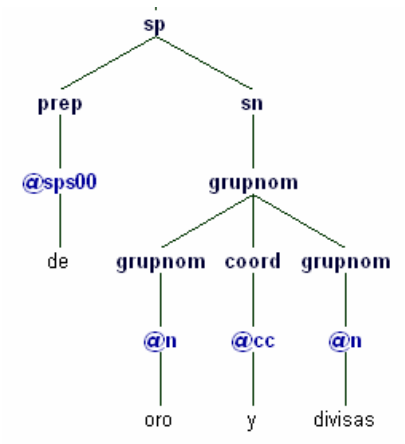
(S	oración
(S.F.C.co-CD	oración
(S.F.C	oración
(sn-SUJ	frase nominal
(espec.fp	especificador
(da0fp0 Las el))	determinante femenino plural *la*
(grup.nom.fp	grupo nominal femenino plural
(ncfp000 reservas reserva)	sustantivo femenino plural
(sp	frase preposicional
(prep	preposición
(sps00 de de))	preposición
(sn	sintagma nominal
(grup.nom.co	grupo nominal coordinante
(grup.nom.ms	grupo nominal masculino singular
(ncms000 oro oro))	sustantivo masculino singular
(coord	coordinante
(cc y y))	coordinante
(grup.nom.fp	grupo nominal femenino plural
(ncfp000 divisas divisa)))))	sustantivo femenino plural
(sp	frase preposicional
(prep	preposición
(sps00 de de))	preposición
(sn	sintagma nominal
(grup.nom	grupo nominal
(np00000 Rusia Rusia))))))	sustantivo propio
(gv	grupo verbal
(vmis3p0 subieron subir))	verbo masculino singular tercera persona plural
(sn-CC	sintagma nominal
(grup.nom	groupo nominal
(Zm 800_millones_de_dolares 800_millones_de_dolares))))	número

Figura 19. Una oración con etiquetas originales a partir del treebank 3LB. “Las reservas de oro y divisas de Rusia subieron 800 millones de dólares”

6.3.2.3 Extracción de la gramática

El método para extraer la gramática del Treebank 3LB consta de los siguientes pasos:

Simplificación del Treebank de constituyentes: El Treebank 3LB divide las etiquetas en dos partes. La primera especifica la categoría gramatical (por ejemplo oración, sustantivo, verbo, frase nominal, etc.) Esta es, para nuestros propósitos, la parte más importante de la etiqueta. La segunda parte especifica características adicionales como género y número para frases nominales, o el tipo de oración subordinada. Estas características pueden ser eliminadas con objeto de disminuir el número de reglas gramaticales sin que se afecte por esto el proceso de transformación. Por ejemplo, para cierta oración, el 3LB usa S (oración), S.F.C. (oración subordinada), o S.F.C.co_CD (oración subordinada de objeto). Canalizamos las tres a una sola etiqueta: *S*. Para grupos nominales, 3LB usa grup.nom (grupo nominal), grup.nom.fp (grupo nominal femenino plural), grup.nom.ms (grupo nominal masculino singular), grup.nom.co (grupo nominal coordinado), etc; canalizamos todos




```
(sp
   (prep
      (@sps00 de de))
   (sn
      (grupnom
         (grupnom
            (@n oro oro))
         (coord
            (@cc y y))
         (grupnom
            (@n divisas divisa)))))
```

Figura 20. Los nodos que tienen sólo una hoja son marcados como núcleos

ellos a una sola etiqueta grupnom. La Figura 19 muestra una parte de 3LB que usa las etiquetas originales.

Para reducir el número de patrones en la gramática resultante, también simplificamos el etiquetado de 3LB eliminando todas las marcas de puntuación.

Extracción de patrones. Para extraer todas las reglas de la gramática, cada nodo con más de un hijo se considera como la parte izquierda de una regla, y sus hijos son la parte derecha de la regla. Por ejemplo, los patrones extraídos de la Figura 21 se muestran en la Figura 22. Aquí *grupnom* es grupo nominal, *coord* es coordinante, *sp* es sintagma preposicional, *prep* es preposición, *sn* es sintagma nominal, *n* es sustantivo, *espec* es especificador, *S* es oración y *gv* es grupo verbal. Una oración (*S*) puede estar compuesta por una frase nominal (*sn*), frase verbal (*gv*) y sintagma nominal (*sn*).

6.3.2.3.1 Marcaje de los núcleos

Después de extraer todos los patrones que forman la gramática, se marca el núcleo de cada patrón automáticamente usando heurísticas simples. Denotamos el núcleo de una regla con el símbolo @. Las heurísticas que usamos son las siguientes:

1. Si la regla contiene un elemento (o sólo uno de sus elementos) puede ser núcleo (vea las heurísticas 10, 11), es el núcleo. Ejemplo:

   grupnom ← @n

2. Si el patrón contiene un coordinante (*coord*) entonces es el núcleo. Ejemplo:

grupnom ← grupnom @coord grupnom

S ← @coord sn gv sn

3. Si el patrón contiene dos o más coordinantes, el primero es el núcleo. Ejemplo:

   S ← @coord S coord S

   Sp ← @coord sp coord sp

4. Si el patrón contiene un grupo verbal (*gv*), será el núcleo. Ejemplo:

   S ← sn @gv sn

   S ← sadv sn @gv S Fp

5. Si el patrón contiene un pronombre relativo (*relatiu*), éste será el núcleo. Ejemplo:

   sp ← prep @relatiu

   sn ← @relatiu grupnom

6. Si el patrón contiene una preposición (*prep*) como su primer elemento seguido de un solo elemento, cualquiera que éste sea, la preposición será el núcleo. Ejemplo:

   sp ← @prep sn

   sp ← @prep sp

7. Si el patrón contiene un verbo en infinitivo (*infinitiu*), será el núcleo. Ejemplo:

   S ← @infinitiu S sn

   S ← conj @infinitiu

   S ← neg @infinitiu sa

8. Si el patrón contiene un presente participio (*gerundio*), éste será el núcleo. Ejemplo:

   S ← @gerundi S

9. Si el patrón contiene un verbo principal (*vm*), éste será el núcleo. Ejemplo:

   gv ← va @vm

   infinitiu ← va @vm

10. Si el patrón contiene un verbo auxiliar (*va*) y cualquier otro verbo, el verbo auxiliar nunca será el núcleo:

    gv ← va @vs

11. Si el patrón contiene un especificador (*espec*) como su primer elemento, nunca será el núcleo:

    sn ← espec @grupnom

    sn ← espec @sp

12. Para patrones con sintagma nominal (*grupnom*) como nodo padre, si el patrón contiene un sustantivo (*n*), será el núcleo. Por ejemplo:

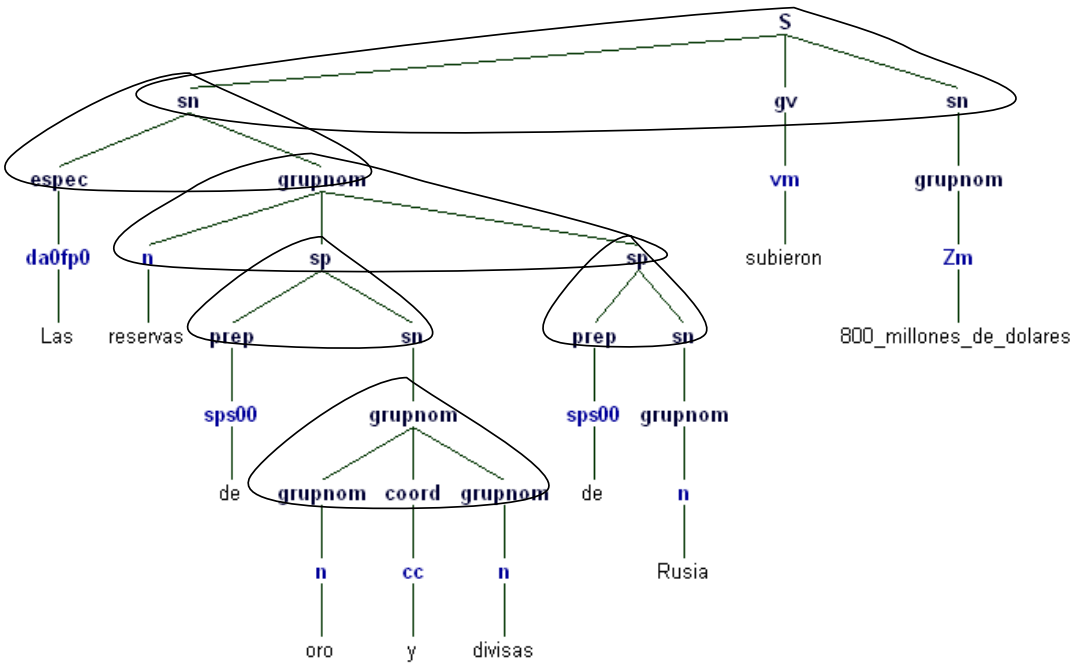


Figura 21. **Árbol con los patrones de la oración: "Las reservas de oro y divisas de Rusia subieron 800 millones de dólares"**

grupnom ← s @n sp

grupnom ← @n sn

grupnom ← s @n S

13. Para patrones con sintagma nominal (*grupnom*) como su nodo padre, si el patrón contiene sintagma nominal, éste será el núcleo. Por ejemplo:

    grupnom ← @grupnom s

    grupnom ← @grupnom sn

14. Para patrones con especificador (*espec*) como nodo padre, si el patrón contiene un artículo definido (*da*), éste será el núcleo:

    espec ← @da di

    espec ← @da dn

15. Si el patrón contiene un adjetivo calificativo (*aq*) y un sintagma preposicional (*sp*), el adjetivo es el núcleo, e.g.:

    S ← sadv @aq sadv

    sa ← sadv @aq sp sp

grupnom ← grupnom coord grupnom
sp ← prep sn
grupnom ← n sp sp
sn ← espec grupnom
S ← sn gv sn

Figura 22. Patrones extraídos de la oración "Las reservas de oro y divisas de Rusia subieron 800 millones de dólares"

El orden de aplicación de las reglas heurísticas es importante. Por ejemplo, si aplicamos la regla 2 en el patrón `S ← coord sn gv sn Fp`, el núcleo sería *gv*, en lugar de marcar el núcleo correcto *coord.* Para que esto ocurra, la Regla 1 debe aplicarse primero.

6.3.2.3.2 Uso de los núcleos marcados para la transformación

El algoritmo de transformación usa recursivamente la información de los patrones marcados con núcleos para determinar qué componentes subirán en el árbol. Esto quiere decir desconectar la cabeza de sus hermanos y ponerla en la posición del nodo padre. A continuación describiremos con más detalle el algoritmo:

1. Recorrer el árbol de constituyentes en profundidad de izquierda a derecha, comenzando por la raíz y visitando los nodos hijos recursivamente.
2. Para cada patrón en el árbol, buscar en las reglas para encontrar cuál elemento es el núcleo.
3. Marcar el núcleo en el árbol de constituyentes. Desconectarlo de sus hermanos y ponerlo en el lugar del nodo padre.

El algoritmo termina cuando un nodo núcleo sube como raíz. Para ilustrar el método, presentaremos un ejemplo. Vea las figuras Figura 23 y Figura 25.

La Figura 23 muestra un árbol de constituyentes que será convertido en un árbol de dependencias. Recuerde que los nodos que tienen sólo una hoja se marcaron en la gramática de extracción.

De acuerdo con el algoritmo, el primer patrón a buscar es `grupnom ← grupnom coord grupnom`, donde *grupnom* es un grupo nominal y *coord* es un coordinante.

Buscando en las reglas encontramos que el núcleo de estos patrones es el coordinante (*coord*). Marcamos el núcleo en el árbol de constituyentes y lo desconectamos colocándolo en la posición del nodo padre, como se muestra en la Figura 25.

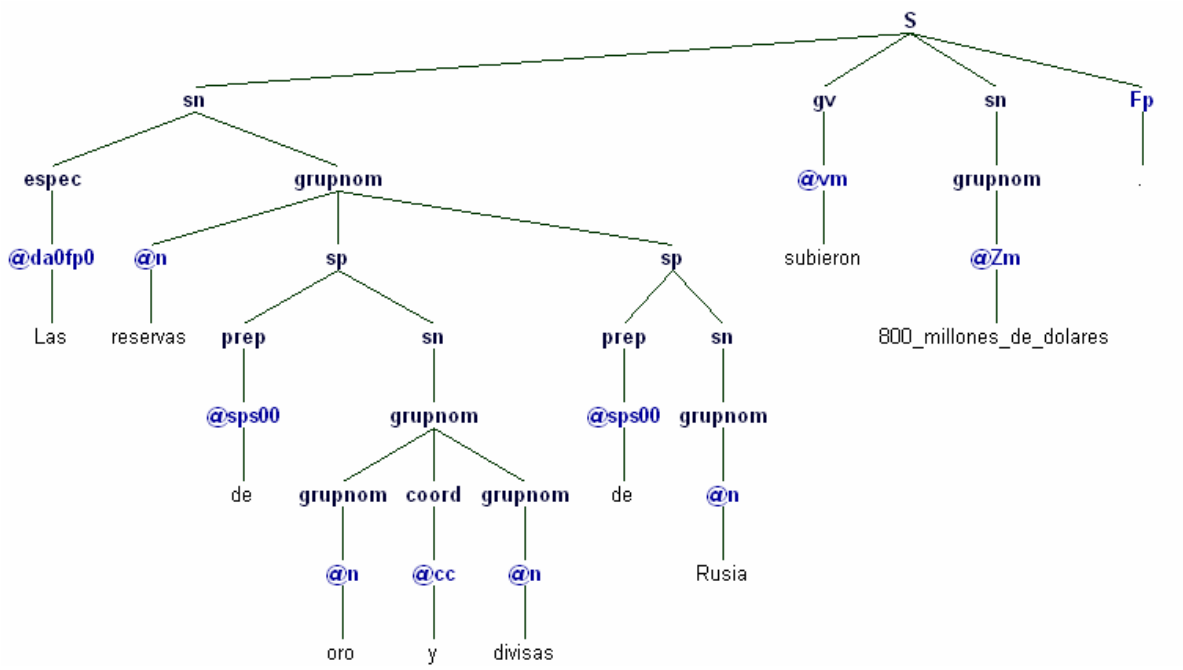


Figura 23. Árbol de constituyentes para la oración "Las reservas de oro y divisas de Rusia subieron 800 millones de dólares.

El algoritmo continúa su ejecución hasta que el nodo raíz sube. El árbol de dependencias resultante se muestra en la Figura 24 y Figura 17.

6.3.2.4 Resultados experimentales

El algoritmo encontró 2663 reglas gramaticales. De éstas, 339 (12%) se repiten más de 10 veces, y 2324 (88%) menos de 10 veces. Las veinte reglas más frecuentes, con su número respectivo de ocurrencias son:

12403 sn ← espec grupnom
11192 sp ← prep sn
3229 grupnom ← n sp
1879 grupnom ← n s
1054 sp ← prep S
968 grupnom ← n S
542 gv ← va vm
535 grupnom ← s n
515 S ← infinitiu sn
454 grupnom ← n s sp

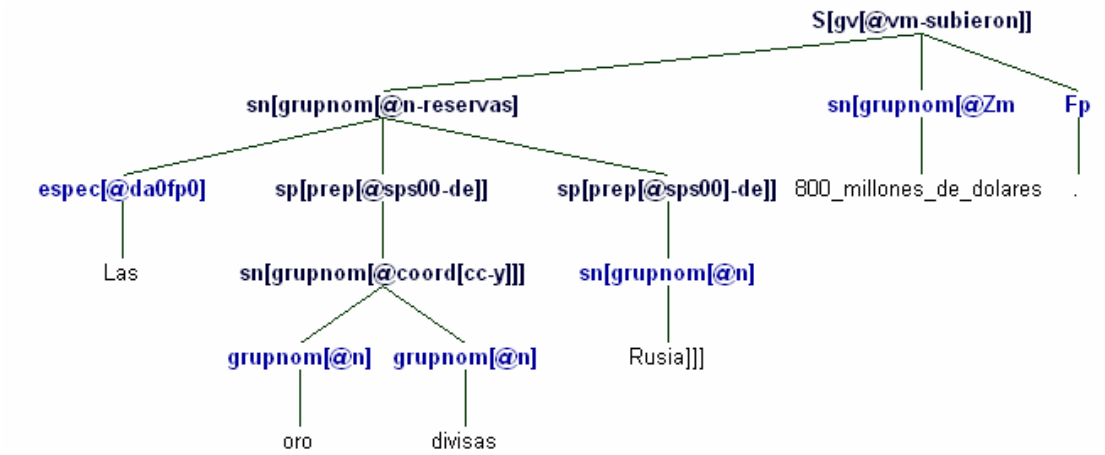


Figura 24. Árbol de dependencias resultante con etiquetas

392 grupnom ← n sn

390 grupnom ← grupnom coord grupnom

386 sn ← sn coord sn

368 grupnom ← s n sp

356 gv ← vm infinitiu

343 S ← S coord S Fp

315 S ← S coord S

276 sp ← prep sn Fc

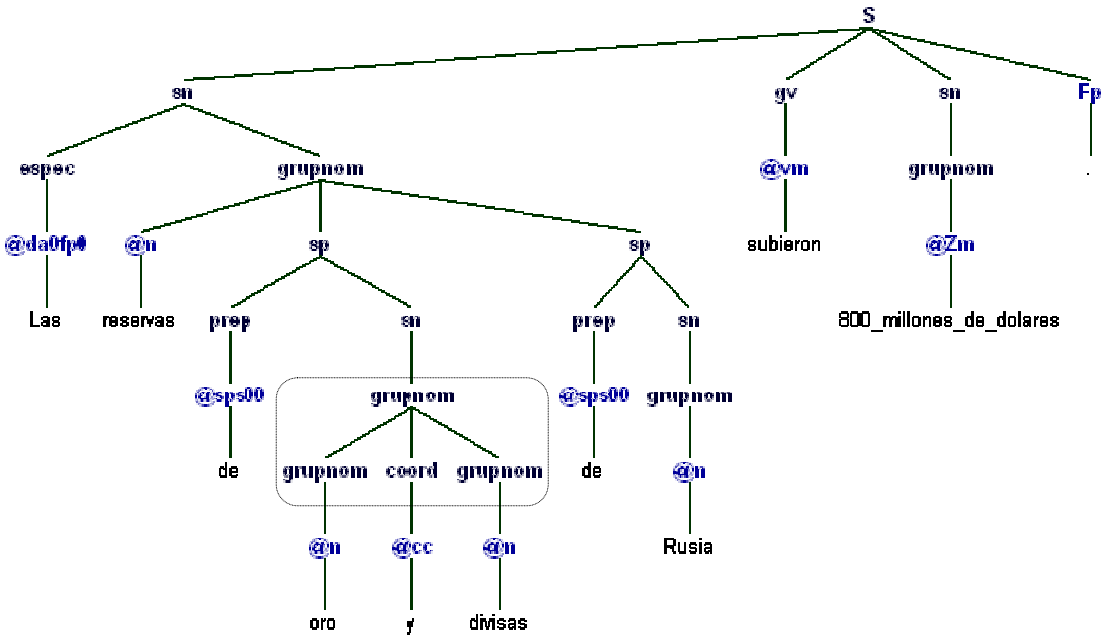


Figura 25. Árbol de constituyentes

270 grupnom ← n sp sp

268 S ← infinitiu sp

6.3.2.4.1 Identificación de los núcleos en las reglas.

Las heurísticas cubrieron, es decir, etiquetaron automáticamente 2210 (79.2%) de todas las reglas gramaticales extraídas. Seleccionamos aleatoriamente 300 de ellas y las marcamos manualmente. La comparación mostró que prácticamente todas (99.9%) exceptuando dos marcas coincidieron. La Figura 18 muestra las reglas que no coincidieron. Estas dos reglas no coincidieron porque las reglas heurísticas no consideran estos casos.

Considerando las estadísticas de comparación, creemos que al menos 95% de las reglas de 3LB marcadas automáticamente han sido marcadas correctamente.

6.3.2.5 Resultados

El algoritmo extrajo 65,997 tripletas de dependencias de todo el treebank 3LB.

Para evaluar, extrajimos aleatoriamente 35 oraciones a partir del treebank y las convertimos manualmente en árboles de dependencias, lo cual dio 399 tripletas de dependencias. Después aplicamos nuestro procedimiento a estas oraciones. Para una oración de n palabras debe haber $(n - 1)$ tripletas. Nuestro procedimiento extrajo 399 tripletas, de las cuales, 368 (92%) coinciden con aquellas manualmente identificadas. Extrapolando las estadísticas, inferimos que más del 90% (unas 60,000) de las tripletas de dependencias que extrajimos del treebank 3LB, son correctas

Hemos presentado una técnica supervisada simple que permite transformar automáticamente árboles de constituyentes en árboles de dependencias. Esta técnica usa ciertas heurísticas simples que dependen del conjunto de etiquetas usado en cierto treebank o gramática. Nuestra técnica no trata con fenómenos difíciles o discutibles de la sintaxis de dependencias, sin embargo, recuperan la masa de relaciones de dependencias. Dichos resultados, a pesar de no ser completamente profundos, son extensamente utilizables en la mayoría de las aplicaciones prácticas. Además, es posible reutilizar los analizadores sintácticos o treebanks existentes para las aplicaciones que requieren estructuras de dependencias.

6.3.3 Evaluación de nuestro sistema usando este esquema

De los analizadores sintácticos con una cobertura realista disponibles para el español, podemos mencionar el analizador de XEROX comercialmente disponible[24], Connexor Machinese Syntax[25] y el sistema gratuito TACAT[26]. Usamos los dos últimos sistemas para comparar su exactitud con la de nuestro sistema. Sólo el sistema de Connexor es realmente basado en dependencias. Está basado en el formalismo de Gramática Funcional basada en Dependencias (Functional Dependency Grammar) [181]. Los otros sistemas están basados en constituyentes.

En esta sección presentamos la comparación de nuestro sistema contra un estándar de referencia anotado a mano (gold standard). También comparamos nuestro analizador con dos analizadores ampliamente conocidos para el español. El primero es Connexor Machinese Syntax para el español (un analizador de dependencias) y TACAT (un analizador de constituyentes).

Seguimos el esquema de evaluación propuesto por Briscoe *et al.* [29], el cual propone evaluar la exactitud de los analizadores sintácticos basándose en las relaciones gramaticales entre núcleos léxicos lematizados. Este esquema es adecuado para evaluar tanto analizadores de dependencia como analizadores de constituyentes, porque considera las relaciones en un árbol que están presentes en ambos formalismos, por ejemplo [Det *coche el*] y [ObjetoDirecto *tirar lo*]. Para nuestros propósitos de evaluación, traducimos la salida de los tres analizadores y el estándar de referencia (gold standard) en una serie de tripletas incluyendo dos palabras y su relación. Después, las tripletas de los analizadores se comparan contra las tripletas del estándar de referencia para encontrar una correspondencia.

Hemos elegido al corpus Cast3LB como nuestro estándar de referencia porque es, hasta ahora, el único corpus etiquetado sintácticamente para el español, que está disponible ampliamente. Cast3LB es un corpus que consiste de 100,000 palabras (aproximadamente 3,700 oraciones) extraídas de dos corpus: el corpus CLiCTALP (75,000 palabras), un corpus balanceado que contiene temas literarios, periodísticos, científicos y otros; y el segundo corpus fue tomado de la agencia noticiosa EFE (25,000 palabras), que corresponden al año 2000. Este corpus se anotó siguiendo los

---

[24] que solía estar en www.xrce.xerox.com/research/mltt/demos/spanish.html, pero parece haber sido cambiado recientemente

[25] www.connexor.com/demo/syntax.

[26] www.lsi.upc.es/~nlp/freeling/demo.php.

Estándares de anotación morfosintáctica para el español [51] usando el enfoque de constituyentes, de tal manera que primero tuvimos que convertirlo en un corpus de dependencias. A continuación presentamos un resumen breve de este procedimiento. Para detalles vea la sección anterior 6.3.2.

1. Extraer patrones del treebank para formar reglas. Por ejemplo, un nodo llamado NP con dos hijos, Det y N, produce la regla NP → Det N.
2. Usar heurísticas para encontrar el componente núcleo de cada regla. Por ejemplo, un sustantivo siempre será el núcleo en una regla, excepto cuando un verbo esté presente. La cabeza se marca con el símbolo @: NP → Det @N.
3. Usar esta información para establecer la conexión entre núcleos de cada constituyente.
4. Extraer tripletas para cada relación de dependencia en el treebank de dependencias.

Como un ejemplo, considere la siguiente tabla. Muestra las tripletas para la oración tomada del corpus Cast3LB: *El más reciente caso de caridad burocratizada es el de los bosnios , niños y adultos.* En algunos casos los analizadores sintácticos extraen tripletas adicionales que no se encuentran en el estándar de referencia.

Tripletas	3LB	Connexor	DILUCT	TACAT
adulto DET el	x			
bosnio DET el	x	x	x	
caridad ADJ burocratizado	x		x	x
caso ADJ reciente	x		x	x
caso DET el	x		x	x
caso PREP de	x	x	x	x
de DET el	x			x
de SUST adulto	x			
de SUST bosnio	x		x	
de SUST caridad	x	x	x	x
de SUST niño	x			
niño DET el	x			
reciente ADV más	x			x
ser PREP de	x		x	x
ser SUST caso	x		x	x
recentar SUST caso		x		
caso ADJ más			x	
bosnio SUST niño			x	
ser SUST adulto			x	
de ,				x
, los				x
, bosnios				x

Extrajimos 190 oraciones aleatorias del treebank 3LB y las analizamos con Connexor y DILUCT. La precisión, recall y medida-F de los distintos analizadores contra Cast3LB son:

	Precision	**Recall**	**Medida-F**
Connexor	0.55	0.38	0.45
DILUCT	0.47	0.55	0.51
TACAT[27]	–	0.30	–

[27] Los resultados para TACAT fueron provistos amablemente por Jordi Atserias.

Note que el analizador Connexor, aunque tiene una precisión ligeramente mejor y una medida-F similar a la de nuestro sistema, no está disponible libremente, y por supuesto, no es de código abierto.

En el siguiente capítulo mostraremos algunas aplicaciones de nuestro sistema.

7 Algunas aplicaciones

7.1 *Desambiguación de sentidos de palabras* (WSD)

En este capítulo se presenta un método para extraer preferencias seleccionales vinculadas a ontologías, en particular Spanish-EuroWordNet. Esta información es utilizada en este trabajo, entre otras aplicaciones posibles, para realizar desambiguación de sentidos de palabras (WSD). La evaluación de este método se realiza usando el texto de entrenamiento de Senseval-2 en español. Los resultados de este experimento son ligeramente superiores a los obtenidos por Resnik usando preferencias seleccionales para el inglés, además de que el método propuesto no requiere ninguna anotación previa del texto (morfológica, sintáctica o semántica), a diferencia de métodos anteriores.

7.1.1 Introducción

Las preferencias seleccionales miden el grado de acoplamiento de un argumento (objetos directo, indirecto y complementos preposicionales) con respecto a un verbo. Por ejemplo para el verbo *beber*, los objetos directos agua, jugo, vodka y leche serán mucho más probables que pan, ideas, o hierba.

Para tener una cobertura adecuada de los posibles complementos de un verbo es necesario tener un corpus de entrenamiento muy grande. Sin embargo, aún para corpus muy grandes de cientos de millones de palabras, existen combinaciones de palabras que no ocurren dentro de este corpus, a pesar de ser combinaciones que son utilizadas de manera cotidiana.

Una de las soluciones para este problema es utilizar clases de palabras. En este caso, tanto agua, jugo, vodka y leche pertenecen a la categoría de *líquido* y por tanto se establece la asociación entre la clase *líquido* y el verbo *beber*. Sin embargo, no todos los verbos tienen una característica de asociación específica. Por ejemplo el verbo *tomar* puede tener argumentos de clases muy diversas: *tomar leche*, *tomar asiento*, o *tomar conciencia*.

Por otra parte, cada palabra puede tener más de una clasificación. Esto último depende no sólo de los sentidos de las palabras, sino de la característica principal que haya sido tomada en cuenta para colocar a la palabra dentro de una clase específica. Si consideramos por ejemplo la coloración de los objetos, colocaríamos *leche* dentro de la clase de los objetos blancos. Si consideramos las propiedades físicas, podemos decir que pertenece a la clase de *fluidos*, *líquidos*, o *antiácidos*; o bien

la leche puede ser un *alimento_básico*, etc. Es decir, la clasificación relevante para una palabra depende del uso que se le dé, y no sólo del sentido.

Para encontrar una correlación entre el uso de un sustantivo, su sentido, y las preferencias seleccionales de los verbos son necesarios los siguientes tipos de información: 1) Información ontológica de una palabra, de tal manera que una palabra no esté vinculada de manera plana a una sola clase y 2) Información del uso de la palabra, dado un verbo, vinculada a una posición específica dentro de la ontología.

En este artículo proponemos un método para extraer preferencias seleccionales vinculadas a una ontología. Esta información ayuda a la solución de múltiples problemas dentro del enfoque de métodos estadísticos combinados con conocimiento [156, 157]. Como un ejemplo, en la Tabla 22 se muestra un extracto de las ocurrencias de argumentos para tres verbos usando la jerarquía de WordNet. El objetivo de este trabajo es obtener una tabla similar, y posteriormente usar esta información para hacer desambiguación de sentidos de palabra (WSD).

7.1.2 Trabajos Relacionados

Uno de los primeros trabajos que trataron la extracción de preferencias seleccionales vinculada con sentidos de WordNet fue el de Resnik [158]. Este trabajo está dedicado principalmente a la desambiguación de sentidos de palabras en inglés. Resnik consideró que no existía texto anotado con sentidos de palabras.

Posteriormente otro trabajo que vincula el uso de verbos con sus argumentos es el de Agirre y Martínez [1,2]. A diferencia de Resnik, Agirre y Martínez partieron de la existencia de un texto anotado con sentidos de palabra: SemCor, en inglés.

Debido a que recursos como SemCor son difíciles de obtener para otros lenguajes como el español, y su construcción es costosa, en este trabajo seguimos la línea de investigación de Resnik, en el sentido de no requerir un texto anotado con sentidos de palabra.

Aún más, en este trabajo consideramos que el análisis debe ser completamente automático, por lo que tanto el texto para extracción de preferencias seleccionales, como el texto para hallar los sentidos de palabras, no tienen ninguna anotación manual previa. Todos los pasos son realizados de forma automática. A continuación describiremos en detalle los pasos para la extracción de preferencias seleccionales vinculadas a una ontología.

Tabla 22. Usos poco comunes y usos comunes de combinaciones de verbo + synset en WordNet

leer	traspaso	0.14	beber	superficie	0.20
leer	fauna	0.17	beber	vertebrado	0.20
leer	comida	0.20	beber	lectura	0.20
leer	mensaje	27.13	beber	sustancia	11.93
leer	escritura	28.03	beber	alcohol	12.50
leer	objeto_inanimado	29.52	beber	líquido	22.33
leer	texto	29.75			
leer	artículo	37.20	tomar	artrópodo	0.20
leer	libro	41.00	tomar	clase_alta	0.20
leer	comunicación	46.17	tomar	conformidad	0.20
leer	periódico	48.00	tomar	sustancia	39.27
leer	línea	51.50	tomar	postura	49.83
			tomar	resolución	89.50
			tomar	control	114.75
			tomar	acción	190.18

7.1.3 Metodología

Para construir la ontología con valores de uso para cada verbo, se utilizó EuroWordNet 1.0.7 en español (S-EWN)[28] y un corpus de 161 millones de palabras correspondiente a la publicación de 4 años de 3 periódicos mexicanos. Aproximadamente este corpus contiene 60 millones de enunciados. Se etiquetó el texto con partes gramaticales (POS) usando el etiquetador estocástico TnT entrenado con el corpus CLiC-TALP. Según la *Evaluación del etiquetador TnT para el español* [134], confirmado por pruebas adicionales realizadas por nosotros, este etiquetador tiene un desempeño superior al 94% para el español. Posteriormente se utilizaron reglas simples de agrupación (*chunks*) para adjetivo + sustantivo, adverbio + verbo, etc. También se delimitaron las frases subordinadas.

Una vez etiquetado el texto se extrajeron las combinaciones de verbo + sujeto a la izquierda, verbo + objeto a la derecha y verbo + preposición + sustantivo. Aquí el símbolo + indica adyacencia inmediata. Para evitar ruido innecesario se ignoraron patrones que incluían palabras intermedias. Por ejemplo, patrones como verbo + X + preposición + sustantivo, fueron ignorados. X aquí sustituye a cualquier palabra o conjunto de palabras.

[28] S-EWN fue desarrollado conjuntamente por la University of Barcelona (UB), la Universidad Nacional de Educación a Distancia (UNED), y la Universidad Politécnica de Cataluña (UPC), España

contar con permiso:

00629673n → **sanción** 58.96 → autorización 231.89 → management 83.01 → control_social 115.54 → acción 1808.92

04368291n → **aprobación** 232.57 → mensaje 570.55 → comunicación 1066.30 → relación_social 761.52 → relación 847.98 → abstracción 1734.82

08562692n → **libertad** 198.76 → libertad 198.76 → estado 640.99

leer > libro:

01712031n → **estómago** 51.30 → órgano_interno 34.20 → órgano 29.01 → parte_del_cuerpo 31.90 → trozo 53.86 → entidad 271.28

02174965n → **producto** 177.33 → creación 164.83 → artefacto 209.80 → objeto_inanimado 232.52 → entidad 271.28

04214018n → **sección** 93.82 → escritura 514.13 → lenguaje_escrito 377.39 → comunicación 831.37 → relación_social 645.04 → relación 628.82 → abstracción 587.39

04222100n → **publicación** 106.00 → obra 282.98 → producto 177.33 → creación 164.83 → artefacto 209.80 → objeto_inanimado 232.52 → entidad 271.28

04545280n → **obra_dramática** 74.49 → escritura 514.13 → lenguaje_escrito 377.39 → comunicación 831.37 → relación_social 645.04 → relación 628.82 → abstracción 587.39

Figura 26. Ontología con valores de uso para las combinaciones *contar con permiso* y *leer libro*

En la Figura 26 se muestra un extracto de los patrones de palabras obtenidos de la manera descrita anteriormente. El símbolo **>** significa que el sustantivo aparece a la derecha inmediata del verbo; el símbolo **<** significa que el sustantivo aparece a la izquierda inmediata del verbo; una preposición indica el formato verbo + preposición + sustantivo.

Posteriormente, el sustantivo de cada combinación se buscó en WordNet y se anotó una ocurrencia para él. En caso de que este sustantivo tuviera más de un sentido, se distribuyó esta ocurrencia entre cada uno de sus sentidos. Por ejemplo, si libro tiene 5 sentidos, se anotó 1/5 para cada uno de sus sentidos.

Con el objeto de aprovechar la estructura ontológica de WordNet, se propagó este valor hacia arriba en la jerarquía siguiendo los hiperónimos de cada uno de los sentidos. Se utilizó una estrategia de

1	contar	con	permiso
2	retirar	de	sistema
3	golpear	>	balón
4	solucionar	>	problema
5	dar	>	señal
6	haber	<	incógnita
7	poner	en	cacerola
8	beber	de	fuente
9	beber	>	vodka

Figura 28. Combinaciones extraídas del CVV

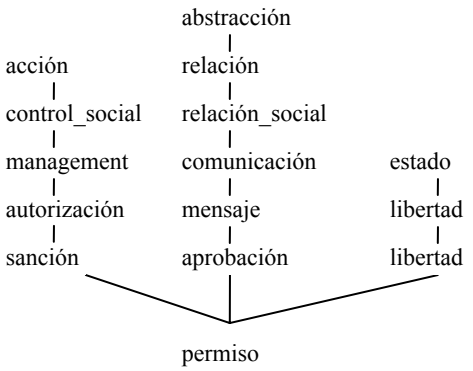


Figura 27. Estructura en S-EWN para *permiso*.

penalización por nivel de $occ = occ + \frac{1}{nivel}$, de tal manera que en el primer nivel se anota una ocurrencia de 1; en el segundo nivel, se anota una ocurrencia de 0.5; en el tercer nivel una ocurrencia de 0.33, etc. Por ejemplo, para el primer caso de la Figura 28, *permiso* tiene tres sentidos diferentes. El fragmento de S-EWN correspondiente a *permiso* aparece en la Figura 27. Cuando en el texto de entrenamiento aparecen la combinación *contar con permiso*, se anotará una ocurrencia de 1 para *permiso*, una de $\frac{1}{2} \cdot \frac{1}{3}$ para los sentidos de permiso correspondientes a *sanción*, *aprobación* y *libertad*; $\frac{1}{3} \cdot \frac{1}{3}$ para *autorización*, *mensaje* y *libertad*, etc.

Cada uno de estos valores se sumó para cada una de las ocurrencias en el corpus de entrenamiento. El resultado fue una ontología con usos ponderados (preferencias seleccionales) de los argumentos para cada verbo. En la Figura 26 aparecen los ejemplos para *contar con permiso* y para *leer > libro*. Los números que aparecen en dicha figura corresponden al synset correspondiente de S-EWN.

En la Figura 26 puede verse que la información obtenida puede usarse para elegir el sentido más probable de la palabra, dado el verbo con el que se está utilizando. Por ejemplo, para *leer*, el sentido menos probable es el de *estómago* (*órgano interno*), en tanto que el más probable es el de *producto* (*creación*), seguido de *publicación* (*obra*). En la siguiente sección describimos un experimento llevado a cabo para medir el desempeño de este método en la tarea de desambiguación de sentidos de palabras.

7.1.4 Resultados experimentales

Senseval es una serie de ejercicios de evaluación de desambiguación de sentidos de palabras organizado por la ACL-SIGLEX. La penúltima competición se llevó a cabo en 2001. Los datos de esta competición están disponibles en línea. Esta competición incluyó, entre 10 idiomas, al español. Sobre los archivos de esta competición aplicamos nuestro método. Una evaluación de resultados exactos mostró que 830 de 1111 casos fueron resueltos (una cobertura de 74.7%), y de los casos resueltos, 223 fueron resueltos adecuadamente (precisión de 45.3%). Este resultado es similar al obtenido por Resnik [158] para el idioma inglés: una media de 42.5% para las relaciones de verbo-sujeto y verbo-objeto, y es superior a la media lograda con desambiguación al azar (28% según Resnik [158]).

7.1.4.1 Discusión

Los resultados muestran un desempeño inferior al de otros sistemas de WSD como el presentado en *Un sistema de desambiguación de sentidos de palabra basado en máxima entropía* [180], el cual obtiene un *score* de 0.702 en la parte de sustantivos para el mismo conjunto de evaluación de Senseval-2; un punto importante a considerar es que esta medición ha sido obtenida tomando en cuenta grupos de sentidos, en tanto que en los resultados de nuestro método presentados anteriormente consideran resultados exactos. Si consideramos grupos de sentidos, la precisión de nuestro método es de 62.9%.

Por otra parte, la baja cobertura de nuestro método se debe a que sólo se están considerando relaciones verbo-sujeto y verbo-objeto. Algunas de las desambiguaciones de sentido dependen de otras relaciones como sustantivo-adjetivo, y otros modificadores. Por ejemplo en el texto de evaluación aparece la frase: *el Apocalipsis no tiene nada que ver con la guerra de las galaxias ni con la bomba atómica.* En este caso, el sentido de *bomba* está restringido principalmente por el adjetivo *atómica*, y no por el verbo central de la oración subordinada (*ver*). Determinar el sentido de *bomba* mediante la combinación *ver, con, bomba* no es la mejor estrategia para desambiguar el sentido de palabras que presentan este fenómeno.

Para mejorar el funcionamiento de nuestro método, es necesario incluir información de uso de combinaciones entre adjetivos y adverbios. Esta tarea forma parte del trabajo futuro presentado en la siguiente sección.

7.1.5 Resumen

En este artículo presentamos un método para extraer preferencias seleccionales vinculadas a ontologías. La información obtenida es útil para resolver diversas tareas que requieren de información acerca del uso de las palabras dado cierto verbo en una oración. En particular presentamos un experimento que aplica este método para desambiguación de sentidos de palabras. Los resultados de este experimento se encuentran aún lejos de aquellos obtenidos mediante otros métodos, sin embargo hemos partido de no requerir ninguna clase de anotación morfológica, de partes gramaticales, o semántica del texto. Además, hemos identificado los puntos específicos para mejorar el funcionamiento de este método bajo la misma línea de métodos estadísticos combinados con métodos de conocimiento.

Por otra parte, como trabajo futuro, la información obtenida es útil no sólo para resolver problemas de WSD, sino también para otros problemas importantes como desambiguación sintáctica. Por ejemplo, considere la frase *pintó un cuadro un pintor*. Existe ambigüedad entre cuál sustantivo es el

sujeto y cuál es el objeto, pues el español permite un orden casi libre de constituyentes. La frase *pintó un pintor un cuadro* tiene exactamente el mismo significado. Para determinar el sujeto más probable, se puede consultar la ontología con preferencias seleccionales construida con el método presentado en este artículo, considerando que la mayor parte de las veces el sujeto aparece del lado izquierdo. Monedero *et al*. [131] reportan que esto ocurre en un 72.6% de las oraciones.

La búsqueda de *un pintor pintó* regresa la siguiente cadena de hiperónimos valuados: `pintor→artista 1.00→creador 0.67→ ser_humano 2.48→causa 1.98`, en tanto que la búsqueda de *un cuadro pintó* regresa: `escena→situación 0.42→estado 0.34`. Es decir, pintor es más probable como sujeto para la oración mencionada anteriormente. Una implementación a mayor escala de este método queda como trabajo futuro.

7.2 Esteganografía lingüística

7.2.1 Introducción

Existen muchas formas de ocultar la información. Las más conocidas son las técnicas de criptografía que desordenan un mensaje de tal manera que la información original no pueda ser comprendida. Pero enviar un mensaje encriptado puede llamar la atención: ¿qué tipo de información puede haber en dicho mensaje, que no puede ser vista por alguien más? Entonces es deseable que el mensaje oculto pase desapercibido. Esto se ha hecho desde épocas muy remotas: los griegos tatuaban mensajes cortos en la cabeza afeitada de los mensajeros, y esperaban a que el pelo les creciera para enviarlos.

La escritura oculta es llamada *esteganografía*[29], palabra compuesta por las palabras griegas *stèganos* (oculto) y *gràfein* (escribir). Ejemplos menos antiguos de estas técnicas incluyen micropuntos en cartas que parecían ordinarias, ocultando textos espías durante la segunda guerra mundial. Usted incluso puede haber usado la *esteganografía* si jugó alguna vez con la tinta invisible hecha de limón.

Hoy en día, la palabra *esteganografía*, en un contexto informático, hace referencia a ocultar un mensaje dentro de otro tipo de información. Por ejemplo, incluir un texto en una imagen o en un archivo de sonido que puede viajar por la red libremente sin despertar sospechas. Estas técnicas

[29] No confundir con *estenografía*, métodos abreviados de escritura como la taquigrafía.

están ligadas estrechamente a los archivos digitales, que pueden contener información adicional a la que puede verse o escucharse a simple vista.

En los últimos años se han hecho investigaciones para ocultar información dentro de textos sin que sea notorio. Peter Wayner creó en 1997 un sistema para codificar un mensaje como la narración de un juego de béisbol[30] hipotético entre dos equipos. Una variación de este sistema es la codificación de mensajes ocultos que aparentan ser un mensaje de correo basura como los que circulan diariamente por Internet. Pruebe decodificar este mensaje en `http://www.spammimic.com`.

Este texto se encuentra en `http://likufanele.com/stegano` para poderlo copiar y pegar:

```
Dear Professional , This letter was specially selected to be sent to you . If you
no longer wish to receive our publications simply reply with a Subject: of
"REMOVE" and you will immediately be removed from our mailing list . This mail is
being sent in compliance with Senate bill 2416 ; Title 3 ; Section 301 . THIS IS
NOT MULTI-LEVEL MARKETING ! Why work for somebody else when you can become rich
as few as 47 weeks . Have you ever noticed nobody is getting any younger & most
everyone has a cellphone ! Well, now is your chance to capitalize on this ! We
will help you process your orders within seconds and process your orders within
seconds . You are guaranteed to succeed because we take all the risk ! But don't
believe us ! Mr Ames of Florida tried us and says "Now I'm rich many more things
are possible" ! This offer is 100% legal ! We BESEECH you - act now ! Sign up a
friend and you'll get a discount of 50% . Thanks .
```

Estos sistemas de codificación se basan en detalles de los textos como la puntuación, los espacios, los números, y la sustitución de palabras y frases que no tienen relevancia para el texto. Por ejemplo *Dear Professional* puede ser sustituido por *Dear professional* (con minúscula), o *Dear Friend* sin importar demasiado en este texto. Estas variaciones permiten codificar, por ejemplo, una *a* en el primer caso, una *b* en el segundo, etc. Sin embargo, cualquiera que lea con atención el mensaje del párrafo anterior podrá darse cuenta de que hay algo extraño en él: la falta de coherencia y la puntuación. Y si alguien nota algo extraño, entonces el mensaje oculto ya no pasa desapercibido. Además, al depender de la puntuación y otros detalles como mayúsculas y minúsculas, este mensaje no puede ser transmitido por otros medios que no sean digitales como el teléfono, la radio, una carta de correo terrestre, etc.

Para ocultar un mensaje en texto de forma automática, el resultado debe verse lo más natural posible. Es aquí donde entra la *esteganografía lingüística*, pues el conocimiento lingüístico es

[30] http://www.wayner.org/texts/mimic/

necesario para crear textos con cierta coherencia. Para codificar un mensaje dentro de un texto usando *esteganografía lingüística*, generalmente se sustituyen las palabras por otras equivalentes. Por ejemplo, un fragmento de Caperucita Roja puede escribirse de dos formas muy similares:

{Había / Érase} una vez una niña muy bonita a la que su {madre / mamá} le había {hecho / confeccionado} una capa roja. La {muchachita / jovencita} la usaba siempre, por eso todo el mundo la {llamaba / nombraba} Caperucita Roja. Un día, la {madre / mamá} le {pidió / solicitó} que llevase unos pastelitos a su abuela que {vivía / habitaba} al otro lado del bosque. Caperucita Roja {puso / colocó} los pastelitos en la {cesta / canasta} y echó a {andar / caminar} por el {camino / sendero} para {ir / acudir} a {casa / domicilio} de su abuelita. La niña no tenía miedo porque allí siempre se {encontraba / topaba} con muchos amigos. Mientras tanto, el lobo {llamó / tocó} suavemente a la puerta...

Este fragmento puede tener hasta 16 cambios, mostrados en corchetes. Cada uno de ellos puede usarse para representar un *bit* de información (por ejemplo *madre* = 0, *mamá* = 1), por lo que en este texto podemos esconder 16 bits. En una computadora cada letra se representa con 8 bits, por ejemplo a la A le corresponden los bits 100000001, a la B 10000010, a la C 1000011,... hasta llegar a la Z, que es 1011010. Si consideramos que cada letra en las computadoras se representa con 8 bits (1 byte), en el fragmento de Caperucita podemos esconder dos letras. Por supuesto, previamente el receptor debe tener un diccionario que le indique que *madre* = 0 y *mamá* = 1; de lo contrario no podrá decodificar el mensaje.

Elegir el sinónimo adecuado para sustituir una palabra no es una tarea trivial. Por ejemplo, podemos decir *echó a caminar* o *echó a andar*, pero no podemos decir *echó a marchar*. Otro ejemplo lo encontramos en *llamar a la puerta*: Nombrar, invocar, denominar son sinónimos de *llamar*, pero no podemos usar ninguno de ellos con el mismo sentido (*nombrar a la puerta*, *denominar a la puerta*), aunque sí podemos usarlo en *todo el mundo la* ***nombraba*** *Caperucita*. Otra complicación que aparece es el uso que pueden tener los verbos. Observe cómo *Érase* se cambia por *Había*, y no por *Habíase*, que es la forma que le corresponde en idéntica estructura.

7.2.2 Aplicación

La esteganografía lingüística permite ocultar información en un texto. El texto resultante debe ser gramáticamente correcto y semánticamente coherente para no ser sospechoso. Entre muchos métodos de esteganografía lingüística, nos adherimos a los enfoques previos que usan paráfrasis por sinónimos, es decir, sustituir las palabras de contenido por sus equivalentes. El contexto debe ser considerado para evitar posibles sustituciones que rompen la coherencia, como (*tiempo independiente* por *tiempo libre*). Basamos nuestro método en trabajos previos de esteganografía lingüística que usan colocaciones para verificar el contexto. Proponemos usar preferencias de selección en lugar de colocaciones porque las preferencias de selección pueden ser recolectadas automáticamente de una manera confiable, permitiendo que nuestro método se aplique para cualquier lenguaje.

Este trabajo se basa en trabajos previos [14, 16] que usan una base de datos recolectada manualmente. Recolectar colocaciones manualmente es una tarea que puede llevar muchos años. Por ejemplo, para completar una base de datos de colocaciones rusas [15] se ha trabajado en ella por más de 14 años. Por otra parte, usar la Internet para verificar colocaciones [19] no es adecuado para colocaciones divididas, como *cometer un* horrible *error*, porque los motores de búsqueda actuales no permiten ajustar el alcance de dichas búsquedas. Por ejemplo el uso de la búsqueda con el operador NEAR no permite restringir el resultado a la misma oración.

7.2.2.1 Algunas definiciones

La esteganografía lingüística es un conjunto de métodos y técnicas que permiten ocultar información en un texto basándose en conocimiento lingüístico. Para ser efectivo, el texto resultante debe tener corrección gramatical y cohesión semántica.

Hay dos enfoques principales para lograr esto: 1) generar texto y 2) cambiar texto previamente escrito. Para ilustrar el primer enfoque, imagine un modelo de generador de oraciones que usa patrones de verbo-preposición-sustantivo. Este modelo podría generar oraciones válidas como *ir a la cama*, *cantar una canción*, etc. Un problema no trivial surge cuando se trata generar texto coherente usando estas oraciones: *Juan va a la cama, y luego canta una canción*. Textos no coherentes de este tipo no están libres de sospecha. Como señalan Chapman *et al*. [46], lo mismo pasa cuando se quiere usar modelos oracionales extraídos de texto previamente escrito.

En el segundo enfoque, para ocultar información algunas palabras en el texto fuente se reemplazan por otras palabras dependiendo de la secuencia de bits a ser escondida. Estos cambios son

detectables sólo en el lado del receptor deseado. En los mejores casos, el texto resultante mantiene el significado del texto original.

Al igual que en los trabajos de Igor Bolshakov [14, 16, 19], nos adherimos al segundo enfoque porque es bastante más realista; de hecho, generar texto desde cero necesita no sólo información sintáctica o semántica, sino también información pragmática.

En este trabajo no consideramos otros métodos de esteganografía textual como formateo de textos, variación de espacios entre palabras, u otros métodos de codificación no lingüísticos. Esto es porque los métodos lingüísticos genuinos permiten que un mensaje se transmita independientemente del medio: la esteganografía lingüística permite transmitir un mensaje por Internet, por teléfono, por transmisión de radio, etc.

En resumen, continuamos el desarrollo del método de esteganografía que reemplaza palabras textuales por sus sinónimos [14]. Este trabajo permite mantener la información escondida en textos no sospechosos a la vez que se mantiene la corrección lingüística y el significado del texto original. En adición, tomamos ventaja de los recursos existentes para extender la cobertura de este método a virtualmente cualquier lenguaje, siempre que existan los recursos requeridos para dicho lenguaje. Esto se logra considerando una fuente alternativa automática para el contexto de las palabras.

7.2.2.2 El contexto de una palabra

El contexto de una palabra dada son las palabras que la rodean dentro de una oración. En muchos artículos se considera como contexto únicamente a las palabras que aparecen junto a la palabra. Otros autores consideran a las colocaciones como *una secuencia de dos o más palabras* ***consecutivas*** *que tienen características de una unidad sintáctica y semántica, y cuyo significado exacto y no ambiguo, o bien su connotación, no pueden ser derivadas directamente del significado o connotación de sus componentes*, según define Choueka [49]. Ejemplos que caen dentro de esta definición, incluyendo modismos, son: *hot dog*, *vino blanco* (en realidad el vino blanco es amarillo), *estirar la pata*, y *enfermedad grave*.

Adicionalmente, el uso actual del término *colocación* incluye también combinaciones de palabras que conservan su sentido original como *café fuerte*, pero se consideran colocaciones porque sustituir cualquiera de sus componentes por palabras equivalentes sustituye una combinación entendible pero que suena extraño, como *café poderoso*, *lluvia pesada* (en lugar de *lluvia fuerte*), o *hacer un error* (en lugar de *cometer un error*). Adicionalmente, las colocaciones no son necesariamente palabras adyacentes, como en *cometer un* horrible *error* y pueden involucrar cuestiones de subcategorización, como ilustraremos más adelante.

Los enlaces entre los componentes de las colocaciones son sintagmáticos. Estos son, por ejemplo, el vínculo entre un verbo y un sustantivo que llenan su valencia (*hecho → de piedra*), o el vínculo entre un sustantivo y su adjetivo modificativo (*piedra → negra*). Este tipo de relaciones puede ser claramente visto en una representación de dependencias. El **contexto para una palabra** está dado por sus relaciones de dependencias. Por ejemplo, vea la figura **Figura 29** para la oración *Mary nos leyó un cuento de hadas*.

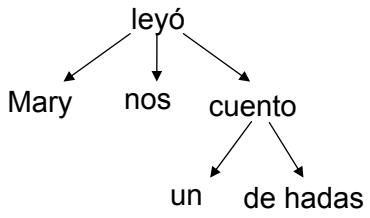


Figura 29. Representación de dependencias simplificada para la oración *Mary nos leyó un cuento de hadas*.

Para ilustrar la influencia del contexto de una palabra en una oración, sustituiremos algunas palabras por sus equivalentes (sinónimos). Por ejemplo, entre los sinónimos de *hada* están *ninfa, hechicera*. Sustituir hada por estos equivalentes produce, sin embargo, una oración que suena extraña: *Mary nos leyó un cuento de ninfas*, o *Mary nos leyó un cuento de hechiceras*, que son posibles, pero suenan extrañas, puesto que *hadas* depende fuertemente de *cuentos*. Otro ejemplo es sustituir *cuento* por *patraña,* o *aventura* (olvidándonos por un momento de *hadas*). En este caso suena extraño *nos leyó una patraña*, y sonaría mucho más natural *nos contó patrañas*, ¡sin tomar en cuenta *nos contó una patraña de hadas*! Esto muestra la fuerza entre el verbo (*leer, contar*) y uno de sus argumentos (*cuento y patraña*, respectivamente).

La subcategorización juega un rol importante cuando se consideran las colocaciones. Por ejemplo, considere los sinónimos de *contar*: *relatar, platicar, decir*. Si uno quisiera cambiar *leer* por alguno de estos sinónimos, deben considerarse el contexto y la estructura para mantener el mismo significado de la oración, manteniéndola *natural*. Simplemente cambiar *leyó* por *dijo* daría **Mary nos dijo un cuento de hadas*. Para que esta oración fuera natural, se necesitaría usar una estructura distinta (marco de subcategorización): *Mary nos dijo de un cuento de hadas*.

En contraste con este último ejemplo, en este trabajo nos enfocamos sólo en sinónimos que mantienen la estructura y el orden de palabras de una oración, así como el número de palabras (contando multipalabras estables como *hot dog* como una sola unidad). Usamos combinaciones de

palabras para verificar que los resultados de la paráfrasis con sinónimos son naturales y producen un texto coherente y natural.

7.2.2.3 Verificación de las combinaciones de palabras

Nuestro objetivo es hacer paráfrasis usando sinónimos considerando el contexto. En trabajos previos de Bolshakov *et al*. [14, 16] se usó una base de datos con colocaciones previamente recopiladas manualmente. Actualmente sólo existen unas cuantas bases de datos electrónicas de colocaciones. Hasta donde sabemos, las bases de datos de colocaciones disponibles públicamente no existían hasta 1997, cuando surgió el *Advanced Reader's Collocation Searcher* (ARCS—Buscador de Colocaciones para el Lector Avanzado) para el inglés English [13], pero es inferior al ahora disponible Diccionario de Colocaciones Oxford [141] en todos los aspectos.

Uno de los pocos proyectos de la década actual enfocado a desarrollar una base de datos de colocaciones muy grande que está disponible para uso local está dedicado al ruso y produjo un sistema interactivo llamado CrossLexica [15, 17, 18]. Su parte central es una base de datos grande de colocaciones rusas, pero contiene también algo como una WordNet en Ruso. Particularmente, la parte similar a WordNet contiene un diccionario de sinonimia y una jerarquía de hipónimos/hiperónimos.

Una base de datos de colocaciones no puede listar todas las combinaciones de palabras, particularmente combinaciones libres de palabras como *chico* (*muchacho*) *grande*, *caminar en la calle*, etc. Hay diversos métodos para extraer colocaciones automáticamente (Vea *Fundamentos de procesamiento estadístico de lenguaje natural* [120] *y Una evaluación comparativa de técnicas de extracción de colocaciones* [144]). Sin embargo, la calidad de estas colocaciones obtenidas automáticamente no es tan buena como la de aquellas obtenidas a mano. Adicionalmente, como mostramos en la sección 7.2.2.2, el contexto de una palabra está fuertemente relacionado a la estructura de la oración. Por tanto, necesitamos conocimiento lingüístico en adición a métodos puramente estadísticos.

Aparte de ello, las palabras polisémicas tienen diversos sinónimos que no pueden sustituir a la palabra original sin cambiar el significado de un texto, porque son sinónimos de otros sentidos de la palabra. Por ejemplo, *planta* puede ser sustituida por *vegetal* o por *fábrica*, dependiendo ampliamente del contexto.

Hasta ahora hemos identificado lo siguientes requerimientos para la determinación automática de las posibles combinaciones de palabras: un corpus del cual aprender, conocimiento semántico, y la

estructura de la oración. El conocimiento lingüístico que cubre la semántica y permite determinar la estructura de la oración es un conjunto de **preferencias de selección**.

7.2.2.4 Preferencias de selección para paráfrasis sinonímica

Las preferencias de selección miden el grado en el cual un verbo *prefiere* un argumento: un sujeto, un objeto o un modificador circunstancial. El principio de preferencias de selección puede aplicarse también a relaciones adjetivo–sustantivo, verbo–adverbio y sintagmas preposicionales, produciendo una base de datos de *preferencias* que pueden verse como colocaciones con una medición de grados con la ayuda de generalizaciones semánticas. Por ejemplo, si *cosechar plantas* aparece en el corpus de entrenamiento, y sabemos que *cosechar* prefiere argumentos del tipo *flora*, entonces podemos restringir los sinónimos de *planta* a aquellos relacionados con *flora*, excluyendo aquellos relacionados con el proceso de fabricación.

Asimismo, las preferencias seleccionales pueden ser una ayuda para determinar la estructura de las oraciones. Por ejemplo, la estructura sintáctica de *Veo un gato con un telescopio* se desambigua considerando que *ver con {instrumento}* es más frecuente que *gato con un {instrumento}*. Calvo y Gelbukh presentan en *Obtención de preferencias de selección a partir de texto no etiquetado para desambiguación de unión de frase preposicional* [37] un método para desambiguación de unión de sintagma preposicional, y muestran en *Aprendizaje no supervisado de preferencias de selección vinculadas a una ontología* [35] cómo puede usarse esta información para restringir el sentido de una palabra.

Para este trabajo, usamos una base de datos de preferencias de selección basada en un corpus de cuatro años de periódicos mexicanos con 161 millones de palabras.

7.2.2.5 El algoritmo

El algoritmo esteganográfico propuesto tiene dos entradas:

- La información a ocultar, en forma de una secuencia de bits

- El texto fuente en lenguaje natural de longitud mínima, evaluado como aproximadamente 500 veces más grande que la información a ocultar. El formato del texto puede ser arbitrario, pero el texto debe ser ortográficamente correcto, para disminuir la probabilidad de correcciones no intencionadas durante la transmisión. Las correcciones pueden cambiar el número de palabras sinónimo en el texto o las condiciones para su verificación, y pueden, por tanto, desincronizar la esteganografía vs. el esteganálisis. El texto no debe ser semánticamente específico, es decir, no debe ser meramente una lista de nombres o una secuencia de números. En este sentido, los

artículos de noticias o artículos políticos son bastante aceptables. Cualesquier fragmentos de tipo inapropiado incrementan la longitud total requerida para uso esteganográfico.

Los pasos de este algoritmo son:

A1. Etiquetado y lematizado. El texto se etiqueta usando el etiquetador TnT entrenado con el corpus en Español LEXESP [169]. Se ha reportado que tiene una exactitud de 94% para el español. Posteriormente el texto es lematizado probando diversas variantes contra un diccionario [111].

A2. Identificación de combinaciones de palabras que pueden ser parafraseadas. Se extraen los siguientes patrones para cada oración; las oraciones subordinadas se tratan como oraciones separadas, de tal manera que hay sólo un verbo por oración.

i) sustantivo+verbo

ii) verbo+sustantivo

iii) sustantivo+preposición+sustantivo

iv) verbo, preposición+sustantivo

El símbolo '+' denota adyacencia, en tanto que una coma denota *cerca de* en la misma oración. Todas las demás palabras (adjetivos, artículos, etc.) se descartan.

Para los patrones iii e iv, es posible que surja una ambigüedad cuando cierto sustantivo podría ser unido al sustantivo previo o al verbo principal de la oración. Por ejemplo, en *Como arroz con palillos*, el sustantivo *palillos* podría estar unido a *arroz* o a *comer*. Esta ambigüedad se resuelve considerando la fuerza de las preferencias de selección de ambas posibilidades. Sólo la combinación más fuerte se considera [37]. En el ejemplo previo, *comer con palillos* (patrón iv) es más fuerte que *arroz con palillos*. Compare esto con *Como arroz con frijoles*, donde *arroz con frijoles* (patrón iii) es más fuerte que *comer con frijoles.*

A3. Evaluación de sinónimos mediante preferencias de selección. Se generan sinónimos para cada palabra, exceptuando las preposiciones, en el patrón. Posteriormente, se prueban diferentes combinaciones contra una base de datos recopilada previamente. Los detalles para extraer esta base de datos se describen en la sección 7.2.2.4. Esta base de datos da un *score* para una combinación dada. Este *score* se calcula usando una fórmula de información mutua: $freq(w1,w2)$ / [$freq(w1)$ +$freq(w2)$ + $freq(w1,w2)$]. Diversas fórmulas para calcular la información mutua se mencionan en *Fundamentos de procesamiento estadístico de lenguaje natural* [120]. Si el *score* de una combinación es más grande que un umbral, la combinación se lista como una sustitución posible.

Algunos patrones podrán tener más de una sustitución posible. Cada uno de ellos se lista en un orden particular, por ejemplo, comenzando de uno con valor más alto en la base de datos de preferencias de selección, a aquél más cercano al umbral. La construcción original también se califica, usando la misma base de datos de preferencias de selección.

A4. Cifrado. Cada bit de información a ser codificado decide qué paráfrasis sinonímica será hecha. Puesto que para algunos patrones hay diversas opciones para sustituir, cada paráfrasis puede representar más de un bit. Por ejemplo, dadas cuatro sustituciones posibles, es posible representar cuatro combinaciones de dos bits: 00, 01, 10 y 11.

A5. Concordancia. Si hay alguna sustitución que requiere cambios simples en la estructura sintáctica, éstos se realizan en esta etapa. Por ejemplo, en español *historia* puede ser sustituido por *cuento*, pero *historia* es femenino y *cuento* es masculino. Así que es necesario cambiar el artículo *la* por *el*, resultando *el cuento* y evitando **la cuento*.

Del lado del receptor, es necesario decodificar la información oculta. Esta es la tarea de un decodificador-esteganalizador específico. Posee los mismos recursos que el codificador: la base de datos de preferencias de selección y el módulo de etiquetado. El texto se etiqueta como en A1; los patrones se extraen como en A2. Las paráfrasis sinonímicas se ordenan como en A3; los bits se extraen mapeando cada combinación posible en la misma forma que en A3 y A4. La concordancia no representa un problema cuando se decodifica porque los artículos y otras palabras se descartan como en A1.

{Atrincherados / Resguardados / Guarecidos} en el Madison Square Garden para {asegurarse / consolidarse} contra amenazas "terroristas" y de manifestantes, los republicanos {iniciaron / comenzaron / emprendieron / *originaron} su festejo con auto elogios de cómo {encararon / enfrentaron / *desafiaron / *retaron} el 11 de septiembre. De hecho, cuando se {seleccionó / eligió} a NY para {celebrar / hacer / realizar / festejar} la convención, la {idea / *concepto / proyecto / *creencia} era regresar bajo la {sombra / *silueta / *opacidad} de las Torres Gemelas con G. W. Bush como comandante en {jefe / *líder / *patrón} en Irak, Afganistán y {encabezando / *iniciando / *empezando / conduciendo} la gran {lucha / *torneo / *riña / rivalidad} del {bien / *patrimonio / *fortuna / *sí} contra los "ejes del mal". Pero la realidad ha obligado a cambiar el {tono / *fuerza / aire} del programa...

Figura 30. **Texto con sinónimos para paráfrasis. Las sustituciones malas se marcan con ***

7.2.2.6 Un ejemplo examinado manualmente

Para ilustrar el algoritmo presentado en la sección anterior, aplicaremos nuestro método para esconder una pequeña cantidad de información en un fragmento de texto en español extraído de un periódico local[31]—vea la Figura 30.

Para este ejemplo hemos listado diversos sinónimos posibles para las palabras según las enlista un diccionario [111]. No todas las sustituciones son verificables, puesto que nuestra base de datos de preferencias de selección no contiene todas las instancias posibles. Ese es el caso de combinaciones incluyendo *Madison Square Garden*, por ejemplo. Las combinaciones siguientes: *asegurar(se) contra amenazas* vs. *consolidar(se) contra amenazas* pueden verificarse en nuestra base de datos de preferencias seleccionales: la primera arroja un *score* de 3 contra la segunda, que tiene un *score* de 0.2. Si colocamos nuestro umbral alrededor de 0.5, la segunda opción será descartada.

[31] La Jornada, Mexico, Agosto de 2004.

Tabla 23. **Combinaciones verificadas y su *score* (s) para el ejemplo de la Figura 30.**

word combination	s	word combination	s	word combination	s
atrincherar en Madison	?	creencia ser	0.31	conducir torneo	0.03
resguardar en Madison	?	sombra de torre	?	encabezar riña	0
guarecer en Madison	?	silueta de torre	?	empezar riña	0
asegurar contra amenaza	**3**	opacidad de torre	?	conducir riña	0
consolidar contra amenaza	0.2	**comandante en jefe**	**6.7**	lucha de bien	0.7
iniciar festejo	**0.7**	comandante en líder	0.45	lucha de patrimonio	0
comenzar festejo	**0.8**	comandante en patrón	0.4	lucha de fortuna	0
emprender festejo	0.4	jefe en Irak	?	lucha de sí	0
originar festejo	0	líder en Irak	?	torneo de bien	0
encarar 11	?	patrón en Irak	?	torneo de fortuna	0
enfrentar 11	?	**encabezar lucha**	**2.1**	torneo de sí	0
desafiar 11	?	**iniciar lucha**	**1.75**	riña de bien	0
retar 11	?	**conducir lucha**	**0.8**	riña de patrimonio	0
seleccionar a NY	**1.3**	**encabezar rivalidad**	**0.67**	riña de fortuna	0
elegir a NY	**1.2**	iniciar torneo	0.47	riña de sí	0
celebrar convención	**1.9**	empezar lucha	0.4	rivalidad de bien	0
hacer convención	**1.8**	empezar torneo	0.3	rivalidad de patrimonio	0
realizar convención	**1.6**	encabezar torneo	0.28	rivalidad de fortuna	0
festejar convención	**0.5**	iniciar rivalidad	0.08	rivalidad de sí	0
idea ser	**0.7**	iniciar riña	0.05	**aire de programa**	**0.66**
proyecto ser	**0.6**	empezar rivalidad	0.05	**tono de programa**	**0.54**
concepto ser	0.4	conducir rivalidad	0.05	fuerza de programa	0.23

La Tabla 23 muestra combinaciones adicionales de sustantivos verificados por la base de datos de preferencias de selección. Las entradas que no se encuentran en la base de datos de preferencias de selección se marcan con '?'

En la Tabla 23, las combinaciones que se encuentran arriba del umbral (0.5) se muestran en negritas. Las posibilidades alternativas que permiten la representación de un bit, se marcan en gris claro; aquellas que pueden representar dos bits se marcan en gris oscuro.

Este fragmento puede esconder 8 bits (1 byte) de información. El texto tiene alrededor de 500 bytes, de aquí que la razón que mide el ancho de banda esteganográfico es aproximadamente igual a 0.002. Esto significa que el texto debe ser 500 veces más grande que la información a ocultar.

7.2.2.7 Conclusiones

Como en su versión previa, el método propuesto de esteganografía lingüística conserva el significado del texto portador, así como su ausencia de sospecha. Una de las ventajas principales de nuestro método es que no requiere una base de datos de colocaciones recopilada manualmente. En

lugar de ello, se usan preferencias de selección extraídas automáticamente para extraer una base de datos grande. Puesto que el método presentado en este capítulo se basa en recursos automáticamente obtenidos, es posible extender su aplicación a muchos lenguajes. Los resultados son menos refinados que cuando se usan colocaciones recopiladas manualmente, pero son aceptables.

Por otra parte, el valor medio de 0.002 de ancho de banda logrado con paráfrasis local sinonímica podría parecer bajo. Un ejemplo de la entropía máxima de paráfrasis sinonímica que puede ser alcanzado, se encuentra en *Un método de esteganografía lingüística basado en sinonimia verificada colocacionalmente* [14]. En este trabajo se discute que partiendo de las muestras de paráfrasis sinonímica por I. Mel'cuk, el ancho de banda máximo del método de paráfrasis en esteganografía puede alcanzar aproximadamente 0.016. Esto se logra considerando paráfrasis sinonímica para frases completas, como *ayudar* y *dar ayuda*. Compilar una lista de frases que pueden ser sustituidas es una tarea actualmente en desarrollo temprano. Nuestro método alcanza 12.5% del nivel máximo posible, sin considerar las variantes de adjetivos. El valor alcanzable de ancho de banda de paráfrasis sinonímica depende de la saturación de los recursos lingüísticos. Es por esto que éstos deben ser desarrollados fuertemente, aunque no existe un límite claro de perfección. En particular, el ancho de banda logrado con nuestro método puede ser mejorado considerando variantes de adjetivos. Esto es parte del trabajo futuro.

Con respecto a nuestro algoritmo, difícilmente podemos considerarlo infalible. Los siguientes detalles parecen particularmente importantes:

- Cadenas largas de combinaciones de palabras como *encabezando la lucha del bien contra los "ejes del mal"* puede conducir a selecciones erróneas de sinónimos, puesto que cada combinación se considera sólo por pares, ignorando la combinación completa como un todo.
- Una base de datos grande de entidades nombradas se requiere para reconocer las frases como *el 11 de septiembre* o *Madison Square Garden*. Particularmente, usar el modelo de preferencias de selección puede ayudar, puesto que saber que *Madison Square Garden* es un lugar, ayuda a evaluar combinaciones como *Atrincherados / resguardados / guarecidos en el Madison Square Garden*.
- Los ajustes de umbral deben hacerse automáticamente.

8 Conclusiones

La representación de dependencias de la estructura sintáctica tiene ventajas importantes en ciertas aplicaciones, prácticamente en todo lo que se relaciona a la lexicalización y lexicografía. Sin embargo, la mayoría de las herramientas y recursos existentes, como los analizadores sintácticos, gramáticas y treebanks, están orientados al enfoque de constituyentes.

Hemos presentado un sistema que produce una estructura de dependencias con información de roles semánticos simple y robusto para el español. Usa reglas heurísticas hechas a mano para tomar decisiones sobre la pertinencia de elementos estructurales; y usa estadísticas de co-ocurrencias de palabras para la desambiguación. La estadística se aprende a partir de un corpus grande, u obtenido en la Web consultando un motor de búsqueda de una manera no supervisada, es decir, no se usa un treebank creado manualmente para el entrenamiento. En el caso de que el analizador no pueda producir un árbol completo de análisis, se regresa una estructura parcial que consiste en los enlaces de dependencias que pudo reconocer.

Para detalles de la evaluación de este sistema, vea la Sección 6.2. Después de comparar la exactitud de nuestro analizador con respecto a dos sistemas similares disponibles para el español, hemos mostrado que nuestro analizador los supera.

8.1 Formalismos gramaticales

En cuanto al formalismo de constituyentes, presentamos el enfoque de Estructuras con Características Tipificadas (TFS). Este enfoque permite el análisis a distintos niveles para la representación de un texto. Las TFS son un formalismo bien estudiado que garantiza la computabilidad de su lógica. El trabajo que hemos presentado aquí contiene ideas útiles para extraer situaciones de historias circunscritas de tal forma que posteriormente es posible hacer preguntas simples sobre el texto como quién hizo algo, o dónde alguien hizo algo. Esto puede ser usado en un sistema de búsqueda de respuestas para encontrar resultados relevantes acerca de eventos descritos en una historia.

En este trabajo sugerimos también utilizar una base de conocimientos con objetos persistentes (entidades y situaciones) para mantener la co-referencia entre oraciones en formalismos gramaticales tipo HPSG. La base de conocimientos se construye a partir del texto en una manera semiautomática. Las entidades están disponibles durante el análisis completo del texto (el lugar de

sólo una oración) y pueden ser utilizados también después de que el texto ha sido analizado, por ejemplo para búsqueda de respuestas o como una representación semántica del texto.

En cuanto a las reglas de reescritura, presentamos un sistema que puede derivar una o más instrucciones específicas de computación a partir de una petición del usuario final. Los objetos referenciados dentro de esta petición se traducen en símbolos por una gramática de reglas de reescritura con modificación de propiedades, sustitución de comodines, funciones en línea, y el uso de objetos especiales para el manejo del contexto, llamados Escenas. Las instrucciones se ejecutarán posteriormente por un sistema externo.

La representación de dependencias simplifica grandemente ciertas tareas en comparación con el enfoque de constituyentes. Por ejemplo:

- En lexicografía, reunir estadísticas de combinabilidad sintáctica de palabras individuales (*leer un libro* y *clavar un clavo* vs. **leer un clavo* y *clavar un libro*) es trivial en la representación de dependencias: únicamente se cuentan las frecuencias de arcos que conectan a las instancias de dos palabras dadas en el corpus. Una de las numerosas aplicaciones de dichas estadísticas [14, 21,22] es la desambiguación sintáctica: se prefiere el árbol con pares de palabras frecuentemente [199, 80]. En cambio, en el enfoque de estructuras de sintagmas, esto es muy difícil, si no es que imposible.
- En recuperación de información y minería de texto, empatar una frase o una consulta compleja con las oraciones en el corpus es, de nuevo, casi trivial con un árbol de dependencias: una consulta *camiseta de mangas largas y rayas rojas* concuerda fácilmente con la descripción: *una playera de seda de buena calidad con rayas amplias rojas verticales y mangas azules largas* en una base de datos de comercio electrónico, pero no con *una playera roja con rayas largas azules en las mangas*.
- En análisis semántico, transformar el árbol de dependencias en prácticamente cualquier representación semántica, como grafos conceptuales [176] o redes semánticas [124] es mucho más sencillo. De hecho, HPSG construye un tipo de árbol de dependencias para construir su representación de semántica de recursión mínima (MRS) [165].

En este trabajo hemos encontrado que el enfoque de gramática de dependencias presenta ventajas claras con respecto a otros formalismos. Nuestro analizador de estructura sintáctica con roles semánticos está basado en el formalismo de dependencias.

8.2 Preferencias de selección

En este trabajo presentamos un método para extraer preferencias seleccionales de verbos vinculados a una ontología. Es útil para resolver problemas de procesamiento de texto en lenguaje natural que requieren información acerca de la utilización de las palabras con un verbo en particular en una oración. Específicamente, hemos presentado un experimento que aplica este método para desambiguar los sentidos de palabras. Los resultados de este experimento muestran que aún existe camino por recorrer para mejorar los sistemas actuales de desambiguación de sentidos de palabras no supervisados usando preferencias seleccionales; sin embargo, hemos identificado puntos específicos para mejorar nuestro método bajo la misma línea de métodos estadísticos basados en patrones combinados con conocimiento.

Usar preferencias de selección para desambiguar la unión de FP tuvo una precisión de 78.19% y un *recall* de 76.04%. Estos resultados no son tan buenos como aquellos obtenidos con otros métodos, los cuales pueden llegar a lograr una exactitud de hasta 95%. Sin embargo, nuestro método no requiere ningún recurso costoso como un corpus anotado, ni una conexión a Internet (para usar la Web como corpus); ni siquiera se requiere el uso de una jerarquía semántica (como WordNet), puesto que las clases semánticas pueden ser obtenidas a parir de Diccionarios Explicativos Orientados al lector Humano, como se expuso en 5.3.3.

Encontramos también que, al menos para esta tarea, aplicar técnicas que usan el Web como corpus a corpus locales reduce el desempeño de estas técnicas en más del 50%, incluso si los corpus locales son muy grandes.

8.3 Métodos de suavizado

Con respecto a este tema, comparamos los resultados de los tres métodos: sin suavizado, suavizado con WordNet y suavizado con DIA (diccionario de ideas afines).

No todos los casos están cubiertos por estos métodos de suavizado ya sea porque no se puede encontrar sustituto para una palabra (como diversos acrónimos o nombres propios) o porque después de probar todas la posibles sustituciones la tripleta no se encontró en el DTC (Corpus de cuenta de tripletas de dependencia). En general, la cobertura es baja debido al tamaño del corpus para contar las frecuencias de unión. Si bien una enciclopedia provee un texto con muchas palabras diferentes, el número de uniones de FP extraídas es relativamente bajo. Creemos que usando un corpus más grande conducirá a medidas de cobertura más altas, aunque se mantendrá la misma relación entre los métodos de suavizado estudiados.

Usamos un modelo totalmente no supervisado. Esto es, en ambos algoritmos no utilizamos ninguna otra técnica de suavizado para los casos no cubiertos.

8.4 Desambiguación sintáctica

Entre los tres métodos evaluados para la unión de FP, la mejor medida promedio fue 0.707 usando suavizado con DIA, debido a su cobertura más grande comparada con otros métodos. Sin embargo, tiene una precisión menor que el suavizado usando WordNet. El método sin suavizado tiene una cobertura muy baja (0.127) pero para las uniones cubiertas los resultados fueron los mejores: 0.773, lo cual es cercano al acuerdo humano (recuerde que este acuerdo se calcula excluyendo una preposición que causa mucha desviación: *de*, la cual prácticamente siempre se une a los sustantivos). El desempeño del suavizado con WordNet podría incrementarse añadiendo información de la distribución de sentidos para cada palabra, en lugar de asumir una distribución equiprobable, aunque esto acercaría a nuestro método a los enfoques supervisados, además de que no existe un recurso actualmente que provea las distribuciones de sentidos para el español.

Nuestros resultados indican que un recurso construido automáticamente (en este caso, un diccionario de ideas afines) puede ser usado en lugar de uno construido manualmente, y aún así obtener resultados similares.

8.5 Aplicación a WSD

En la aplicación a WSD presentamos un método para extraer preferencias seleccionales vinculadas a ontologías. La información obtenida es útil para resolver diversas tareas que requieren de información acerca del uso de las palabras dado cierto verbo en una oración. En particular presentamos un experimento que aplica este método para desambiguación de sentidos de palabras. Los resultados de este experimento se encuentran aún lejos de aquellos obtenidos mediante otros métodos; sin embargo hemos partido de no requerir ninguna clase de anotación morfológica, de partes gramaticales, o semántica del texto. Además hemos identificado los puntos específicos para mejorar el funcionamiento de este método bajo la misma línea de métodos estadísticos combinados con métodos de conocimiento.

8.6 Esteganografía lingüística

Como en su versión previa, el método propuesto de esteganografía lingüística conserva el significado del texto portador, así como su ausencia de sospecha. Una de las ventajas principales de nuestro método es que no requiere una base de datos de colocaciones recopilada manualmente. En lugar de ello, se usan preferencias de selección extraídas automáticamente para extraer una base de

datos grande. Puesto que el método presentado en este capítulo se basa en recursos automáticamente obtenidos, es posible extender su aplicación a muchos lenguajes. Los resultados son menos refinados que cuando se usan colocaciones recopiladas manualmente, pero son aceptables.

- Por otra parte, el valor medio de 0.002 de ancho de banda logrado con paráfrasis local sinonímica podría parecer bajo. Un ejemplo de la entropía máxima de paráfrasis sinonímica que puede ser alcanzado, se encuentra en [14]. En este trabajo se discute que partiendo de las muestras de paráfrasis sinonímica por I. Mel'cuk, el ancho de banda máximo del método de paráfrasis en esteganografía puede alcanzar aproximadamente 0.016. Esto se logra considerando paráfrasis sinonímica para frases completas, como *ayudar* y *dar ayuda*. Compilar una lista de frases que pueden ser sustituidas es una tarea actualmente en desarrollo temprano. Nuestro método alcanza 12.5% del nivel máximo posible, sin considerar las variantes de adjetivos. El valor alcanzable de ancho de banda de paráfrasis sinonímica depende de la saturación de los recursos lingüísticos. Es por esto que éstos deben ser desarrollados fuertemente, aunque no existe un límite claro de perfección.

8.7 Aportaciones

Las aportaciones principales de este trabajo han sido:

- DILUCT: Un analizador sintáctico de dependencias para el español (realizamos pruebas contra analizadores similares, logrando un mejor desempeño. (Vea el capítulo 6)
- Una base de preferencias de selección para 3 millones de combinaciones diferentes. 0.43 millones de ellas involucran preposiciones. (Vea el capítulo 4)
- Diversos algoritmos para unión de frase preposicional. Mejora de algoritmos existentes. (Vea el Capítulo 5)
- Creación de un tesauro distribucional para el español siguiendo el método de Lin. (Vea Sección 5.3.6.3.1)
- Comparación de diccionarios manuales vs. diccionarios obtenidos automáticamente. El resultado de esta investigación sugiere que los diccionarios obtenidos automáticamente por computadora pueden sustituir a los diccionarios creados manualmente en ciertas tareas, ahorrando años de trabajo. (Vea Sección 5.3.6)
- Un método para convertir un corpus anotado de constituyentes en un corpus de dependencias (Vea Sección 6.3.2)

8.8 Trabajo futuro

Aunque cierto número de reglas gramaticales son específicas para el español, el enfoque en sí mismo es independiente del lenguaje. Como trabajo futuro planeamos desarrollar analizadores similares para otros lenguajes, incluyendo el inglés, para el cual las herramientas necesarias de preprocesamiento, como un analizador de categorías gramaticales y un lematizador, están disponibles.

Como trabajo futuro podemos mencionar en primer lugar la mejora del sistema de reglas gramaticales. Las reglas actuales en algunas ocasiones realizan su trabajo de una manera rápida aunque imperfecta, lo cual resulta en realizar la acción correcta en la mayoría de los casos, pero puede hacerse con más atención en los detalles.

Con respecto al enfoque de los marcos de Minsky, como un trabajo futuro, mediante el análisis de individuos a través de una historia, la conducta de los personajes podría ser generalizada en un modelo para predecir sus reacciones e interacciones, lo cual tendería a la adquisición del sentido común, y de esta manera realizar predicciones sobre lo que puede esperarse, en el sentido de los marcos de Minsky.

Para mejorar los resultados de desambiguación de unión de FP usando preferencias de selección, nuestra hipótesis es que en lugar de usar únicamente las 25 clases semánticas superiores, pueden obtenerse clases intermedias usando una jerarquía completa. De esta forma, sería posible tener una particularización flexible para términos comúnmente usados juntos, es decir, colocaciones, como ‘fin de año’, en tanto que se mantiene el poder de la generalización. Otro punto de desarrollo posterior es añadir un módulo de desambiguación de sentidos de palabras, de tal manera que no se tengan que considerar todas las clasificaciones semánticas para una sola palabra, como se mostró en la sección 5.3.5.

En trabajos posteriores podría explorarse usando corpus mucho más grandes para reunir cuentas de tripletas, así como experimentar con algoritmos más sofisticados que usen diccionarios de ideas afines para determinar uniones.

Por otra parte, como trabajo futuro, la información obtenida es útil no sólo para resolver problemas de WSD, sino también otros problemas importantes como desambiguación sintáctica. Por ejemplo, considere la frase *pintó un cuadro un pintor*. Existe ambigüedad entre cuál sustantivo es el sujeto y cuál es el objeto, pues el español permite un orden casi libre de constituyentes. La frase *pintó un pintor un cuadro* tiene exactamente el mismo significado. Para determinar el sujeto más probable, se puede consultar la ontología con preferencias seleccionales construida con el método presentado

en este artículo, considerando que la mayor parte de las veces el sujeto aparece del lado izquierdo. Monedero *et al*. [131] reportan que esto ocurre en un 72.6% de las oraciones.

La búsqueda de *un pintor pintó* regresa la siguiente cadena de hiperónimos valuados: `pintor→artista 1.00→creador 0.67→ ser_humano 2.48→causa 1.98`, en tanto que la búsqueda de *un cuadro pintó* regresa: `escena→situación 0.42→estado 0.34`. Es decir, pintor es más probable como sujeto para la oración mencionada anteriormente. Una implementación a mayor escala de este método queda como trabajo futuro.

Por otra parte, el ancho de banda logrado con el método de esteganografía lingüística puede ser mejorado considerando variantes de adjetivos. Esto es parte del trabajo futuro. Para mejorar nuestro algoritmo de esteganografía lingüística los siguientes detalles son particularmente importantes:

- Cadenas largas de combinaciones de palabras como *encabezando la lucha del bien contra los "ejes del mal"* puede conducir a selecciones erróneas de sinónimos, puesto que cada combinación se considera sólo por pares, ignorando la combinación completa como un todo.
- Una base de datos grande de entidades nombradas se requiere para reconocer las frases como *el 11 de septiembre* o *Madison Square Garden*. Particularmente, usar el modelo de preferencias de selección puede ayudar, puesto que saber que *Madison Square Garden* es un lugar, ayuda a evaluar combinaciones como *Atrincherados / resguardados / guarecidos en el Madison Square Garden*.

Los ajustes de umbral deben hacerse automáticamente.

Por último, resta explorar otras aplicaciones donde puedan utilizarse los diccionarios de ideas afines distribucionales junto con la información de preferencias de selección para resolver problemas como anáfora, búsqueda de respuestas, y recuperación de información.

Publicaciones derivadas de esta tesis

Las publicaciones marcadas con * están indexadas por el ISI

H. Calvo, A. Gelbukh. Action-request dialogue understanding system
In J. H. Sossa Azuela *et al* (eds.) *Avances en Ciencias de la Computación e Ingeniería de Cómputo*. Proc. CIC'2002, XI Congreso Internacional de Computación, CIC-IPN, Mexico, November 2002, v.II, 231–242

H. Calvo, A. Gelbukh, Natural Language Interface Framework for Spatial Object Composition Systems
In *Procesamiento del Lenguaje Natural No. 31*, Spain, September 2003, 285–292

H. Calvo, A. Gelbukh, Mantaining Inter-Sentential Continuity of Semantic Indices with a Knowledge Base
In *Procs. of the 2003 IEEE International Conference on Natural Language Processing and Knowledge Engineering* (NLP-KE'03), Beijing (China), October 26-29, 2003, 634–637

H. Calvo, A. Gelbukh, Improving Disambiguation of Prepositional Phrase Attachments Using the Web as Corpus*
In *Procs. of 8th Iberoamerican Congress on Pattern Recognition* (CIARP'2003), Havana (Cuba), November 2003, pp. 592–598

H. Calvo, A. Gelbukh, Extracting Semantic Categories of Nouns for Syntactic Disambiguation from Human-Oriented Explanatory Dictionaries*
In A. Gelbukh (Ed.) *Computational Linguistics and Intelligent Text Processing* (CICLing-2004). Lecture Notes in Computer Science, Vol. 2945, Springer-Verlag, 2004, pp. 254–257

H. Calvo, A. Gelbukh, Acquiring Selectional Preferences from Untagged Text for Prepositional Phrase Attachment Disambiguation*

In Farid Meziane and Elizabeth Metais (eds.) *Natural Language Processing and Information Systems*, (NLDB 2004), Lecture Notes in Computer Science 3136, Springer 2004, 207–216

H. Calvo, A. Gelbukh, Unsupervised Learning of Ontology-Linked Selectional Preferences*

In Sanfeliu, Alberto and José Ruiz-Shulcloper (eds.) *Progress in Pattern Recognition, Speech and Image Analysis* CIARP'2004. Lecture Notes in Computer Science 2905, Springer-Verlag, 2004, 604–610

H. Calvo, I. Bolshakov, Selectional Preferences for Linguistic Steganography

J. H. Sossa Azuela et al (eds.) *Avances en Ciencias de la Computación e Ingeniería de Cómputo*. Proc. CIC'2004, XIII Congreso Internacional de Computación, CIC-IPN, Mexico, October 2004, v. II, 231–242

H. Calvo, A. Gelbukh, A. Kilgarriff, Distributional Thesaurus Versus WordNet: A Comparison of Backoff Techniques for Unsupervised PP Attachment.*

In *Computational Linguistics and Intelligent Text Processing* (CICLing 2005), Lecture Notes in Computer Science 3406, Springer 2005, pp. 177–188.

A. Gelbukh, **H. Calvo**, S. Torres. Transforming a Constituency Treebank into a Dependency Treebank

In Procesamiento de Lenguaje Natural No. 35, Spain, 2005.

H. Calvo, A. Gelbukh. Extracting WordNet-like Top Concepts from Machine Readable Explanatory Dictionaries

In International Journal of Translation, Bahri Publications, Vol. 17. No. 1-2 Jan-Dec 2005, pp. 87–95.

H. Calvo, A. Gelbukh. Recognizing Situation Patterns from Self-Contained Stories. En *Advances in Natural Language Understanding and Intelligent Access to Textual Information*. Parte de Hugo Terashima-Marín, Horacio Martínez-Alfaro, Manuel Valenzuela-Rendón, Ramón Brena-Pinero (Eds.). Tutorials and Workshops Proceedings of Fourth Mexican International Conference on Artificial Intelligence, ISBN: 968-891-094-5.

J. Tejada-Cárcamo, A. Gelbukh, **H. Calvo**. Desambiguación de sentidos de palabras usando relaciones sintácticas como contexto local. En *Advances in Natural Language Understanding and Intelligent Access to Textual Information*. Parte de Hugo Terashima-Marín, Horacio Martínez-Alfaro, Manuel Valenzuela-Rendón, Ramón Brena-Pinero (Eds.). Tutorials and Workshops Proceedings of Fourth Mexican International Conference on Artificial Intelligence, ISBN: 968-891-094-5.

Referencias

1. Agirre, Eneko, David Martinez (2001) **Learning class-to-class selectional preferences** In: *Proceedings of the Workshop Computational Natural Language Learning* (CoNLL-2001), Toulousse, France.
2. Agirre, Eneko, David Martinez (2002) **Integrating selectional preferences in WordNet**. In: *Proceedings of the first International WordNet Conference*, Mysore, India.
3. Agirre, Eneko, David Martínez (2004): **Unsupervised WSD based on automatically retrieved examples: The importance of bias**. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing,* EMNLP, Barcelona, Spain.
4. Apresyan, Yuri D., Igor Boguslavski, Leonid Iomdin, Alexandr Lazurski, Nikolaj Pertsov, Vladimir Sannikov, Leonid Tsinman. (1989) **Linguistic Support of the ETAP-2 System** (in Russian). Moscow, Nauka.
5. Asoh, H., Matsui, T., Fry, J., Asano, F., and Hayamizu, S. (1999) **A spoken dialog system for a mobile office robot**, in *Proceedings of Eurospeech* '99, pp.1139-1142, Budapest.
6. Baker, C. F., C. J. Fillmore, y J. B. Lowe (1998) **The Berkeley FrameNet project**. In *Proceedings of the COLING-ACL*, Montreal, Canada.
7. Banerjee, Satanjeev, Ted Pedersen (2003) **The Design, Implementation, and Use of the Ngram Statistic Package**, in *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City, 370–381.
8. Barros, Flavia de Almeida (1995) ***A Treatment of Anaphora in Portable Natural Language Front Ends to Data Bases***, PhD Thesis. University of Essex, UK. 1995. 231p.
9. Baum, L., T. Petria, G. Soules y N. Weiss (1970) **A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains**. En *The Annals of Mathematical Statistics*, 41(1):164-171, 1970.
10. Bechhofer, S., I. Horrocks, P. F. Patel-Schneider, and S. Tessaris (1999) **A proposal for a description logic interface**. In P. Lambrix, A. Borgida, M. Lenzerini, R. Möller, and P. Patel-Schneider, editors, *Proceedings of the International Workshop on Description Logics* (DL'99), pages 33-36.
11. Belleti, A., Rizzi, L. Phych (1988) **Verbs and Q-Theory**, Natural Language and Linguistic Theory, 6, pp. 291-352.
12. Bisbal, E., A. Molina, L. Moreno, F. Pla, M. Saiz-Noeda, E. Sanchis (2003) **3LB-SAT: Una herramienta de anotación semántica**. en *Procesamiento de Lenguaje Natural* No. 31, Sociedad Española para el Procesamiento de Lenguaje Natural (SEPLN), España, pp. 193-200.
13. Bogatz, H. **The Advanced Reader's Collocation Searcher** (ARCS). http://www.elda.fr/catalogue/en/text/M0013.html

14. Bolshakov, Igor A (2004) **A Method of Linguistic Steganography Based on Collocationally-Verified Synonymy**. *Information Hiding 2004*, *Lecture Notes in Computer Science*, 3200 Springer-Verlag, 2004, pp. 180–191. BB
15. Bolshakov, Igor A (2004) **Getting One's First Million... Collocations**. In: A. Gelbukh (Ed.). *Computational Linguistics and Intelligent Text Processing*. Proc. 5th Int. Conf. CICLing-2004, LNCS 2945, Springer, 2004, p:229–242.
16. Bolshakov, Igor A (2004) **Two methods of synonymous paraphrasing in linguistic steganography** (in Russian, abstract in English). *Proc. Intern. Conf. Dialogue'2004*, Verhnevolzhskij, Russia, June 2004, p. 62-67.
17. Bolshakov, Igor A., A. Gelbukh (2000) **A Very Large Database of Collocations and Semantic Links**. In: M. Bouzeghoub *et al*. (Eds.) *Natural Language Processing and Information Systems*. Proc. Int. Conf. on Applications of Natural Language to Information Systems NLDB-2000, LNCS 1959, Springer, 2001, p:103–114. B www.gelbukh.com/ CV/Publications/2000/NLDB-2000-XLex.htm.
18. Bolshakov, Igor A., A. Gelbukh (2002) **Heuristics-Based Replenishment of Collocation Databases**. In: E. Ranchhold, N.J. Mamede (Eds.) *Advances in Natural Language Processing*. Proc. Int. Conf. PorTAL 2002, Faro, Portugal. LNAI 2389, Springer, p:25–32.
19. Bolshakov, Igor A., A. Gelbukh (2004) **Synonymous Paraphrasing Using WordNet and Internet**. In: F. Meziane, E. Métais (Eds.) *Proc. 9th International Conference on Application of Natural Language to Information Systems* NLDB-2004, LNCS 3136, Springer.
20. Bolshakov, Igor A., Alexander Gelbukh (1998) **Lexical functions in Spanish**. Proc. *CIC-98, Simposium Internacional de Computación*, Mexico, pp. 383–395; www.gelbukh.com/CV/Publications/1998/ CIC-98-Lexical-Functions.htm.
21. Bolshakov, Igor A., Alexander Gelbukh (2001) **A Large Database of Collocations and Semantic References**: Interlingual Applications. *International J. of Translation*, V.13, No.1–2, pp. 167–187.
22. Bolshakov, Igor A., Alexander Gelbukh (2003) **On Detection of Malapropisms by Multistage Collocation Testing**. *NLDB-2003, 8th Int. Conf. on Application of Natural Language to Information Systems*. Bonner Köllen Verlag, 2003, pp. 28–41.
23. Bolshakov, Igor y A. F. Gelbukh (2002) ***Computational Linguistics and Linguistic Models***. Serie Lecturas en Lingüística Computacional, Colección en Ciencia de Computación, FCE, 2002.
24. Brants, Thorsten (2000) **TNT—A Statistical Part-of-Speech Tagger**. In: Proc. *ANLP-2000, 6th Applied NLP Conference*, Seattle, Washington, USA
25. Brent, Michael R. (1993) **From grammar to lexicon: Unsupervised learning of lexical syntax**. En *Computational Linguistics*, 19(2):243-262.
26. Bresciani, Paolo, Enrico Franconi, Sergio Tessaris, (1995) **Implementing and testing expressive Description Logics**: a preliminary report. In the *Proceedings of the 1995 International Workshop on Description Logi*cs, Rome, Italy.

27. Brill, Eric (2003). **Processing Natural Language without Natural Language Processing**, In Alexander Gelbukh, ed. *Computational Linguistics and Intelligent Text Processing, 4th International Conference* CICLing 2003, pp. 360-369, Mexico, 2003.
28. Brill, Eric and Phil Resnik (1994) **A Rule Based Approach to Prepositional Phrase Attachment Disambiguation**. In Proceedings of the Fifteenth International Conference on Computational Linguistics (COLING)
29. Briscoe, Ted, John Carroll, Jonathan Graham and Ann Copestake (2002) **Relational evaluation schemes**. In: *Proceedings of the Beyond PARSEVAL Workshop at the 3rd International Conference on Language Resources and Evaluation,* Las Palmas, Gran Canaria, 4–8.
30. Burton, R. (1992) **Phrase-Structure Grammar**. In Shapiro, Stuart ed., *Encyc. of Artificial Intelligence*. Vol. 1.
31. Calvo Hiram, Alexander Gelbukh (2003) **Natural Language Interface Framework for Spatial Object Composition Systems**. *Procesamiento de Lenguaje Natural*, N 31; www.gelbukh.com/CV/Publications/2003/sepln03-2f.pdf.
32. Calvo, Hiram, Alexander Gelbukh (2003), **Improving Disambiguation of Prepositional Phrase Attachments Using the Web as Corpus**, En *Procs. of 8th Iberoamerican Congress on Pattern Recognition* (CIARP'2003), Havana (Cuba), pp. 592-598.
33. Calvo, Hiram, Alexander Gelbukh (2004) **Extracting Semantic Categories of Nouns for Syntactic Disambiguation from Human-Oriented Explanatory Dictionaries**, En A. Gelbukh, ed. *Computational Linguistics and Intelligent Text Processing* (CICLing-2004). Lecture Notes in Computer Science, Vol. 2945, Springer-Verlag.
34. Calvo, Hiram, Alexander Gelbukh (2004) **Extracting Semantic Categories of Nouns for Syntactic Disambiguation from Human-Oriented Explanatory Dictionaries**, In Gelbukh, A. (ed) *Computational Linguistics and Intelligent Text Processing*, Springer LNCS.
35. Calvo, Hiram, Alexander Gelbukh (2004) **Unsupervised Learning of Ontology-Linked Selectional Preferences**, Procs. *of Progress in Pattern Recognition, Speech and Image Analysis* CIARP'2004, LNCS, Springer, 2004. C
36. Calvo, Hiram, Alexander Gelbukh, Adam Kilgarriff (2005) **Distributional Thesaurus versus WordNet: A Comparison of Backoff Techniques for Unsupervised PP Attachment**. In: *Computational Linguistics and Intelligent Text Processing* (CICLing-2005). *Lecture Notes in Computer Science* N 3406, Springer-Verlag, pp. 177–188.
37. Calvo, Hiram, Alexander Gelbukh. (2004) **Acquiring Selectional Preferences from Untagged Text for Prepositional Phrase Attachment Disambiguation**. In: *Proc. NLDB-2004*, *Lecture Notes in Computer Science*, N 3136, pp. 207–216.
38. Cano Aguilar, R (1987) ***Estructuras sintácticas transitivas en el español actual***. Ed. Gredos. Madrid.
39. Caraballo, S. A. (1999) **Automatic construction of a hypernym-labeled noun hierarchy from text**. En *Proceedings of the 37th Annual Meeting of The Association for Computational Linguistics* [2], pp. 120-126.

40. Caraballo, S. A. (2001) ***Automatic Construction of a Hypernym-Labeled Noun Hierarchy from Text***. Tesis Doctoral, Computer Science Department, Brown University, Providence, RI, USA.
41. Caroli, F., R. Nübel, B. Ripplinger y J. Schütz (1994) **Transfer in VerbMobil**, en *IAI Saarbrücken VerbMobil-Report 11,* May 1994
42. Carpenter, Bob (1992), ***The Logic of Typed Feature Structures***. Cambridge Tracts in Theoretical Computer Science 32. Cambridge University Press.
43. Carreras, Xavier, Isaac Chao, Lluis Padró, Muntsa Padró (2004) **FreeLing: An Open-Source Suite of Language Analyzers**. *Proc. 4th Intern. Conf. on Language Resources and Evaluation (LREC-04)*, Portugal.
44. Carroll G. y M. Rooth (1998) **Valence induction with a head-lexicalized PCFG**, En *Proceedings of the 3rd Conference on Empirical Methods in Natural Language Processing*, *ACL,* pp. 36-45, Granada, Spain.
45. Carroll, John, Diana McCarthy (2000) **Word sense disambiguation using automatically acquired verbal preferences**. In *Computers and the Humanities*, 34(1-2), Netherlands.
46. Chapman, M., G.I. Davida, M. Rennhard (2001) **A Practical and Effective Approach to Large-Scale Automated Linguistic Steganography**. In: G.I. Davida, Y. Frankel (Eds.) *Information security*. Proc. of Int. Conf. on Information and Communication Security ICS 2001. LNCS 2200, Springer, p:156–165.
47. Chomsky, Noam (1957) ***Syntactic Structures***. The Hague: Mouton & Co, 1957.
48. Chomsky, Noam (1986) ***Knowledge of Language: Its Nature, Origin and Use***. New York: Praeger.
49. Choueka, Y. (1988) **Looking for needles in a haystack or locating interesting collocational expressions in large textual database**. In Proc. *Conf. User-Oriented Content-Based Text and Image Handling* (RIAO'88), 1988, p:609–623.
50. Civit, Montserrat (2003) ***Criterios de etiquetación y desambiguación morfosintática de corpus en español***. Tesis Doctoral, Departament de Lingüística, Universitat de Barcelona.
51. Civit, Montserrat, Maria Antònia Martí (2004) **Estándares de anotación morfosintáctica para el español**. *Workshop of tools and resources for Spanish and Portuguese*. IBERAMIA 2004. Mexico.
52. Clark, Stephen, David Weir (2002) **Class-based Probability Estimation Using a Semantic Hierarchy**, *Computational Linguistics* 28(2).
53. **CLiC-TALP corpus**: http://clic.fil.ub.es/recursos/corpus.shtml
54. Collins, Michael, James Brooks (1995) **Prepositional Phrase Attachment trhough a Backed-of Model**. In David Yarouwsky and Kenneth Church, eds, *Proceedings of the Third Workshop on Very Large Corpora*, pages 27-38, Cambridge, Massachussets.
55. Copestake, Ann (2001) ***Implementing Typed Feature Structure Grammars***, U. of Chicago Press, 2001.
56. Copestake, Ann, Dan Flickinger (2000), **An open-source grammar development environment and broad-coverage English grammar using HPSG**, in *Second conference on Language Resources and Evaluation (LREC-2000)*, Athens, Greece.
57. Copestake, Ann, Dan Flickinger, Ivan A. Sag. (1997) ***Minimal Recursion Semantics. An introduction***. CSLI, Stanford University.

58. Copestake, Ann, Dan Flickinger, Rob Malouf, Susanne Riehemann and Ivan Sag (1995), **Translation using Minimal Recursion Semantics**, in *Proceedings of The Sixth International Conference on Theoretical and Methodological Issues in Machine Translation* (TMI95), Leuven, Belgium
59. Craig, J., Berezner, S., Homer, C. and Longyear, C. (1966) **DEACON: Direct English Access and Control**. In *Proceedings of AFIPS Fall Joint Conference,* San Francisco, CA. Vol 29, pp. 365-380.
60. Cruse, D. A. (1986) ***Lexical semantics***, Cambridge University Press, Cambridge, England.
61. Cuetos, Fernando, Maria Antonia Martí, and Valiña Carreiras (2000), ***Léxico informatizado del Español***. Edicions de la Universitat de Barcelona.
62. Dagan I., S. Marcus, y S. Markovitch (1993) **Contextual word similarity and estimation from sparse data**. En *Proceedings of the 31st Annual Meeting of The Association for Computational Linguistics* [1]*,* pp. 164-171.
63. Dagan, I., S. Marcus, y S. Markovitch (1995). **Contextual word similarity and estimation from sparse data**. En *Computer Speech and Language*, 9(2):123-152, Abril de 1995.
64. Debusmann, Ralph, Denys Duchier, Geert-Jan M. Kruijff (2004) **Extensible Dependency Grammar: A New Methodology**. In: *Recent Advances in Dependency Grammar. Proc. of a workshop at COLING-2004*, Geneve.
65. Decadt, B. **Literature Survey Unsupervised Machine Learning of Lexical Semantics**, CNTS Language Technology Group, University of Antwerp, Belgium, http://cnts.uia.ac.be/~decadt/semaduct/uls-survey/
66. Di Eugenio, Barbara (1993) ***Understanding Natural Language Instructions: a Computational Approach to Purpose*** Clauses. Ph.D. thesis, University of Pennsylvania, December. Technical Report MS-CIS-93-91.
67. Di Eugenio, Barbara (1996) **Pragmatic overloading in Natural Language instructions**. *International Journal of Expert Systems* 9.
68. Díaz, Isabel, Lidia Moreno, Inmaculada Fuentes, Oscar Pastor (2005) **Integrating Natural Language Techniques in OO-Method**. In: Alexander Gelbukh (ed.), *Computational Linguistics and Intelligent Text Processing* (CICLing-2005), *Lecture Notes in Computer Science*, 3406, Springer-Verlag, pp. 560–571.
69. Dick, J. (1991) ***A conceptual, case-relation representation of text for intelligent retrieval***, Tesis doctoral, Department of Computer Science, University of Toronto, April 1991. Publicada como reporte técnico CSRI-265.
70. Dik, Simon C. (1989) ***The Theory of Functional Grammar, Part I: The Structure of the Clause,*** Foris Publications, Dordrecht.
71. Farreres, X., G. Rigau, H. Rodríguez (1998) **Using WordNet for Building WordNets**. In *Proceedings of COLING-ACL Workshop "Usage of WordNet in Natural Language Processing Systems*", Montreal, Canada..
72. Fillmore Charles. (1977) **The Case for Case Reopened**, En P. Cole y J. Sadock *Syntax and Semantics 8: Grammatical Relations*, Academic Press, New York, pp. 59-82.

73. Fillmore, Charles, (1968) **The Case for Case**, In *Universals in Linguistic Theory*. Edited by Bach, Emmon and Harms, Robert T., 1-90. Chicago: Holt, Rinehart and Winston.
74. Franz, Alexander (1997). **Independence Assumptions Considered Harmful**. In ACL.
75. Freitag, D.,McCallum, A (2000) **Information extraction with HMM structures learned by stochastic optimization**. *En: Proceedings of AAAI,* pp. 584–589.
76. Galicia-Haro, Sofía (2000) ***Análisis Sintáctico Conducido por un Diccionario de Patrones de Manejo Sintáctico para Lenguaje Español,*** Tesis Doctoral, Laboratorio de Lenguaje Natural, Centro de Investigación en Computación, Instituto Politécnico Nacional, México.
77. Galicia–Haro, Sofia, Alexander Gelbukh, Igor A. Bolshakov (2001) **Una aproximación para resolución de ambigüedad estructural empleando tres mecanismos diferentes**. En *Procesamiento de Lenguaje Natural*, No 27, September 2001. Sociedad Española para el Procesamiento de Lenguaje Natural (SEPLN), Spain, pp. 55-64.
78. Gazdar, Gerald (1982) **Phrase Structure Grammar**, in Jacobsen, P. and Pullum, G. K., eds., *The Nature of Syntactic Representation*, Reidel, Boston, Massachussets.
79. Gelbukh, A, G. Sidorov (2003) **Approach to construction of automatic morphological analysis systems for inflective languages with little effort**. In: Computational Linguistics and Intelligent Text Processing (CICLing-2003), Lecture Notes in Computer Science, N 2588, Springer-Verlag, 2003, pp. 215–220.
80. Gelbukh, Alexander (1999) **Syntactic disambiguation with weighted extended subcategorization frames**. *Proc. PACLING-99*, pp. 244–249.
81. Gelbukh, Alexander (2002) **Tendencias recientes en el procesamiento de lenguaje natural**. *Proc. SICOM-2002*, Villahermosa, Tabasco, México.
82. Gelbukh, Alexander y Grigori Sidorov (2004). ***Procesamiento automático del español con enfoque en recursos léxicos grandes***, IPN, 2004.
83. Gelbukh, Alexander, Grigori Sidorov, Francisco Velásquez (2003) **Análisis morfológico automático del español a través de generación**. *Escritos*, N 28, pp. 9–26.
84. Gelbukh, Alexander, Grigori Sidorov, Liliana Chanona (2002) **Corpus virtual, virtual: Un diccionario grande de contextos de palabras españolas compilado a través de Internet**. In: Julio Gonzalo, Anselmo Peñas, Antonio Ferrández, eds.: *Proc. Multilingual Information Access and Natural Language Processing, International Workshop,* in IBERAMIA-2002, VII Iberoamerican Conference on Artificial Intelligence, Seville, Spain, November 12-15, 7–14.
85. Gelbukh, Alexander, S. Torres, H. Calvo. (2005) **Transforming a Constituency Treebank into a Dependency Treebank**. Submitted to *Procesamiento del Lenguaje Natural* No. 34, Spain.
86. Gildea, D., Jurafsky, D (2002) **Automatic labeling of semantic roles**. En *Computational Linguistics 28*, pp. 245–288.
87. Goldstein, R.A., R. Nagel. (1971) **3-D Visual Simulation**. *Simulation 16*, pp. 25-31
88. Grefenstette, G. (1994) ***Explorations in Automatic Thesaurus Discovery***. Kluwer.

89. Grosz, B.J., D. Appelt, P. Martin, F.C. N. Pereira (1987) **TEAM: An Experiment in the Design of Transportable Natural-Language Interfaces**. In *Artificial Intelligence*, vol 32, pp 173-243.
90. Gruber, J. (1967) ***Studies in Lexical Relations***, Disertación doctoral en el MIT y en *Lexical Structures in Syntax and Semantics*, North Holland.
91. Haarslev, Volker, Ralf Möller (2000) **Consistency Testing: The RACE Experience**, In *Proceedings of Automated Reasoning with Analytic Tableaux and Related Methods* TABLEAUX 2000, University of St Andrews, Scotland, 4-7 July, Springer-Verlag.
92. Harris, L. (1984), **Experience with INTELLECT: Artificial Intelligence Technology Transfer**. In *The AI Magazine*, vol 2(2), pp 43-50.
93. Hendrix, G.G., E. Sacerdoti, D. Sagalowowicz, J. Slocum (1978) **Developing a Natural Language Interface to Complex Data**. In *ACM transactions on Database Systems*; vol 3(2), pp 105-147.
94. Hindle, Don (1990) **Noun classification from predicate-argument structures**. En *Proceedings of the 28th Annual Meeting of The Association for Computational Linguistics*, ACL, pp. 268-275, University of Pittsburgh, Pittsburgh, PA, USA.
95. Hindle, Don, Mats Rooth (1993) **Structural ambiguity and lexical relations**. *Computational Linguistics* 19:103–120.
96. Hoffmann, T. (1999) **Probabilistic latent semantic indexing**. En *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval*, ACM, pp. 50-57, University of California, Berkeley, CA, USA.
97. Hoppe, Th., C. Kindermann, J.J. Quantz, A. Schmiedel, M. Fischer (1993) ***BACK V5 Tutorial & Manual,*** KIT Report 100, Tech. Univ. Berlin.
98. Hoppenbrowers, J., B. van der Vos, y S. Hoppenbrouwers (1996) **NL Structures and Conceptual Modelling: The KISS Case**. En R. van de Riet, J. Burg y A. van der Vos, eds, *Application of Natural Language to Information Systems,* pp. 197-209. IOS Press.
99. Huffman (1996) **Learning information extraction patterns from examples**. En Wermter, S.,Scheler, G., Riloff, E., eds.: *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing. Springer-Verlag,* pp .246–260.
100. Hughes, J, E. Atwell (1994) **The automated evaluation of inferred word classifications**. En *Proceedings of the 11th European Conference on Artificial Intelligence*, ECCAI, pp. 535-539, Amsterdam, The Netherlands, ECCAI.
101. Jackendoff, R (1972) ***Semantic Interpretation in Generative Grammar,*** MIT PRess, Cambridge.
102. Jackendoff, R. (1990) ***Semantic Structures***, Cambridge, Mass., The MIT PRess.
103. Joshi, Aravind (1992) **Phrase-Structure Grammar**. In Shapiro, Stuart ed., *Encyclopedia of Artificial Intelligence*. Vol. 1. John Wiley & Sons, Inc. Publishers, New York.
104. Jurafsky, Daniel, James H. Martin (2000) ***Speech and Language Processing***. Prentice Hall, 2000. p. 672.
105. Kay, Martin (1979) **Functional grammar**. In *Proceedings of the 5th Annual Meeting of the Berkeley Linguistic Society*. 142-158.

106. Keller, Frank, Mirella Lapata (2003) **Using the Web to Obtain Frequencies for Unseen Bigrams**. *Computational Linguistics* 29:3.

107. Kilgarriff, Adam (2003) **Thesauruses for Natural Language Processing**. *Proceedings of NLP-KE 03*, Beijing, China, 5–13.

108. Knight, Kevin (1992) **Unification**. In Stuart Shapiro (ed.), *Encyclopedia of Artificial Intelligence*. Vol. 2. John Wiley & Sons, Inc. Publishers, New York.

109. Kudo, T., Y. Matsumoto (2000) **Use of Support Vector Learning for Chunk Identification**. In *Proceedings of CoNLL-2000 and LLL-2000*, Lisbon, Portugal.

110. Lara, Luis Fernando (1996) ***Diccionario del español usual en México***. Digital edition. Colegio de México, Center of Linguistic and Literary Studies.

111. Lázaro Carreter, F. (Ed.) (1991) ***Diccionario Anaya de la Lengua***, Vox. L

112. Leek, T. (1997) ***Information extraction using hidden Markov models***. Tesis de maestría, U C San Diego.

113. Levin, B. (1993) ***English verb classes and alternations: a preliminary investigation***. The University of Chicago Press, Chicago, IL, USA.

114. Li, Hang, Naoki Abe (1998) **Word clustering and disambiguation based on co-ocurrence data**. *Proceedings of COLING '98*, 749–755.

115. Lin, Dekang (1998) **An information-theoretic measure of similarity**. *Proceedings of ICML'98*, 296–304.

116. Lüdtke, Dirk, Satoshi Sato (2003) **Fast Base NP Chunking with Decision Trees - Experiments on Different POS Tag Settings**. In Gelbukh, A. (ed.) *Computational Linguistics and Intelligent Text Processing*, Springer LNCS, 2003, pp. 136-147.

117. M. A. Hearst (1992) **Automatic acquisition of hyponyms from large text corpora**. En *Proceedings of the 14th International Conference on Computational Linguistics*, ICCL, pp. 120-126, Nantes, France.

118. MacGregor, Robert (1991) **Using a Description Classifier to Enhance Deductive Inference**, In *Proceedings of the Seventh IEEE Conference on AI Applications*, Miami, Florida, February, pages 141-147.

119. Manning, C. D. (1993) **Automatic acquisition of a large subcategorization dictionary from corpora**. En *Proceedings of the 31st Annual Meeting of The Association for Computational Linguistics*, ACL, pp. 235-242.

120. Manning, C. D., H. Schütze (1999). ***Foundations of Statistical Natural Language Processing***. The MIT Press, Cambridge, MA, USA, second edition.

121. McLauchlan, Mark (2004) **Thesauruses for Prepositional Phrase Attachment**. *Proceedings of CoNLL-2004*, Boston, MA, USA, 73–80.

122. Mel'čuk, Igor A (1996) **Lexical Functions: A Tool for the Description of Lexical Relations in the Lexicon**. In: L. Wanner (ed.), *Lexical Functions in Lexicography and Natural Language Processing*, Amsterdam/Philadelphia: Benjamins, 37–102.

123. Mel'čuk, Igor A. (1981) **Meaning-text models: a recent trend in Soviet linguistics**. *Annual Review of Anthropology* 10, 27–62.

124. Mel'čuk, Igor A. (1988) ***Dependency Syntax: Theory and Practice***. State University Press of New York, 1988. M

125. Merlo, Paola, Matthew W. Crocker, Cathy Berthouzoz (1997) **Attaching Multiple Prepositional Phrases: Generalized Backer-off Estimation**. In Claire Cardie and Ralph Weischedel, editors, *Second Conference on Empirical Methods in Natural Language Processing*, pp 149-155, Providence, R.I., August 1-2.

126. Microsoft, **Biblioteca de Consulta Microsoft Encarta 2004**, Microsoft Corporation, 1994–2004.

127. Miller, George (1990) **WordNet: An on-line lexical database**, In *International Journal of Lexicography*, 3(4), December 1990, pp. 235-312

128. Miller, George (1994) **Nouns in WordNet: a Lexical Inheritance System**, En *International Journal of Lexicography*, Volumen 3. núm. 4, pp. 245-264.

129. Minsky, Marvin (1975) **A Framework for Representing Knowledge**, in P. Winston (ed.): *The Psychology of Computer Vision*, McGraw Hill, New York, pp. 211- 277.

130. Mitchell, Brian (2003) ***Prepositional phrase attachment using machine learning algorithms***. Ph.D. thesis, University of Sheffield, 2003.

131. Monedero, J., J. González, J. Goñi, C. Iglesias, A. Nieto (1995) **Obtención automática de marcos de subcategorización verbal a partir de texto etiquetado: el sistema SOAMAS**. In *Actas del XI Congreso de la Sociedad Española para el Procesamiento del Lenguaje Natural* SEPLN 95, Bilbao, Spainm 241—254.

132. Montes-y-Gómez, Manuel, Alexander F. Gelbukh, Aurelio López-López (2002) **Text Mining at Detail Level Using Conceptual Graphs**. In: Uta Priss *et al*. (Eds.): *Conceptual Structures: Integration and Interfaces*, 10th Intern. Conf. on Conceptual Structures, ICCS-2002, Bulgaria. *Lecture Notes in Computer Science*, N 2393, Springer-Verlag, pp. 122–136; ccc.inaoep.mx/~mmontesg/publicaciones/2002/DetaiedTM-iccs02.pdf.

133. Montes-y-Gómez, Manuel, Aurelio López-López, and Alexander Gelbukh (2000) **Information Retrieval with Conceptual Graph Matching**. *Proc. DEXA-2000, 11th Intern. Conf. on Database and Expert Systems Applications, England. Lecture Notes in Computer Science*, N 1873, Springer-Verlag, pp. 312–321.

134. Morales-Carrasco, R. y Alexander Gelbukh (2003) **Evaluation of TnT Tagger for Spanish**, in Procs. *Fourth Mexican International Conference on Computer Science ENC'03*, Tlaxcala, México, pp. 18-28.

135. Moravcsik, J (1991) **What Makes Reality Intelligible? Reflections on Aristotle's Theory of *Aitia*** En L. Judson, ed., *Aristotle's Physics: A Collection of Essays*. New York: Clarendon, pp. 31-48.

136. Navarro, Borja, Montserrat Civit, M. Antonia Martí, R. Marcos, B. Fernández (2003) **Syntactic, semantic and pragmatic annotation in Cast3LB**. *Shallow Processing of Large Corpora (SProLaC)*, a Workshop of Corpus Linguistics, Lancaster, UK.

137. Nebel, Bernhard (1999) **Frame-Based Systems**, in Robert A. Wilson and Frank Keil (eds.), *MIT Encyclopedia of the Cognitive Sciences*, MIT Press, Cambridge, MA, pp. 324-325.

138. Nebel, Bernhard (2001) **Logics for Knowledge Representation**, in N. J. Smelser and P. B. Baltes (eds.), *International Encyclopedia of the Social and Behavioral Sciences*, Kluwer, Dordrecht, 2001.

139. Nebel, Bernhard, Gert Smolka (1991) **Attributive description formalisms... and the rest of the world**, in O. Herzog and C. Rollinger, eds., *Text Understanding in LILOG*, Springer, Berlin, pp. 439-452.

140. Ogden, C. K., I. A. Richards (1984) ***The Meaning of Meaning***, 8th ed. (1923; New York: Harcourt Brace Jovanovich, 1946), pp.9-12. cited on J.F. Sowa, *Conceptual Structures. Information Processing in Mind and Machine*. Addison Wesley, 1984, p.11.

141. ***Oxford Collocations Dictionary for Students of English***. Oxford University Press, 2003.

142. Pantel, Patrick, Dekang Lin (2000) **An Unsupervised Approach to Prepositional Phrase Attachment using Contextually Similar Words**. *Proceedings of Association for Computational Linguistics (ACL-00)*, Hong Kong, 101–108.

143. Patel-Schneider, Peter F., Merryll Abrahams, Lori Alperin Resnick, Deborah L. McGuinness, and Alex Borgida (1996) ***NeoClassic Reference Manual: Version 1.0*** Artificial Intelligence Principles Research Department, AT&T Labs Research.

144. Pearce, Darren (2002) **A Comparative Evaluation of Collocation Extraction Techniques**. In Procs. *Third International Conference on Language Resources and Evaluation,* Las Palmas, Canary Islands, Spain.

145. Pereira, F., N. Tishby, L. Lee (1993) **Distributional clustering of english words**. En *Proceedings of the 31st Annual Meeting of The Association for Computational Linguistics*, ACL pp. 183-190.

146. Pineda, L. A., A. Massé, I. Meza, M. Salas, E. Schwarz, E. Uraga, L. Villaseñor (2002). ***The DIME Project***. Department of Computer Science, IIMAS, UNAM.

147. Pineda, L. A., G. Garza (2000). **A Model for Multimodal Reference Resolution**. *Computational Linguistics*, Vol. 26, No. 2., pp. 139-193.

148. Pollard, Carl and Ivan Sag (1994) ***Head-Driven Phrase Structure Grammar***. University of Chicago Press, Chicago, IL and London, UK.

149. Prescher, D., Riezler S., M. Rooth (2000) **Using a probabilistic class-based lexicon for lexical ambiguity resolution**. In *Proceedings of the 18th International Conference on Computational Linguistics*, Saarland University, Saarbrücken, Germany, July-August 2000. ICCL.

150. Pullum, Geoffrey K. (1999), **Generative Grammar**, In Frank C. Keil and Robert A. Wilson (eds.), *The MIT Encyclopedia of the Cognitive Sciences*, pp. 340-343, Cambridge, MA, The MIT Press.

151. Pustejovsky, J (1995) ***The Generative Lexicon,*** Cambridge, Mass.: MIT Press.

152. Rappaport, M., B. Levin (1988) **What to do with Θ-roles?** En W. Wilkins, ed. *Syntax and Semantics 21: Thematic Relations*, Academic Press.

153. Ratnaparkhi Adwait, Jeff Reynar, and Salim Roukos (1994) **A Maximum Entropy Model for Prepositional Phrase Attachment**. In *Proceedings of the ARPA Human Language Technology Workshop*, 1994, pp. 250-255.
154. Ratnaparkhi, Adwait (1998) **Statistical Models for Unsupervised Prepositional Phrase Attachment**, In *Proceedings of the 36th ACL and 17th COLING*, pp. 1079-1085.
155. Ratnaparkhi, Adwait (1998) **Unsupervised Statistical Models for Prepositional Phrase Attachment**. *Proceedings of COLINGACL98*, Montreal, Canada.
156. Resnik, Philip (1993) ***Selection and Information: A Class-Based Approach to Lexical Relationships***. TesisDoctoral, University of Pennsylvania.
157. Resnik, Philip (1996) Selectional constraints: An information-theoretic model and its computational realization. *Cognition*, 61(1-2):127-159.
158. Resnik, Philip (1997) **Selectional preference and sense disambiguation**. En *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, ACL, pp. 52-57, Washington, DC, USA.
159. Riloff, E. y J. Shepherd (1999) **A corpus-based bootstrapping algorithm for semi-automated semantic lexicon construction**. *Natural Language Engineering*, 5(2):147-156.
160. Roark, B. y E. Charniak (1998) **Noun-phrase co-occurence statistics for semi-automatic semantic lexicon construction**. En *Proceedings of the 17th International Conference on Computational Linguistics*, ICCL, pp. 1110-1116, Université de Montréal, Montréal, Canada.
161. Rooth M., S. Riezler, D. Prescher, G. Carroll y F. Beil. (1998) **EM-Based clustering for NLP applications**. *Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung,* 4(3):97-124 Lehrstuhl für Theoretische Computerlinguistik, Universität Stuttgart.
162. Rooth, M. (1995) **Two-dimensional clusters in grammatical relations**. En *Proceedings of the Symposium on Representation and Acquisition of Lexical Knowledge*, AAAI, Stanford University, Stanford, CA, USA.
163. Rooth, M., S. Riezler, D. Prescher, G. Carroll y F. Beil (1999) **Inducing a semantically annotated lexicon via EM-Based clustering**. En *Proceedings of the 37th Annual Meeting of The Association for Computational Linguistics* [2], ACL.
164. Roth, D. (1998) **Learning to Resolve Natural Language Ambiguities: A Unified Approach**. In *Proceedings of AAAI-98*, Madison, Wisconsin, 806–813.
165. Sag Ivan, Tom Wasow, Emily M. Bender (2003) ***Syntactic Theory. A Formal Introduction*** (2nd Edition). CSLI Publications, Stanford, CA.
166. Sag, Ivan A., Tom Wasow (1999) ***Syntactic Theory: A Formal Introduction***, Center for the study of language and information, CSLI Publications.
167. Schulte, S. (1998) **Automatic semantic classification of verbs according to their alternation behaviour**. *Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung*, 4(3):55-96, Lehrstuhl für Theoretische Computerlinguistik, Universität Stuttgart.

168. Schütze, H. (1992). **Dimensions of meaning**. En *Proceedings of the 6th International Conference on Supercomputing*, ACM, pp. 787-796, Minneapolis, MN, USA.

169. Sebastián, N., M. A. Martí, M. F. Carreiras y F. Cuestos (2000) ***Lexesp, léxico informatizado del español***, Edicions de la Universitat de Barcelona.

170. Shieber, Stuart (1986) **An Introduction to Unification-Based Approaches to Grammar**, CSLI Publications.

171. Shinyama, Y., Tokunaga, T., Tanaka, H. (2000) **Kairai - Software Robots Understanding Natural Language**. *Third Int. Workshop on Human-Computer Conversation*, Bellagio, Italy.

172. Somers, H. (1987) ***Valency and Case in Computational Linguistics***, Edinburgh Information Technology Series 3, Edinburgh University Press.

173. Sowa, J. F. (1996) **Top-level ontological categories**, En *International Journal of Human-Computer Studies*, **43:5/6**, pp. 669-686.

174. Sowa, J. F. (1996) **Top-level ontological categories**, En *International Journal of Human-Computer Studies*, 43:5/6, pp. 669-686.

175. Sowa, J. F. (2000) ***Knowledge Representation: Logical, Philosophical, and Computational Foundations***, Brooks Cole Publishing Co., Pacific Grove, CA.

176. Sowa, John F. (1984) ***Conceptual Structures: Information Processing in Mind and Machine***. Addison-Wesley Publishing Co., Reading, MA.

177. Sparck Jones, Karen (1986) ***Synonymy and Semantic Classification***. Edinburgh University. Press, 1986

178. Steele, James (ed.) (1990) ***Meaning-Text Theory. Linguistics, Lexicography, and Implications***. Ottawa: Univ. of Ottawa Press.

179. Stetina, Jiri, Makoto Nagao (1997) **Corpus based PP attachment ambiguity resolution with a semantic dictionary**. *Proceedings of WVLC '97*, 66–80.

180. Suárez, A., M. Palomar (2002) **A Maximum Entropy-based Word Sense Disambiguation System**. In: Hsin-Hsi Chen and Chin-Yew Lin, eds.: *Proceedings of the 19th International Conference on Computational Linguistics*, COLING 2002, Taipei, Taiwan, vol. 2, 960—966.

181. Tapanainen, Pasi (1999) ***Parsing in two frameworks: finite-state and functional dependency grammar***. Academic Dissertation. University of Helsinki, Language Technology, Department of General Linguistics, Faculty of Arts.

182. Tesnière Lucien (1959) ***Eléments de syntaxe structurale***. Paris: Librairie Klincksieck.

183. Thompson C., R. Levy, C. D. Manning (2003) **A Generative Model for Semantic Role Labeling**. En *Proceedings of ECML'03*.

184. V. Gladki (1985) ***Syntax Structures of Natural Language in Automated Dialogue Systems*** (in Russian). Moscow, Nauka.

185. Vandeghinste, Vincent (2002) **Resolving PP Attachment Ambiguities Using the WWW**. In the *Thirteenth meeting of Computational Linguistics in the Netherlands*, CLIN 2002 Abstracts, Groningen, 2002

186. Volk, Martin (2000) **Scaling up. Using the WWW to resolve PP attachment ambiguities**. In *Proceedings of Konvens* 2000*,* Ilmenau, October 2000.

187. Volk, Martin (2001) **Exploiting the WWW as a corpus to resolve PP attachment ambiguities**. In *Proceeding of Corpus Linguistics* 2001. Lancaster.

188. Watt, W. (1968) **Habitability**. *American Documentation* 19, pp. 338-351.

189. Webber, Bonnie (1995) **Instructing animated agents: Viewing language in behavioral terms**. *Proceedings of the International Conference on Cooperative Multi-modal Communications*, Eindhoven, Netherlands.

190. Weeds, Julie (2003) ***Measures and Applications of Lexical Distributional Similarity***. Ph.D. thesis. University of Sussex.

191. Weinreich, Uriel (1972) ***Explorations in Semantic Theory***, Mouton, The Hague.

192. Weischedel, R.M. (1989) **A Hybrid Approach to Representation in the JANUS Natural Language Processor**. In *Proceedings of the* 27th *ACL*, Vancouver, British Columbia. pp 193-202.

193. Williams, E. (1981) Argument Structure and Morphology, *Linguistic Review*, 1, 81-114.

194. Winograd, Terry (1972) ***Understanding Natural Language***. New York: Academic Press.

195. Winograd, Terry (1983) ***Language as a Cognitive Process. Volume I: Syntax***. Stanford University. Addison-Wesley Publishing Company.

196. Woods, W.A., R.M. Kaplan, R.M., B. L. Nash-Webber (1972) ***The Lunar Science Natural Language Information System: Final Report*****,** BBN Report No. 2378. Bolt, Beranek and Newman Inc. Cambridge, MA.

197. Yarowsky, D. (2000) **Hierarchical decision lists for word sense disambiguation**. In *Computers and the Humanities*, 34(2) 179–186.

198. Yarowsky, David, S. Cucerzan, R. Florian, C. Schafer, R. Wicentowski (2001) **The Johns Hopkins SENSEVAL-2 System Description**. In: Preiss and Yarowsky, eds.: *The Proceedings of SENSEVAL-2: Second International Workshop on Evaluating Word Sense Disambiguation Systems*, Toulouse, France, (2001) 163–166

199. Yuret, Deniz (1998) Discovery of Linguistic Relations Using Lexical Attraction, PhD thesis, MIT.

200. Zavrel, Jakub, Walter Daelemans (1997) **Memory-Based Leraning: Using Similarity for Smoothing**. *Proc. ACL'97*

201. Zeling S. Harris (1968) ***Mathematical Structures of Language***. Wiley & Sons, New York, NY, USA, 1968.