Instituto Politécnico Nacional

Centro de Investigación en Computación

Maestría en Ciencias de la Computación

Laboratorio de Lenguaje Natural y Procesamiento de Texto

Aprendizaje automático de la base de datos estadística de combinaciones de palabras en español

TESIS QUE PRESENTA

Lic. Tania Lugo García

PARA OBTENER EL GRADO DE

MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

DIRECTOR DE TESIS

Dr. Alexander Gelbukh

México, D. F., 2006

SIP-14

INSTITUTO POLITECNICO NACIONAL

SECRETARIA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 18:00 horas del día 6 del mes de Noviembre de 2006 se reunieron los miembros de la Comisión Revisora de Tesis designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis de grado titulada:

"APRENDIZAJE AUTOMÁTICO DE LA BASE DE DATOS ESTADÍSTICA DE COMBINACIONES DE PALABRAS EN ESPAÑOL"

LUGO	**GARCÍA**	**TANIA**
(Apellido Paterno)	(Apellido Materno)	(Nombre)

Con registro: B 0 1 1 4 1 5

aspirante al grado de: ***MAESTRA EN CIENCIAS DE LA COMPUTACIÓN***

Después de intercambiar opiniones los miembros de la Comisión manifestaron ***SU APROBACIÓN DE LA TESIS***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Presidente

Dr. Igor Bolshakov

Secretario

Dr. Sergio Suárez Guerra

Primer vocal (Director)

Dr. Alexander Gelbukh Kahn

Segundo vocal

Dr. Héctor Jiménez Salazar

Tercer Vocal

Dra. Sofía Natalia Galicia Haro

Suplente

M. en C. Alejandro Botello Castillo

EL PRESIDENTE DEL COLEGIO

Dr. Hugo César Coyote Estrada

INSTITUTO POLITECNICO NACIONAL
CENTRO DE INVESTIGACION
EN COMPUTACION
DIRECCION

INSTITUTO POLITECNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESION DE DERECHOS

En la Ciudad de México, D.F. el día 10 del mes noviembre del año 2006, el (la) que suscribe Tania Lugo García alumno (a) del Programa de Maestría en Ciencias de la Computación con número de registro B011415, adscrito a Centro de Investigación en Computación, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de Dr. Alexander Gelbukh y cede los derechos del trabajo intitulado "Aprendizaje automático de la base de datos estadística de combinaciones de palabras en español", al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección Av. Juan de Dios Bátiz s/n casi esq. Miguel Otón de Mendizábal, Unidad Profesional "Adolfo López Mateos" Edificio CIC. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Nombre y firma

AGRADECIMIENTOS

Al **Instituto Politécnico Nacional** (**IPN**), por ofrecer programas de excelencia como son los que se imparten en el Centro de Investigación en Computación.

Al **Centro de Investigación en Computación**, a todos sus maestros, personal de apoyo, personal académico, porque día a día y en cada pequeña o grande acción ayudan a la profesionalización de este país. ¡Gracias!

Al **Consejo Nacional de Ciencia y Tecnología** (**CONACyT**), la **Coordinación General de Posgrado e Investigación** y a la **Fundación TELMEX**, por el apoyo económico y tecnológico brindado durante el desarrollo de esta maestría y este trabajo de tesis.

Al **Dr. Alexander Gelbukh** por su orientación, tiempo y apoyo en la realización de este trabajo.

RESUMEN

En el procesamiento del lenguaje natural, el análisis sintáctico es una de las áreas con mayor riqueza en cuanto a conocimiento requerido y con mayores problemas a resolver.

Entre los problemas del análisis sintáctico, la ambigüedad tiene gran importancia, y se genera debido a la inmensa variedad de interpretaciones posibles que existen para una sola oración.

Este problema ha sido manejado mediante diversos métodos, que combinan técnicas lingüísticas como gramáticas libres de contexto probabilísticas, redes semánticas, uso de diccionarios, entre otras, con métodos estadísticos y técnicas propias de la inteligencia artificial.

En esta tesis presentamos una modificación del método presentado por Galicia-Haro et al., para el análisis sintáctico de textos: a saber, se investiga el efecto del uso de combinaciones de palabras en lugar de patrones de manejo sintáctico.

El método compila una base de datos de elementos como patrones de manejo o, en el caso de esta tesis combinaciones de palabras, por medio de un algoritmo que crea las variantes y les asigna pesos estadísticos basados en su frecuencia, iterativamente, hasta que los pesos convergen. Con esta base de datos es posible desambiguar variantes de árboles sintácticos.

ABSTRACT

From Natural Language Processing, syntactic analysis is one of the richest areas in knowledge requirement, and also in issues.

Syntactic disambiguation is one of the main issues in text analysis. Syntactic ambiguity appears due to the possibility to interpret a given sentence in different ways, each of which are equally legal from syntactic point of view, and all but one of them lead to incorrect semantic interpretation of the sentence.

Many existing methods of syntactic disambiguation are statistical in nature, often combined with other methods that involve lexical or semantic knowledge.

In this thesis we present a modification to the method suggested by Galicia-Haro et al. for syntactic analysis; namely, we use word combinations or *collocations* database instead of government syntactic patterns.

The referenced method compiles a database of government patterns, in our case word combinations, by means of an algorithm that generates the variants or entries to the database and compiles the frequency for all the combinations , iteratively until the calculated weights converged. The main use of this database is the disambiguation of syntactic trees.

ÍNDICE DE ALTO NIVEL

ÍNDICE DETALLADO

RELACIÓN DE FIGURAS y TABLAS

CAPÍTULO 1 INTRODUCCIÓN

En este capítulo introducimos al lector al tema de la tesis y justificamos su desarrollo.

1.1 Ubicación

El procesamiento del lenguaje natural, que es objeto de estudio de esta tesis, se basa en teorías de lingüística teórica. La lingüística teórica considera cinco niveles de análisis en la comprensión de textos: fonético / fonológico, morfológico, sintáctico, semántico y pragmático.

La ambigüedad se presenta en cada nivel de análisis debido a que cada uno de sus objetos, que pueden ser sonidos, fonemas, palabras, oraciones, el sentido de las oraciones o uso de las oraciones de acuerdo al contexto, puede tener más de una interpretación o clasificación.

Por ejemplo considérese la siguiente oración:

Oigo la música que tocan con alegría

Esta expresión puede tener más de un sentido:

1. Que el sujeto esté alegre al escuchar la música.

2. Que los músicos estén tocando con alegría y el sujeto esté escuchando.

Para analizar una expresión como en el ejemplo anterior, ésta debe ser partida en constituyentes, y se debe determinar el rol que cada constituyente juega. Durante este proceso, se deben contestar las siguientes preguntas:

– ¿Cuál es la categoría sintáctica de las palabras? ¿Son sustantivos, verbos, adjetivos, adverbios, etc?

– ¿Cuáles son los constituyentes más grandes de la oración? Por ejemplo, que frases nominales, frases verbales, frases preposicionales, y cláusulas subordinadas ocurren en la oración?

– ¿Cómo deben ser combinados los constituyentes para formar toda la estructura sintáctica en la oración?

Estas preguntas generan más de una respuesta, resultando en más de una configuración. La preposición *con* en la oración: *oigo la música que tocan con alegría*, puede referirse a los músicos o al sujeto. Este es un ejemplo de ambigüedad y esta ambigüedad en particular es la que en esta tesis, por medio del método presentado en [GALICIA, 00], deseamos resolver:

- ¿Oigo con alegría?

- ¿Tocan con alegría?

1.2 Objetivo general

Al desarrollar este trabajo de tesis, se desea alcanzar el siguiente objetivo general:

– Modificación del método de desambiguación sintáctica presentado por Galicia-Haro et al. de tal manera que se base en las estadísticas de combinaciones de palabras y no en patrones de manejo sintáctico.

1.3 Metas particulares

Las metas particulares que se desean alcanzar al desarrollar este trabajo de tesis son:

– La modificación del método Galicia-Haro, et al. para generar las combinaciones de palabras que utilizará el algoritmo iterativo.

– La construcción de la base de datos estadística de combinaciones de palabras en español (diccionario) por medio del algoritmo iterativo aplicado al diccionario de combinaciones de palabras en español.

– El uso del diccionario para la desambiguación de árboles sintácticos.

– La evaluación del método con la modificación propuesta que consiste en el uso de combinaciones de palabras en español.

1.4 Importancia y relevancia: justificación de la investigación

El análisis del lenguaje natural es el proceso de recuperación de la estructura de una oración. Por recuperación de la estructura de una oración entenderemos técnicas de programación de lenguaje natural, utilizadas para determinar la función de cada objeto que forma una oración.

Las oraciones no son solo objetos lingüísticos, sino que poseen una estructura interna. Otros objetos como palabras, frases y cláusulas también son estructuras. Durante el análisis, la estructura lingüística se recupera en cada uno de sus niveles.

La lingüística ha documentado regularidades significativas en cada nivel de análisis. Por ejemplo, una regularidad en el análisis morfológico sería, por ejemplo: En español, añadiendo el sufijo *–or* y alguna otra inflexión dependiendo del verbo, se genera un sustantivo que significa “persona que ejecuta el acto denotado por el sustantivo”. Así de jugar generamos jugador. Otra regularidad del español podría ser: una preposición está relacionada con el sustantivo, verbo o frase nominal de la parte de la oración en la que se encuentre, sea esta sujeto, verbo o

complemento.

Estas regularidades son usadas durante el análisis para recuperar la estructura de la entrada, y por lo tanto para desenmarañar la información utilizada en la entrada. Y aquí es cuando entra la ambigüedad: Tomando cada nivel aisladamente, más de una regla puede ser aplicada en muchos casos. Por ejemplo, si hay una palabra que termine con el sufijo *-or*, puede ser un sustantivo del tipo descrito arriba o puede ser un sustantivo simple como calor, amor.

La ambigüedad ocurre en cada nivel de análisis, y tiene el potencial de multiplicarse a través de los niveles. (Algunas veces otras restricciones solo permiten algunas interpretaciones). Esto rápidamente resulta en un gran número de posibles interpretaciones de una sola oración.

Existen muchos más puntos de decisión durante el análisis de lenguaje natural que generan ambigüedad. En esta tesis la discusión se limita a la ambigüedad de las Partes de la Oración (*parts-of-speech* en inglés) y específicamente a la ambigüedad sintáctica.

Por ejemplo: en la oración *la niña con vestido rojo juega a saltar la cuerda con nudos*, cambiando una palabra, cambia la interpretación de las oración y en el análisis sintáctico, cambia la estructura de las salidas intermedias, veamos: *la niña con vestido rojo juega a saltar la cuerda con su amiga* genera un árbol sintáctico *correcto* distinto en estructura.

Se observa que en este último ejemplo la preposición *con* de la frase *con su amiga* se refiere o depende de *niña*. La combinación de sustantivo - preposición - sustantivo que debe generar el análisis sintáctico es: *niña con amiga*.

La base de datos de combinaciones de palabras en español, con pesos asignados a las variantes, ayudará en herramientas que requieran de análisis sintáctico para

verificar la elegibilidad de las estructuras intermedias del análisis para uso en herramientas como: resumen automático de información, traducción automática de textos, extracción automática de información, entre otras.

La importancia de esta tesis radica en que la generación análisis sintácticos sin ambigüedad se utiliza en muchas tareas del lenguaje natural, incluyendo recuperación de información, extracción de datos desde texto, resumen de textos y clasificación de textos.

Existen muchas técnicas para análisis del lenguaje natural, desde métodos de inteligencia artificial tales como análisis conceptual, hasta técnicas gramaticales basadas en valor-atributo para formalismos como HPSG ó *Head-driven Phrase Structure Grammar*, que es una gramática de unificación basada en rasgos y valores, un formalismo declarativo y basado en la estructura superficial de la oración.

Pero cuando estas técnicas se aplican a un vocabulario más amplio, los mecanismos de análisis no alcanzan niveles óptimos debido a las múltiples ambigüedades que se generan.

Se han desarrollado varias técnicas para la desambiguación sintáctica de textos, pero aún quedan por desarrollar modelos eficientes que abarquen solo algunos tipos específicos y relevantes para la lingüística como por ejemplo la desambiguación de frases preposicionales en el nivel sintáctico, lo cual es materia de esta tesis; por tanto, consideramos que se justifica ampliamente su desarrollo.

1.5 Aportaciones

En esta sección se presentan las aportaciones que el desarrollo de esta tesis generó.

– Un método sencillo para compilar automáticamente una base de datos de combinaciones de palabras en español, con pesos asignados de acuerdo al método presentado en [GALICIA, 00].

– Evaluación de un método completo y robusto para la desambiguación sintáctica como es el presentado en [GALICIA, 00].

– Evaluación de los parámetros utilizados en el método [GALICIA, 00].

– Utilizar una estructura de datos diferente a la presentada en el método de referencia aporta al mismo, un nuevo módulo que puede utilizarse como entrada al módulo de votación.

1.6 Estructura de la tesis

La elaboración de esta tesis está dividida en cinco capítulos.

En el capítulo I se describe brevemente el problema de la ambigüedad sintáctica, la relevancia del problema dentro de la lingüística computacional, y se definen también los objetivos generales y las metas particulares que se persiguen al desarrollar esta tesis. También se mencionan las aportaciones de esta tesis.

En el capítulo II se presenta el marco teórico del procesamiento del lenguaje natural y de la ambigüedad sintáctica; los problemas inherentes al procesamiento del lenguaje natural y los enfoques que se utilizan para resolver la ambigüedad sintáctica.

Se describe también el uso de los diccionarios en el procesamiento del lenguaje natural y se presentan brevemente ejemplos de algunos diccionarios usados actualmente.

Por último se describe brevemente el método aplicado [GALICIA, 00] para la

compilación de la base de datos de combinaciones de palabras en español.

En el capítulo III se describe el modelo matemático en el que se basa el método, que es el presentado en [GALICIA, 00]; también el algoritmo para la asignación de pesos a las combinaciones, y el algoritmo de la obtención de árboles de dependencias, estos dos son tomados del método de referencia. Se describe también el algoritmo de obtención de combinaciones de palabras, que es nuestra aportación al método descrito.

En el capítulo IV se describe la metodología utilizada para llevar a cabo los experimentos, y los resultados obtenidos; y por último en el capítulo V se presentan las conclusiones y el trabajo futuro.

CAPÍTULO 2 PROCESAMIENTO DEL LENGUAJE NATURAL

En este capítulo se describe el marco teórico del procesamiento del lenguaje natural y las herramientas utilizadas en esta tesis, también algunos trabajos relacionados.

2.1 Antecedentes y aplicaciones del procesamiento del lenguaje natural

La comunicación en sus diversas formas y manifestaciones es la característica fundamental para que una especie sobreviva. La comunicación no siempre es explícita. La comunicación tiene múltiples niveles, en los que el medio por el cual se transmite el mensaje es el distintivo. Los sistemas desarrollados en estos niveles de comunicación se denominan procesos comunicativos. Entre estos procesos comunicativos el habla es el que más complejidad ha alcanzado y es en el que se centra en nuestro objeto de estudio: la lengua, o en el caso del procesamiento del lenguaje natural, el lenguaje.

Los lenguajes formales, se distinguen del lenguaje natural en que fueron desarrollados artificialmente para un fin específico, como el lenguaje matemático, el lenguaje ensamblador, el lenguaje de la lógica, los lenguajes de programación, el lenguaje de consultas estructurado o *Structured Query Language*, etcétera. Como ejemplo, daremos la definición de lenguaje de [LEWIS,98]: “Cualquier conjunto de cadenas sobre un alfabeto Σ –esto es, cualquier subconjunto Σ^*- será llamado lenguaje”.

En contraste, [CRISTAL, 91] define al lenguaje como “el uso convencional y sistemático de sonidos, signos o símbolos escritos en una sociedad humana para

la comunicación".

Con lenguaje natural nos referimos al lenguaje humano. Este tiene una complejidad mayor a los lenguajes formales, y su representación se basa en las teorías de la lingüística teórica.

El PLN o procesamiento del lenguaje natural se ocupa del desarrollo de herramientas computacionales en las cuales los datos de entrada o salida son o serán textos en lenguaje natural.

El desarrollo del PLN inicia en la década de los 40's, en la que se construyó el primer traductor automático. A partir de entonces, se desarrollaron varias herramientas como SHRLDU, desarrollado por Terry Winograd en el MIT entre 1968 y 1970, que trabajaba con "bloques de palabras" con vocabularios restringidos o delimitados, que ayudaron a tener resultados bastante buenos, despertando el optimismo excesivo, mismo que terminó cuando los sistemas fueron extendidos a situaciones reales.

Con la experiencia, los desarrolladores e investigadores se dieron cuenta de que requerían de métodos más estructurados para el manejo del lenguaje natural, por lo que comenzaron a basarse en las teorías lingüísticas. Actualmente las propuestas de la teoría lingüística tienen frecuentemente su componente computacional y viceversa.

También a medida que se desarrollaron más aplicaciones de PLN, se observó que se requería el manejo de una gran cantidad de conocimiento de diversa índole, y que eran aplicables las técnicas desarrolladas en la inteligencia artificial. A partir de entonces, estas dos disciplinas se han retroalimentado mutuamente, por lo que muchos autores consideran al PLN y la rama que lo contiene, la lingüística computacional, como parte de la inteligencia artificial.

El procesamiento del lenguaje natural, según [CORTÉS, 93], no forma parte de la inteligencia artificial exclusivamente, sino que utiliza técnicas y formalismos de ella y también de la lingüística teórica y de la lingüística computacional, así como de otras disciplinas para llegar a sus fines, mismos que son la construcción de sistemas computacionales para la comprensión y la generación de textos en lenguaje natural.

Como mencionamos antes, el PLN y la lingüística computacional se alimentan de muchas áreas de conocimiento, relacionadas principalmente con la lingüística, por ejemplo la psicolingüística, que se encarga de estudiar los procesos de comprensión del lenguaje.

Sin embargo, el PLN y la lingüística computacional no intentan modelar los procesos que el cerebro humano lleva a cabo para la comprensión del lenguaje, sino aproximarse a sus resultados.

2.1.1 Principales aplicaciones del procesamiento del lenguaje natural

El PLN tiene diversas aplicaciones. Entre ellas están:

2.1.1.1 Reconocimiento del lenguaje hablado

Los sistemas de reconocimiento de voz son aquellos en los en que la entrada está constituida por mensajes de voz digitalizados [BOLSHAKOV, 04]. En estos sistemas se requiere analizar la parte física y acústica del mensaje audible y además se requiere el análisis en cada nivel lingüístico de la información procesada.

Los sistemas de reconocimiento de voz utilizan por tanto un decodificador acústico-fonético, para la información acústica, fonética, fonológica y léxica.

También utilizan módulos morfológico, sintáctico y semántico.

Una de las soluciones planteadas para resolver estos problemas es el uso de gramáticas restringidas. Esta técnica consiste en que el mensaje audible está delimitado en un subconjunto del lenguaje natural que consiste en respuestas cortas y sencillas a la solicitud de información específica.

La gramática se limita a afirmaciones, números, opciones, etc.

2.1.1.2 Traducción automática

Fue de las primeras tareas en el PLN. En 1946, Weaver y Both presentaron el primer sistema de traducción automática, seguido por el sistema GAT ("Georgetown Automatic Translator"), y ya en 1961 el CETA ("Centre d' etudes pour la Traduction Automatique") en Grenoble [CORTÉS, 93].

En sus inicios, se creía que la traducción sería una sustitución de términos o palabras en el idioma correspondiente. Sin embargo, con los avances en el área, los investigadores y desarrolladores se dieron cuenta de la complejidad que conlleva la traducción automática.

La traducción requiere de conocimiento morfológico, sintáctico y semántico. También se requiere de un corrector de estilo.

En la actualidad se tienen traductores especializados en una materia de estudio o con un contexto de lenguaje bien definido, por ejemplo un traductor para textos médicos será muy diferente de un traductor para crónicas deportivas.

2.1.1.3 Consultas en lenguaje natural

Según [BOLSHAKOV, 04], las consultas en lenguaje natural o *Natural Language Interface* a una base de datos, se dedican a la comprensión de preguntas introducidas por un usuario en lenguaje natural, pero algunas veces también se trata de salidas con un cierto formato. La información en una base de datos

normalmente se refiere a un solo tema. Esto es, tiene una cierta especialización, por tanto, la gramática requerida para el análisis lingüístico y en especial semántico es mucho más simple que en otras áreas de PLN.

En la mayoría de los casos, el principal factor para obtener buenos resultados en este tipo de sistemas, resultados que se traduzcan en datos correctos de acuerdo a la formulación de la consulta en lenguaje natural, es la especialización de la base de datos.

Respecto al lenguaje natural, menciona [BAEZA, 99], algunas aplicaciones cuyas interfaces están basadas en él, como el uso de algoritmos de posicionamiento estadístico o *Statistical Rank Algorithms*, con los cuales se construyen listas de documentos que contienen los términos de búsqueda expresados en lenguaje natural.

También distingue la respuesta a preguntas en lenguaje natural cuyo propósito y diseño es diferente a las consultas realizadas a sistemas manejadores de bases de datos, en cuanto a que no se tiene un esquema de bases de datos sino un documento a partir del cual se intentará contestar una pregunta como describe [JACOBS, 93]. Este tipo de aplicaciones también son objeto de estudio de la Recuperación de Información.

2.1.1.4 Recuperación de información

La recuperación de información se refiere a la representación, almacenamiento, organización y acceso a las unidades de información [BAEZA, 99]. Las populares herramientas de búsqueda Google, Lycos y Copernic son algunas de las aplicaciones más populares de la RI ó recuperación de información.

La RI es un área que involucra disciplinas como la lingüística, la biblioteconomía, la informática y el diseño de sistemas.

La RI y la recuperación de datos son campos distintos, opuestos entre sí en cuanto a que un lenguaje de recuperación de datos se utiliza para recuperar datos que satisfacen a una consulta. Esta consulta está escrita por medio de expresiones regulares o álgebra relacional y de existir un subconjunto que satisfaga el enunciado, todos los objetos que este devuelva tendrán un cien por ciento de pertenencia al mismo. Esto es, satisfacen por completo la demanda de información. No hay datos incorrectos [BAEZA, 99].

En cambio, en la RI el usuario busca información sobre el tema, más que datos. Los objetos que se obtienen pueden ser no del todo precisos, debido a que solo se verifica que la información contenida esté relacionada, aunque no sea exacta.

La RI requiere información sintáctica y semántica para "interpretar" la solicitud de información del usuario. Actualmente, se utiliza en indexación de textos, clasificación y categorización de documentos, arquitectura de sistemas, visualización de datos, filtrado de datos, entre otros.

2.1.1.5 Extracción de información

Según [BOLSHAKOV, 04], la extracción de información o *Extraction of Factual Data from Texts*, es la extracción automática de datos en una base de datos que contendrá campos o parámetros basados en textos en línea.

Una aplicación de la extracción de Información es el llenado de una base de datos estructurada a partir de textos en lenguaje natural.

2.1.1.6 Corrección automática de textos

En este tipo de aplicaciones podemos encontrar los siguientes subtipos: corrección ortográfica, corrección gramatical y corrección de estilo.

La corrección ortográfica se dedica corregir errores ortográficos en el texto.

[BOLSHAKOV, 04] menciona que este tipo de herramientas debería ayuda a corregir automáticamente los errores tipográficos en los textos, así como errores por deletreado incorrecto de las palabras, que llevan a combinaciones imposibles de palabras, por ejemplo: *La verdad os hará liebres* en lugar de: *La verdad os hará libres*.

Solo algunos de estos correctores ortográficos tienen esa capacidad.

En estos correctores se utiliza un mayor conocimiento lingüístico para realizar la corrección de errores ortográficos. Según [BOLSHAKOV, 04] existe una variante de correctores ortográficos con componente de combinaciones de palabras.

Este tipo de correctores ortográficos en inglés son llamados *spell checkers*, y trabajan con diccionarios de todas las palabras válidas para un lenguaje específico, lo cual es costoso en recursos de la computadora.

Los correctores ortográficos y gramaticales más eficientes y poderosos utilizan conocimiento morfológico detallado, el cual facilita la creación de diccionarios más compactos y manejables como en [CASTILLO, 03].

Los errores gramaticales son los que violan la estructura de la oración [BOLSHAKOV, 04] . Estos errores solo se corrigen por completo en el análisis sintáctico. Es por esto que la mayoría de los correctores de gramática, están bastante incompletos.

2.1.1.7 Generación automática de resúmenes

El propósito de estas herramientas es determinar el tema de un documento automáticamente, y se utilizan para clasificación de documentos, localizar documentos en Internet, indexar documentos, entre otros.

[CASTILLO, 03] menciona que existen diferentes variantes de la tarea de resumir.

Por ejemplo, se puede buscar la opinión más común sobre un tema. Una variante es el resumen temático de texto: presentar un breve informe sobre los temas (aunque no las ideas) que se discuten en un texto dado.

2.2 Tareas y problemas del procesamiento del lenguaje natural

El análisis lingüístico se divide en varios niveles, que se distinguen entre sí por la complejidad de su objeto de estudio.

Según [CORTÉS, 93], se suelen presentar los niveles de descripción en forma estratificada, comenzando por los más próximos al análisis superficial (voz, frases escritas) y acabando por los más próximos a las capacidades cognitivas de quien produce el lenguaje.

Estos niveles de análisis o llamados niveles de lenguaje son:

– Nivel fonético

Su objeto de estudio son los sonidos: sus características físicas como frecuencia, intensidad, modulación, etc.

– Nivel fonológico

En este nivel se estudian los sonidos en forma de voz. Sus unidades son los fonemas.

– Nivel léxico

Las palabras como unidades de significado. Sus unidades serían los lexemas.

– Nivel morfológico

En este nivel se estudian las palabras en cuanto a sus procesos y componentes:

la flexión, la derivación o la composición. Sus unidades son los morfemas.

– Nivel sintáctico

Aquí se estudia la forma en que las palabras se agrupan para formar frases.

[CORTÉS, 93] también reconoce los niveles lógico e ilocutivo dentro de los niveles de análisis lingüístico.

– Nivel lógico

Trata del significado literal de la frase (sin tomar en cuenta el contexto). En este nivel se estudia el concepto de forma lógica.

– Nivel semántico

Aquí se estudia el significado lógico de la frase dentro del contexto en que se utiliza.

– Nivel pragmático

Trata de la forma en que se usa el lenguaje en un contexto, esto es, que tipo de lenguaje se utiliza en una comida familiar, por ejemplo, contra el lenguaje utilizado en un juzgado.

– Nivel ilocutivo

En este nivel se estudian las intenciones del lenguaje. Se pueden estudiar también los actos del habla directos e indirectos, objetivos, intenciones, planificación de los diálogos, entre otros. Ejemplos clásicos de este nivel de análisis podrían tomarse de cursos de técnicas de negociación o cierre de ventas.

El procesamiento del lenguaje natural, cuya principal tarea es la comprensión del lenguaje natural se basa en la teoría lingüística para el desarrollo de herramientas.

Estas herramientas están orientadas a los niveles del lenguaje y para realizar el análisis tienen diferentes metodologías.

Una vez delimitados la unidad a analizar y el contexto en el que se va a llevar a cabo el análisis, tiene lugar el proceso de comprensión. Normalmente, el análisis se lleva a cabo con metodología de cascada, esto es, para cada nivel del lenguaje hay una herramienta que alimenta el siguiente nivel.

La ventaja de esta metodología es la independencia entre los niveles. El inconveniente es que con frecuencia un nivel requiere información del siguiente nivel para tener un análisis completo, el más claro ejemplo es la resolución de ambigüedad sintáctica por medio de información semántica.

Cada nivel de análisis tiene sus propias tareas y problemas, en las siguientes secciones se describen los más importantes.

2.2.1 Análisis léxico

De acuerdo a [CORTÉS, 93] El análisis léxico de la información es donde el sistema debe reconocer las palabras que forman a las frases y la información que en ella se deposita, sea morfológica, sintáctica o semántica.

La complejidad del análisis léxico radica en que por una parte, se tienen los problemas intrínsecos al propio léxico: segmentación e identificación de las palabras, homonimia, polisemia, desplazamientos de significado (por ejemplo en las metáforas), lexias (frases u oraciones), locuciones, etc.

Adicional a esto, se tienen problemas ligados al volumen de la información léxica necesaria: representación, redundancia, acceso eficiente, adquisición, etc.

El análisis léxico comienza por segmentar el texto en palabras. Si solo se requiriera identificar ortográficamente las palabras el proceso sería sencillo, sin

embargo existen palabras gramaticales que se componen de dos palabras por ejemplo: no obstante. También existen palabras ortográficas que contienen más de una palabra gramatical como dámelo = da me lo, del = de él.

El resultado del análisis léxico, independientemente de la representación, debe contener la siguiente información:

1. Categorización sintáctica: Es una etiqueta, de acuerdo al formalismo sintáctico utilizado, y a la gramática utilizada. Aquí se clasifican y etiquetan las unidades léxicas en categorías cerradas como preposiciones, determinantes, etc., y categorías abiertas como nombre, adjetivo, etc.

2. Propiedades sintácticas de concordancia como el género, número, persona, caso, etc.

3. Otras propiedades sintácticas como las restricciones selectivas, por ejemplo el tipo de argumentos que un verbo admite.

4. La información morfológica, como el patrón de formación de la palabra.

5. La información semántica, como la categoría semántica, la forma lógica asociada, los rasgos semánticos, etc.

2.2.1.1 Representación de la información léxica

Uno de los aspectos importantes en el análisis léxico es la construcción de diccionarios o lexicones. Las características funcionales y operativas del diccionario son, según [CORTÉS, 93]: volumen, diccionario de palabras, lexemas o frases, tipo de información asociada, características de expansión: herencia, reglas, morfología, etc.

Para organizar la información en memoria es común el uso de árboles binarios con

mecanismos de búsqueda o BTREES.

También en la actualidad es común el uso de estructuras de rasgos o matrices de rasgos [CORTÉS, 93], que fueron introducidas por Schieber en 1983, y son listas de atributos a las que se asocian valores. Estos valores pueden ser atómicos o nuevamente estructuras de rasgos. También se conocen como DAGs o *Directed Acyclic Graphs*.

Un ejemplo de un DAG, sería:

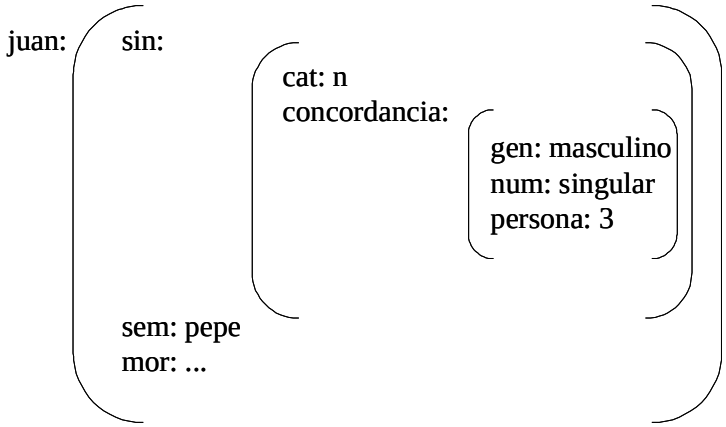

Figura 1. Directed Acyclic Graphs.

En la Figura 1. *Directed Acyclic Graphs*, se observa que la entrada, o información sintáctica, tiene como valores la categoría sintáctica (atributo atómico) y la concordancia (una FS o matriz de rasgos), que a su vez tiene como valor, tres atributos atómicos.

Estas estructuras que representan conocimiento léxico son de vital importancia para el análisis sintáctico.

Según [SCHIEBER, 89], las estructuras de rasgos pueden ser consideradas como estructuras de grafos con una raíz, orientados y acíclicos, cuyos arcos llevan como identificadores los nombres de rasgos. Cada arco va dirigido a otro DAG o símbolo atómico.

Se utiliza la teoría de grafos en esta representación debido a que la teoría de grafos ofrece un vocabulario simple y matemáticamente bien definido en el cual se pueden representar las estructuras lingüísticas.

En los grafos, según [SCHIEBER, 89], se pueden definir formalmente conceptos como la unificación, la generalización, la disyunción, la negación, la sobreescritura y otras operaciones.

La unificación, o el concepto más relevante en este formalismo, se refiere a la combinación de los pares rasgo / valor, y la combinación recursiva de estos valores, siempre y cuando existan valores que satisfagan ambos conjuntos.

Las reglas utilizadas para construir esta asociación, deben describir:

- Como las cadenas se concatenan para formar cadenas más largas.
- Como se relacionan las estructuras de rasgos asociadas a ellas.

[SCHIEBER, 89] utiliza estas reglas de combinación, como reglas de una Gramática Libre de Contexto, de la forma (ejemplo):

$S \rightarrow SN\ \ SV$

<S núcleo> = <SV núcleo>

<S núcleo sujeto> = <SN núcleo>

El nombre de un constituyente representa el valor del rasgo *categoría* para este

constituyente (como S, SN o SV).

Un ejemplo de *unificación* de valores atómicos con su categoría, usando la gramática presentada, sería:

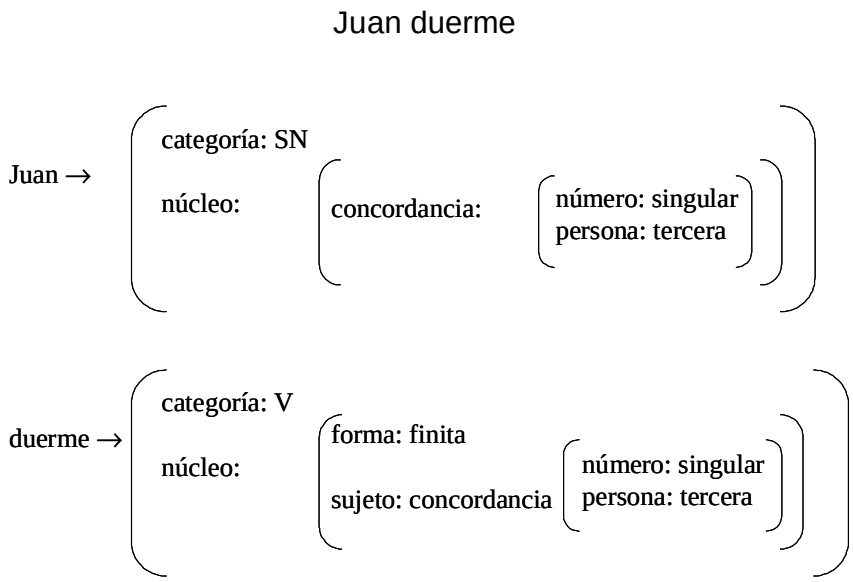

Figura 2. Ejemplo de unificación usando DAGs.

La concordancia del número y el género en ambos DAGs, significa que la oración es gramaticalmente correcta.

2.2.2 Análisis morfológico

En lenguas aglutinantes, flexivas o derivativas, el análisis morfológico es de gran importancia y también tiene un alto grado de complejidad.

En el análisis morfológico, se analizan los mecanismos de formación de las palabras. Las palabras se forman por concatenación o composición de formas más simples, conocidas como *rasgos*. Otro mecanismo de formación de palabras es la adjunción, donde la construcción consiste en que a una raíz se le adjuntan

uno o varios afijos. Estos afijos pueden ser prefijos, infijos o sufijos.

El mecanismo básico de cualquier herramienta de análisis morfológico, es la descomposición de una palabra en un conjunto de morfemas. También se trata de obtener el lexema asociado a la forma léxica para, a través de él, acceder a la información semántica.

Existen analizadores morfológicos de un nivel y de dos niveles. Los AM o analizadores morfológicos de un nivel trabajan a nivel superficial. En estos casos solo se establecen reglas válidas para la concatenación de morfemas.

Los AM de dos niveles funcionan como transductores de estado finito, y tienen un nivel de entrada y uno de salida. La entrada es la palabra que se analiza y la salida es el lexema.

2.2.3 Análisis sintáctico

El análisis sintáctico tiene como objetivo determinar si una frase es correcta, y proporcionar una estructura de la frase que refleje sus relaciones sintácticas y que pueda ser usada en los tratamientos posteriores.

En la sección 2.3 se describe ampliamente los detalles del análisis sintáctico.

2.2.4 Análisis semántico

El análisis semántico tiene por objeto el estudio del significado de las frases. Según [CORTÉS, 93], la interpretación semántica es el proceso de extracción de dicha información. Según [ALLEN, 95] para el PLN, son necesarias ciertas características en el proceso de interpretación semántica, como:

– La IS debe ser compositiva. Esto es, la representación semántica de una frase debe poder formarse a partir de la representación semántica de sus

componentes.

– La IS debe respaldarse en la lingüística teórica

– Se debe definir una representación semántica, para que la IS genere objetos semánticos.

– Debe existir una interfaz entre la sintaxis y la semántica.

– La IS debe ser capaz de tratar fenómenos complejos como la cuantificación, la predicación, negación, etc.

– La IS debe ayudar a resolver la ambigüedad léxica y sintáctica. La representación debe ser no ambigua.

– El sistema de representación debe soportar inferencias (herencia, conocimiento no explícito).

La forma de representación más común de la IS es la lógica en sus formas diversas, principalmente cálculo de predicados de primer orden.

Esto es, si se pretende construir una IS compositiva, se puede definir el nivel atómico, es decir, el que no admite mayor descomposición. Un ejemplo de una representación basada en cálculo de predicados podría componerse de la siguiente forma:

– Los nombres propios serían constantes: “Pedro” → pedro

– Los verbos intransitivos podrían representarse como predicados unarios:

“ríe” → (lambda (x), reir(x))

Otra de las representaciones más utilizadas para la información semántica son las redes conceptuales. Según [GALICIA, 00] red semántica es un conjunto de

relaciones entre pares de palabras, o una combinación de palabras, refiriéndose a una cosa específica o idea.

Las redes semánticas tienen una fuerte fundamentación sicológica, se considera que en la mente humana los conceptos se encuentran relacionados entre sí, formando una red.

Los elementos que forman una red semántica son:

1. Estructuras de datos agrupados en *nodos*. Estos representan conceptos.
2. Un conjunto de procedimientos de inferencia que actúan sobre las estructuras de datos.

Existen distintos tipos de redes semánticas pero mencionaremos las más utilizadas debido a que son las que se usan en el modulo de proximidad semántica.

Las redes *IS-A* (es-un), son jerarquías taxonómicas cuyo núcleo está constituido por un sistema de enlaces de herencia entre los objetos o conceptos de representación de información conocidos como *nodos*. Estos enlaces están etiquetados por diferentes tipos de relaciones, que en su mayoría son especializaciones de las relaciones *IS-A*.

Las taxonomías utilizadas en la descripción del mundo real son el principal esquema de lo que una red semántica debe representar, esto es: conceptos más generales abarcan otros más detallados o específicos como : un perro es un cánido, un cánido es un mamífero, un mamífero es un animal.

Los nodos de las estructuras *IS-A* se han usado para representar muchas cosas, pero la división más importante es la interpretación genérica o específica de los nodos, es decir, si estos representan un solo individuo o varios. Los nodos

situados en lo más bajo de la jerarquía y que denotan individuos son llamados *tokens*, mientras que los nodos superiores, que denotan clases de individuos son llamados *types.*

Un ejemplo de una red semántica es, tomado de [GALICIA, 00]:

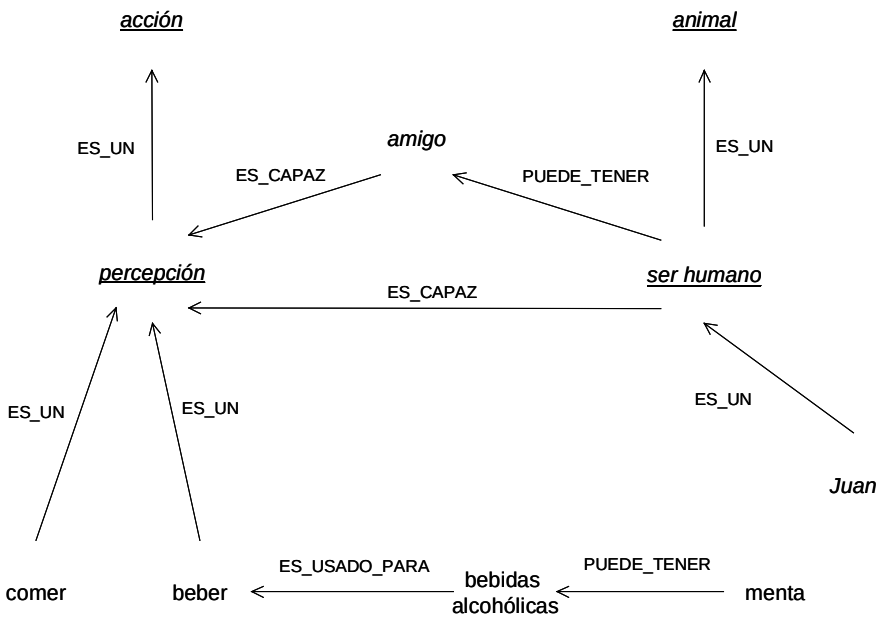

Figura 3. Red semántica de la frase *Juan bebe bebidas alcohólicas con sus amigos*.

Las redes semánticas se han utilizado para resolver cierta clase de ambigüedad. Por ejemplo: para obtener restricciones semánticas de ocurrencia concurrente para conjuntos de palabras relacionadas sintácticamente a partir de un corpus de textos.

2.2.5 Pragmática

La pragmática se asocia con el uso del lenguaje en un contexto, esto es, evalúa si la frase produce el efecto deseado. Se dedica al 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. La Pragmática toma en consideración los factores extralingüísticos que determinan el uso del lenguaje, esto es, todos aquellos factores a los que no se hace referencia en un análisis lingüístico.

2.2.6 Principales problemas en el PLN

En cada uno de los niveles de análisis se presentan dificultades o tareas por resolver, de acuerdo a su objeto de estudio.

La principal de estas tareas es la ambigüedad. En cada uno de los niveles del lenguaje, se presentan también problemas inherentes a sus procesos de extracción de información.

En el nivel léxico, el principal problema es la polisemia, o ambigüedad léxica. Esto se debe a que una misma palabra puede tener diferentes significados, y por tanto, la selección del significado apropiado se debe deducir a partir del contexto de la frase.

2.2.6.1 Polisemia (word sense disambiguation)

En el PLN, la desambiguación de sentido de la palabra se refiere a elegir en que sentido se usa una palabra, dentro de una frase. *Está sentado en el banco*, vs. *Ahí está el Banco Nacional*.

Uno de los principales problemas con el WSD es decidir que categoría sintáctica tiene cada uno de los sentidos de la palabra. En muchos casos el sentido de la palabra no es claro, por ejemplo en las metáforas.

Una solución que algunos investigadores han utilizado es delimitar el diccionario al

contexto de la aplicación.

Según [MANNING, 00], el problema de la desambiguación es de clara importancia debido a que, por ejemplo, en los sistemas de traducción automática, si se tradujeran ambas oraciones: *Está sentado en el banco*, vs. *Ahí está el Banco Nacional*, al alemán, en la primera, banco debería ser traducido como *ufer*, en cambio en la segunda, sería traducido como *bank.*

De igual forma, un sistema de recuperación de la Información que consulte documentos referentes a instituciones bancarias, debería arrojar solo aquellos que usen banco, en el sentido de la segunda oración.

Existe otro tipo de ambigüedad, que está relacionada con las partes de la oración, o en inglés *parts-of-speech* o *POS*. La gramática tradicional clasifica las palabras de acuerdo a la forma en que se usan, etiquetándolas de acuerdo a la parte de la oración correspondiente: verbo, sustantivo, artículo, adjetivo, adverbio, preposición, etcétera. A esto se refiere el POS. Esta se presenta cuando una palabra tiene varias categorías sintácticas como en: juego de pókar vs. (yo) juego tenis.

El etiquetar el uso de una palabra en términos de partes de la oración (*parts-of-speech*), es un proceso que se conoce como marcado de textos, o en inglés *tagging*.

El uso de una palabra como sustantivo en lugar de verbo, lleva claramente a una representación sintáctica diferente, lo que puede también verse como un problema del WSD.

Entre las principales formas de tratar el problema del WSD están la desambiguación supervisada, y la no supervisada, tal como la desambiguación basada en recursos léxicos, como diccionarios y tesauros.

En la desambiguación supervisada, hay un corpus sin ambigüedad disponible para aprendizaje automático o entrenamiento. También hay un conjunto de ejemplos donde cada ocurrencia de la palabra ambigua *w* está etiquetada con una etiqueta semántica correcta (con el sentido correcto de acuerdo al contexto s_k).

En contraste la desambiguación basada en diccionarios, como ejemplo de la desambiguación no supervisada, está el algoritmo de Lesk, que toma las definiciones $D_1,...,D_k$ en el diccionario, para los sentidos $s_1,...,s_k$ de la palabra ambigua *w*, que se encuentra en el conjunto de palabras que representan la frase, cuya representación es propuesta como una bolsa (*bag*) o una colección de objetos no ordenados que admite duplicados.

Por otra parte se tiene una bolsa E_{vj}, que es un conjunto de definiciones de sentidos, tomadas del diccionario, de *las palabras a las que se refieren los sentidos* $s_1,...,s_k$ *en cada una de las definiciones* $D_1,...,D_k$.

Esto es si $s_{j1},...s_{ji}$ son los sentidos de v_j que es el sentido que se dio a la palabra *w* en D_j, entonces E_{vj} será $\cup_{ji} D_{ji}$

El algoritmo de Lesk, que describe [MANNING, 00], es el siguiente:

comment: Given: context c
for all senses s_k of w **do**
    score(s_k) = overlap (D_k, U_{vj} in $_c$ E_{vj})
end
return s' s.t.s' = arg max s_k score(s_k)

Figura 4. Algoritmo para WSD no supervisado.

Tomemos las oraciones *está sentado en el banco*, y *ahí está el Banco Nacional*. Las definiciones de banco son las siguientes, tomadas del diccionario de la lengua

española de la Real Academia Española:

1. Asiento, con respaldo o sin él, en que pueden sentarse varias personas.
2. Establecimiento público de crédito, constituido en sociedad por acciones.

Digamos que la información referente a los sentidos de las entradas D_1 y D_2 antes listadas, serían:

Tabla 1. Sentido de los términos en E_{vj} para el algoritmo de Lesk.

Sentido		Definición
s_1	asiento	Mueble para sentarse.
s_2	acción	Título crediticio de participación financiera de una empresa.

De acuerdo al algoritmo de Lesk, tendríamos que de acuerdo al contexto, el puntaje sería:

Tabla 2. Desambiguación con el algoritmo de Lesk.

Puntaje		Contexto
s_1	s_2	
1	0	*Está sentado en el banco.*
0	1	*Ahí está el Banco Nacional.*

Este algoritmo es útil cuando las categorías semánticas de los sentidos de la palabra no son muy cercanas.

A nivel sintáctico, tenemos el problema de la ambigüedad sintáctica que está muy relacionada con el WSD, pero aquí lo que se desea desambiguar son las diferentes representaciones que el análisis sintáctico puede arrojar.

2.2.6.2 Ambigüedad

La ambigüedad en el PLN, como ya hemos mencionado antes, surge debido a las interpretaciones que se tienen de los distintos objetos en cada nivel de análisis.

Enumeramos algunos ejemplos breves tomados de [CORTÉS, 93] para delinear la idea de ambigüedad:

Ambigüedad léxica:

1. Se sentó en el banco.
2. Entró al banco y fue a la ventanilla.
3. El avión localizó el banco y comunicó su posición.

Banco en (1) se refiere a un mueble que sirve para sentarse, en (2) se refiere a una oficina de una entidad financiera, que realiza operaciones a través de una ventanilla y (3) se refiere, tal vez, a un banco de pesca.

Ambigüedad sintáctica:

4. La vendedora de periódicos del barrio

¿La vendedora es del barrio o los periódicos son del barrio?

Ambigüedad semántica:

5. Pedro dio un pastel a los niños

¿Un pastel a cada niño o un pastel que dividió entre todos los niños?

2.2.6.3 Anáfora

Según [ALLEN, 95], existen dos formas principales de referencia a frases nominales. Una referencia anafórica involucra una frase nominal que hace referencia a un objeto mencionado previamente, o en una oración anterior.

Por otra parte, en una referencia no anafórica se identifica un objeto que no ha sido mencionado previamente.

Ejemplos de anáfora son:

– Juan tomó el libro azul$_i$, y lo$_i$ vendió.

– Juan$_i$ tomó el libro azul, y se fue$_i$.

Las frases subrayadas se refieren al mismo objeto. Aquí también hay referencias no anafóricas, en ambas oraciones Juan es un sujeto no mencionado previamente.

Entre las referencias y la anáfora, la resolución cambia de acuerdo al tipo y la complejidad:

– Referencia indefinida: Introduce nuevos objetos en el contexto. Es simple de representar, ya que se puede crear un nuevo objeto del tipo apropiado y referenciarlo en forma lógica: por ejemplo la frase nominal *perro*, podría representarse como (INDEF/SING P1 PERRO) y mapear el objeto con un identificador único como PERRO01.

– Referencia definida: Menciona objetos ya existentes o previamente

mencionados. Las referencias definidas no anafóricas son más complicadas de representar debido a que la referencia debe ser una constante que ya existe en la base de conocimiento. Típicamente los sistemas manejan los nombres propios como *Juan*, asignándolos directamente a constantes en la base de conocimiento en una simple tabla de búsqueda.

Otros problemas del PLN son: ambigüedad de marcaje de textos (*POS tagging*), ambigüedad sintáctica, ambigüedad semántica, elipsis, ambigüedad morfológica, detección de colocaciones, límite de las oraciones, diferencias estructurales entre lenguajes, etcétera.

Los métodos tradicionales que utiliza el PLN, son:

– Basados en reglas (simbólicas o lingüísticas):

  - La descripción de todas las reglas en un formalismo, el uso de diccionarios grandes genera aplicaciones complejas en su construcción.

  - Las aplicaciones típicas son analizadores morfológicos, herramientas para reconocer nombres propios y búsqueda de colocaciones.

– Basados en métodos estadísticos:

  - Usando frecuencias de palabras absolutas o relativas, co-ocurrencia de palabras, etc.

  - Las aplicaciones típicas son sistemas de generación automática de resúmenes basados en bigramas de frecuencia.

  - Los sistemas frecuentemente son entrenados con textos manualmente codificados.

– Métodos híbridos que combinan reglas y métodos estadísticos

  - Se usan en marcaje de textos, lematización, generación de índices basados en términos o similitudes entre documentos.

  - Un uso clásico de estos métodos es la desambiguación léxica y sintáctica.

2.3 Análisis sintáctico

Lo que se pretende en esta parte del análisis es determinar si una frase pertenece o no al lenguaje que se trata de analizar [CORTÉS, 93]. Si los elementos de la frase son palabras, podemos decir que una frase está constituida por una cadena de palabras, es decir, $w \in V^*$, donde V denota al vocabulario terminal de la gramática o lo que es lo mismo, al conjunto de palabras válidas. Entonces, lenguaje será el conjunto L de cadenas válidas.

2.3.1 Procedimientos de reconocimiento sintáctico

Los primeros formalismos para análisis sintáctico utilizados fueron las redes de transición [CORTÉS, 93]. Cada red consta de una serie de nodos y una serie de arcos. El nodo origen se señala por medio de una flecha, los nodos finales con un doble círculo. Los nodos representan estados y los arcos, transiciones entre los estados.

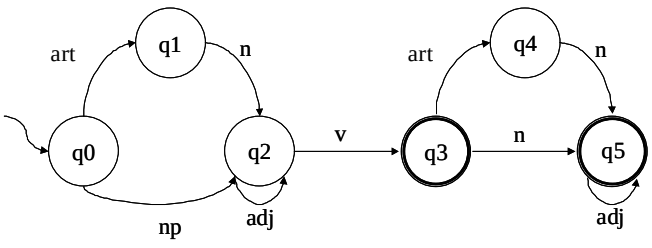

Figura 5. Redes de transición.

Las transiciones se realizan de acuerdo a la categoría de la palabra, que debe coincidir con la etiqueta de los arcos (elementos del vocabulario terminal de la gramática).

El proceso de reconocimiento comienza posicionándose en el estado de inicio, y tomando como entrada la primera palabra. El proceso continúa, realizándose transiciones válidas entre estados y desplazándose paralelamente la ventana sobre la cadena de entrada.

Si al consumir completamente la cadena estamos en un estado final, entonces la frase es correcta.

Un avance a las redes de transición fueron las RTN o redes de transición recursivas, donde cada estado en la red de inicio supone el acceso a una subred que en un momento dado puede posicionarse en la red de inicio. De ahí su nombre.

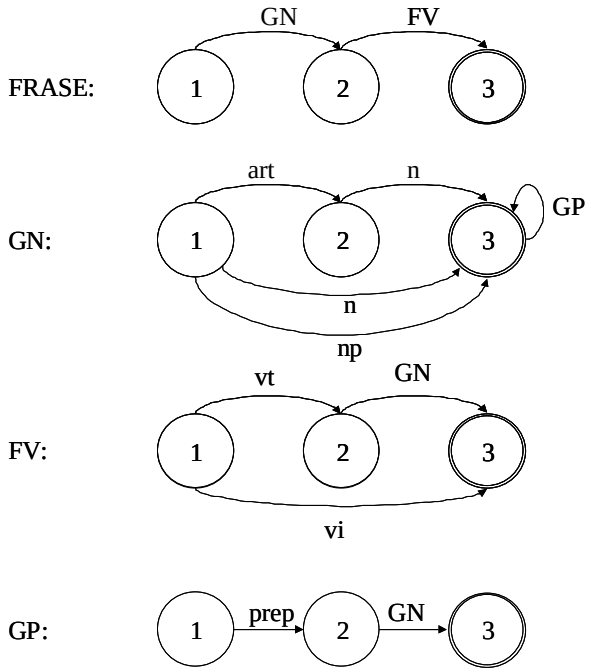

Figura 6. Redes de transición recursivas.

En la red de transición recursiva mostrada en la figura 6, la primera red analiza frases, la segunda grupos nominales, la tercera frases verbales y la cuarta grupos preposicionales.

Las llamadas recursivas a un ejemplo como: “el gato de Juan come atún”, serían:

Tabla 3. Derivación de la frase “el gato de Juan come atún” usando una red de transición recursiva.

Posición	Red	Estado	Salida	Etiqueta
El gato de Juan come atún	FRASE	1	-	-
El gato de Juan come atún	GN	1	FRASE:2	GN
gato de Juan come atún	GN	2	FRASE:2	det
de Juan come atún	GN	3	FRASE:2	N
de Juan come atún	GP	1	GN:3	GP
Juan come atún	GP	2	GN:3	prep
Juan come atún	GN	1	GP:3	GN
come atún	GN	3	GP:3	np
come atún	GP	3	GN:3	fin GN
come atún	GN	3	FRASE:2	fin GP
come atún	FRASE	2	-	fin GN
come atún	FV	1	FRASE:3	FV
atún	FV	2	FRASE:3	Vt
atún	GN	1	FV:3	GN
-	GN	3	FV:3	N
-	FV	3	FRASE:3	fin GN
-	FRASE	3	-	fin FV

Cuando se alcanza un estado final, con la última palabra de la cadena, podemos

afirmar que la frase es correcta.

La principal limitante de las RTNs o redes de transición recursivas fue la ambigüedad. Una palabra puede pertenecer a más de una categoría sintáctica: *el* juego de pókar vs. (yo) Juego tenis.

Posteriormente se introdujeron RTNs que permitían filtros al realizar transiciones, esto es, se etiquetaban los arcos con condiciones. Aún así estas herramientas son limitadas en el análisis sintáctico.

2.3.2 Gramática

La gramática como área de estudio tiene por objeto la lengua, su estructura y significado. Para la lingüística computacional, el concepto de gramática o de gramáticas de estructuras sintagmáticas es muy relevante en cada etapa del análisis.

Según [CORTÉS, 93], una gramática G es una tupla de 4 elementos:

$$G = <N, T, P, S>$$

donde

- N es el vocabulario no terminal, esto es, el grupo de elementos no terminales de la gramática
- T es el vocabulario terminal, es decir, el conjunto de elementos terminales de la gramática
- S, que pertenece a N es el símbolo de inicio
- P es el conjunto de reglas de producción de la gramática

De acuerdo a [CHARNIAK, 93], el conocimiento sintáctico se construye mediante

gramáticas, que son especificaciones de las estructuras permitidas en un lenguaje. El tipo más común de gramáticas utilizadas son las gramáticas libres de contexto (CFG por sus siglas en inglés), que consisten de:

- Un conjunto de *símbolos terminales*, que son los símbolos que aparecen al final de las cadenas (las palabras y los signos de puntuación)
- Un conjunto de *símbolos no terminales*, que son símbolos que son expandidos dentro de otros símbolos (partes del habla o *parts of speech* como frases nominales *fs*, frases verbales *fv*, oración *o*, frases preposicionales *fp,* etc.)
- Un símbolo no terminal específico que es el símbolo de inicio
- Un conjunto de reglas de escritura, cada una de las cuales tiene un no-terminal en el lado izquierdo y uno o más símbolos terminales o no-terminales del lado derecho

Tabla 4. Ejemplo de símbolos terminales para una CFG.

No-terminales	Ejemplos
Oración-principal (op)	“Juan viene hacia el auto.”
Oración (o)	“Juan viene hacia el auto”
Frase Verbal (fv)	“viene hacia el auto”
Frase Nominal (fn)	“Juan”
Frase Preposicional (fp)	“hacia el auto”

Entonces una gramática para generar este tipo de frases, podría representarse como:

op → o sp
o → fn fv
fv → v fp
fp → p fn
fn → art sust
v → {*verbos del español*}
p → {*preposiciones del español*}
sp → {*signos de puntuación en español*}
art → {*artículos del español*}
sust→ {*sustantivos del español, nombres propios*}

Una derivación a partir de nuestra gramática libre de contexto (CFG) sería:

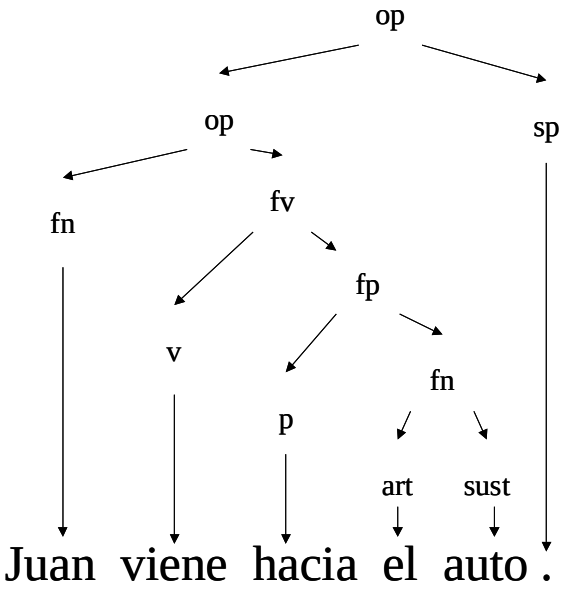

Figura 7. Ejemplo de una derivación de una CFG.

Las gramáticas libres de contexto son las herramientas básicas para el análisis sintáctico.

2.3.3 Analizadores sintácticos básicos

Según [CORTÉS, 93] existen dos resultados básicos que se espera obtener de cualquier analizador sintáctico: la estructura sintáctica y la estructura lógica o semántica básica.

Normalmente la representación de la información sintáctica es el árbol de derivación o árbol de análisis. Este nos muestra la estructura sintagmática o de componentes de la oración que analizamos.

Los analizadores sintácticos por excelencia son las gramáticas libres de contexto. De acuerdo a [WINOGRAD, 83] un analizador sintáctico o *parser*, utiliza un algoritmo de análisis junto con una gramática y un diccionario para producir un árbol de estructura de frase que corresponde a una oración. Hay una correspondencia directa entre las reglas de la gramática y la estructura que el analizador sintáctico asigna.

Un problema que se presenta al diseñar un analizador sintáctico basado en una CFG, es el manejo de la ambigüedad. Es por ello que comúnmente se utilizan las CFGs junto con otras herramientas para el análisis sintáctico.

2.3.4 Técnicas de análisis sintáctico

Podemos describir el proceso básico de análisis sintáctico (en el ámbito del PLN) como: “La generación de una representación de la estructura sintáctica de una frase, a partir de la derivación de la misma en base a una gramática libre de contexto.”

Esto significa que la parte más importante del proceso de análisis sintáctico es el

obtener una conjunto de estructuras que representen la frase en alguna forma convenida, describiendo las categorías sintácticas de las palabras que la forman y describiendo la forma en que se relacionan entre sí.

Se especifica que será un conjunto de estructuras debido a que, por el problema de la ambigüedad sintáctica, una frase puede tener más de una derivación.

En el proceso de análisis sintáctico se deben tomar en cuenta los siguientes puntos:

– Estrategia del análisis

Las más comunes son la descendente (dirigida por objetivos, o *top-down*) y la ascendente o dirigida por hechos o *bottom-up*.

– Dirección del análisis

El enfoque más común es de izquierda a derecha. También se utiliza con frecuencia el enfoque de los analizadores activados por islas en los que una palabra activa el proceso ascendente en forma de capas alrededor de dicha isla.

Un enfoque que sobresale es el *head driven*, en el cual se inicia el análisis a partir del núcleo o cabeza (*head*) de cada parte de la frase (verbo para la frase, núcleo nominal para el grupo nominal, etc.).

– Orden de aplicación de las reglas

Debido a que las reglas de una gramática llevan hacia los terminales, la forma en que se escriba o se apliquen es irrelevante en la mayoría de los casos. Dado el problema de la ambigüedad sintáctica, que se describe ampliamente en la siguiente sección, se tendrá con frecuencia más de una representación, esto es, más de una derivación aplicando diferentes reglas de la gramática a una misma

frase. En algunos analizadores sintácticos, se ponderan las reglas a aplicar.

– La ambigüedad

La ambigüedad sintáctica se presenta debido a que hay palabras que pertenecen a más de una categoría sintáctica, y debido a que una frase puede tener más de una representación sintáctica, esto es, de acuerdo a la gramática, es posible obtener más de dos árboles sintácticos correctos para la misma frase.

La resolución de la ambigüedad sintáctica es un proceso posterior al análisis y generalmente se asignan probabilidades a los árboles de representación, para determinar la elegibilidad de un árbol sintáctico respecto a los demás en el conjunto obtenido.

– No determinismo

El análisis sintáctico presenta varias características de no determinismo. Es necesario modelar analizadores que puedan manejarlo, por ejemplo con *backtracking* o procesamiento en paralelo.

2.3.5 Representación de la información sintáctica

De acuerdo a [ALLEN, 95], la representación de la estructura sintáctica expone la forma en que las palabras se relacionan unas con otras. Esta estructura indica la forma en que las palabras se agrupan en frases, que palabras modifican a otras palabras y que palabras son de central importancia en la oración.

Un proceso de análisis sintáctico extrae las propiedades estructurales de las oraciones y produce una representación sintáctica que asigna un nombre estructural a cada grupo principal de palabras.

Una de las representaciones propuestas por [ALLEN, 95] identifica los valores

estructurales de cada grupo:

(S **SUBJ** (NP **NAME** Jhon
NUM {3s})
MAIN sold
TENSE {PAST}
VOICE {ACTIVE}
OBJ (NP **DET** the
HEAD book
NUM {3s})
MODS (PP **PREP** to
POBJ (NP **NAME** Mary
NUM {3s}))

2.3.6 Ejemplo de la representación de un *chart*

Según [WINOGRAD, 83] una de las ineficiencias de los analizadores sintácticos basados en CFGs, es la necesidad de un estrategia de *backtracking* para generar variantes de árboles sintácticos de una frase. El autor menciona que este tipo de procesos son altamente ineficientes porque las estructuras se desechan para después volver a generarlas.

Para solucionar este problema, el autor describe los *well-formed substring tables* o *charts*, que guardan el registro de constituyentes que fueron construidos previamente y que pueden ser usados por otras reglas.

Un *chart* puede ser visualizado como una red de vértices, representando puntos en la oración, unidos por aristas representando a los constituyentes. Cada arista nombra al constituyente en el que inicia, y termina en el vértice que conecta. Un *chart* puede iniciar conteniendo solo las aristas correspondientes a la palabra

individual y su categoría léxica:

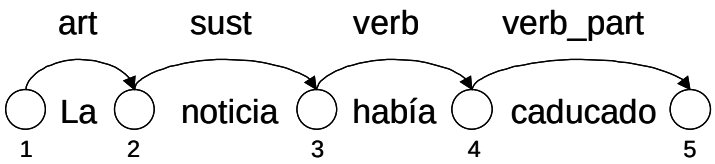

Figura 8. Un *chart* inicializado para análisis sintáctico.

Cuando se encuentra un constituyente, este se añade al *chart*. El analizador sintáctico usa el *chart* introduciendo un nuevo constituyente siempre que una regla aplique exitosamente y busca más constituyentes siempre que haya un no-terminal al inicio del remanente.

Si existe una arista para el símbolo correspondiente al punto actual en el *chart*, el analizador puede avanzar hacia el punto donde termina esa arista sin repetir el análisis. Así se ahorra una gran cantidad del tiempo empleado en un analizador basado en una CFG con enfoque *top-down* o descendente.

Por ejemplo, teniendo la siguiente gramática:

S → NP VP	NP → NP2
NP2 → NP3 PREPS	PREPS → PP PREPS
S → NP VP PREPS	NP2 → sustantivo
NP3 → sustantivo	PP → preposición NP
NP → art NP2	NP2 → NP2 adjetivo
PREPS → PP	VP → verbo

La derivación de la frase *la niña con vestido rojo juega con su amiga* sería:

S -> NP VP PREPS	NP2 -> NP3 PREPS
NV -> art NP2	NP3 -> sustantivo
artículo -> la	sustantivo -> niña

PREPS -> PP
PP -> preposición NP
preposición -> con
NP -> NP2
NP2 -> NP2 adj
NP2 -> sustantivo
sustantivo -> vestido
adj -> rojo
VP -> verbo
verbo -> juega
PREPS -> PP
PP -> preposición NP
preposición -> con
NP -> art NP2
art -> su
NP2 -> sustantivo
sustantivo -> amiga

Debido al orden de las reglas en la gramática, un analizador sintáctico basado en una CFG, aplicaría primero la regla S -> NP VP.

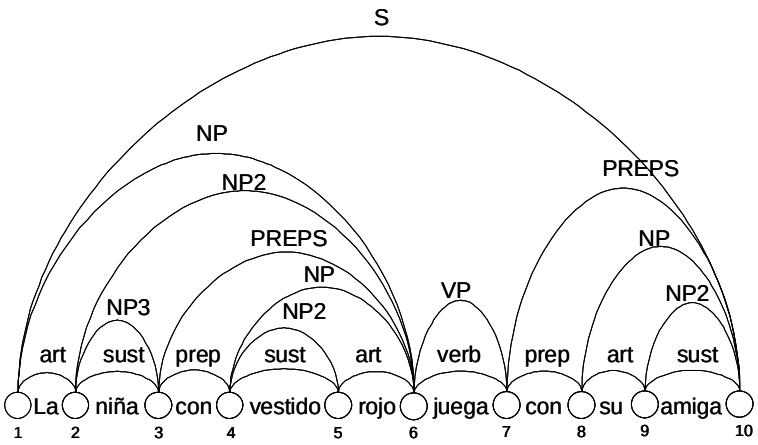

Figura 9. *Chart* de la frase *la niña con vestido rojo juega con su amiga*.

Utilizando *charts*, se busca una arista para el símbolo correspondiente al punto actual en el *chart*, y si existe, el analizador avanza hacia el punto donde termina esa arista sin repetir el análisis.

Para este trabajo de tesis, se utilizó la herramienta PARSER del Laboratorio de Lenguaje Natural del CIC, para generar los árboles sintácticos y representarlos como *charts*.

Un ejemplo de un *chart* generado por el PARSER es:

```
S -> @:CLAUSIN $PERIOD
   CLAUSIN ->  (&subj) NP(SG,FEM,3PRS) @:VP_SV(SG,3PRS,MEAN)
      NP(SG,FEM,3PRS) ->  (&det) ART(SG,FEM) @:N(SG,FEM,3PRS)
         ART(SG,FEM) -> <*TDFS0> ( La: la, 0/0)
         N(SG,FEM,3PRS) -> <*NCFS000> ( noticia: noticia, 1/0)
      VP_SV(SG,3PRS,MEAN) -> #*$$haber# @:PART(SG,MASC)
         #*$$haber# -> <*$$haber> ( había: haber, 2/0)
         PART(SG,MASC) -> <*VMPP0SM> ( caducado: caducar, 3/1)
   $PERIOD -> <*Fp> ( .: ., 4/0)
```

En esta herramienta las reglas tienen la siguiente estructura:

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

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

Las reglas de la gramática que utiliza el PARSER, se pueden consultar en Anexo A. Gramática generativa usada.

El *chart* presentado generado por el PARSER, puede verse gráficamente en la figura 10.

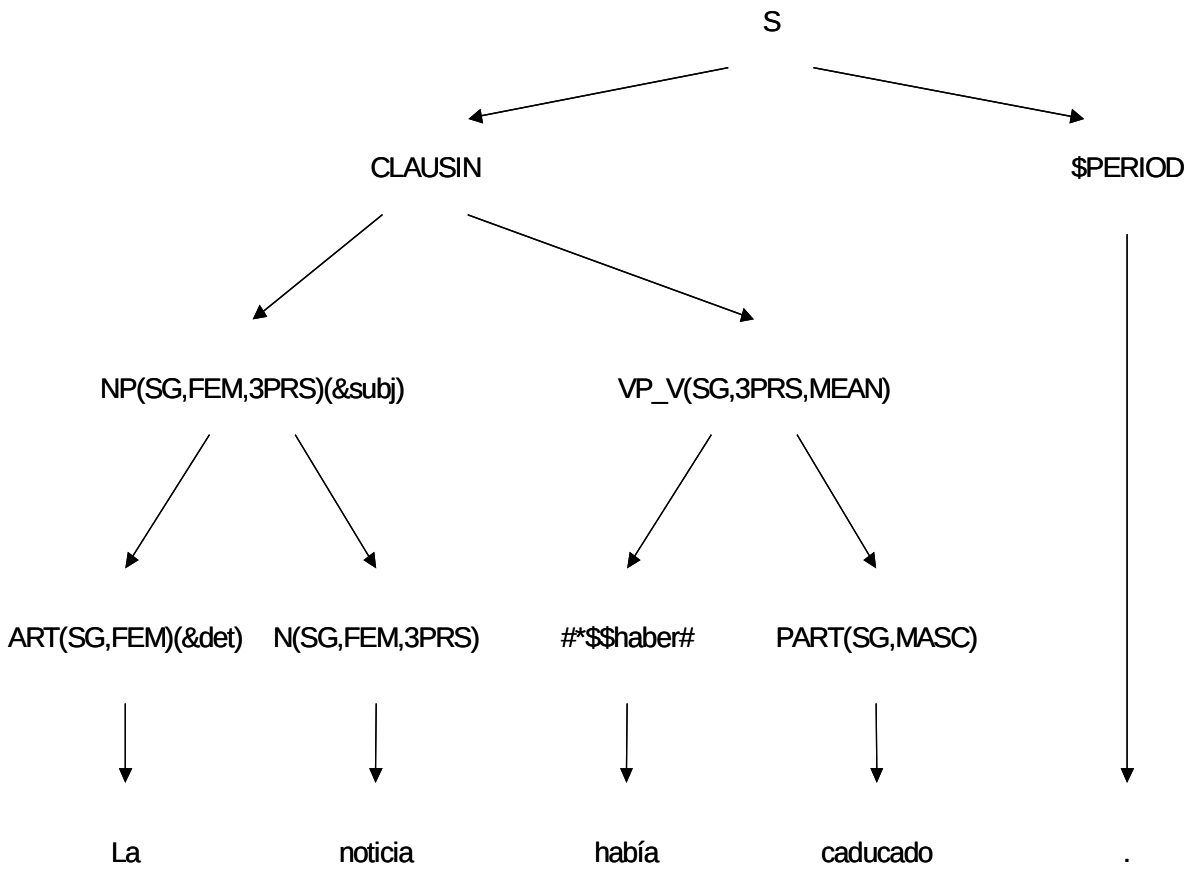

Figura 10. Representación gráfica del árbol sintáctico generado por el PARSER.

Entonces la indentación en el árbol sintáctico generado por el PARSER, representa los niveles en orden descendente y de izquierda a derecha.

Existen otras representaciones de la información sintáctica; sin embargo, se utilizan principalmente los *charts*, de estos se muestra un ejemplo por ser la representación utilizada por el PARSER.

2.4 Ambigüedad sintáctica

Según [FRANZ, 96], el análisis sintáctico es el proceso de recuperar la estructura de la entrada en el lenguaje natural.

Este autor, reconoce las siguientes estructuras lingüísticas:

Tabla 5. Estructuras lingüísticas que se obtienen en el análisis sintáctico.

Nivel de Análisis	Descripción
Morfología	Estructura de la palabras
Frase	Estructura de frases nominales, frases adjetivales, frases verbales, etcétera.
Cláusula	Estructuras de combinaciones de frases donde hay un verbo presente.
Oración	Estructura de combinaciones de cláusulas.
Modificadores	Aquí se agregan modificadores opcionales a las frases y cláusulas.

No solo las oraciones son objetos lingüísticos, también lo son las palabras, frases y cláusulas [FRANZ,96].

Durante el análisis sintáctico, se recupera la estructura sintáctica en cada nivel. En cada nivel se tienen diferentes reglas, tomadas de las regularidades que en el objeto de estudio de cada nivel se pueden observar.

Estas reglas, cuando son aplicadas en cada nivel de forma aislada, aplican para más de un caso, es por esta razón que se presenta la ambigüedad.

Según [ALLEN, 95], existen dos problemas relacionados a la ambigüedad en cada fase de análisis. El primero es el problema de la representación: considerar las diferentes representaciones posibles en un nivel dado. El segundo problema es la interpretación: la forma en que se producen las representaciones correctas en cada nivel.

A continuación se presentan algunos enfoques y herramientas probabilísticas para la desambiguación sintáctica.

2.4.1 Gramáticas libres de contexto probabilísticas

En una oración del tipo *la probabilidad con que se incendian los bosques depende de la velocidad del viento al momento de iniciarse el fuego*, el verbo *depende* coincide en número con *velocidad* que es la cabeza de la frase nominal y no con el sustantivo que lo precede (*bosques*).

Para eliminar la ambigüedad en este tipo de frases, se usan las gramáticas libres de contexto probabilísticas ó PCFG por *Probabilistic Context Free Grammar* [MANNING, 00], que son gramáticas libre de contexto (CFG) con probabilidades añadidas a las reglas, que indican la elegibilidad de las variantes.

Una PCFG *G* consiste de:

- Un conjunto de terminales $\{w_k\}$, $k = 1,...,V$
- Un conjunto de no terminales, $\{N_i\}$, $i = 1,...,n$
- Un símbolo de inicio específico, N_1
- Un conjunto de reglas, $\{N_i \rightarrow \zeta_j\}$, donde ζ_j es una secuencia de

terminales y no terminales

- Un conjunto de probabilidades sobre las reglas tal que

$$\forall \text{i} \sum_j P(N^i - \varsigma^j) = 1$$

Para aclarar el ejemplo: $w_1 \ldots w_m$ es una secuencia de palabras que representa la oración que se va a analizar; w_{ab} es una subsecuencia $w_a \ldots w_b$, producida por el no terminal N_j.

Entonces, con la gramática probabilística:

O $\rightarrow$ FN FV	1.0
FP $\rightarrow$ P FN	1.0
FV $\rightarrow$ V FN	0.7
FV $\rightarrow$ FV FP	0.3
P $\rightarrow$ *con*	1.0
V $\rightarrow$ *examinan*	1.0
FN $\rightarrow$ FN FP	0.4
FN $\rightarrow$ *médicos*	0.1
FN $\rightarrow$ *pacientes*	0.4
FN $\rightarrow$ *influenza*	0.1

Considerando la gramática descrita, se tienen dos posibles árboles para la frase *médicos examinan pacientes con influenza*, junto con las probabilidades asignadas, dadas las reglas.

.

En el primer árbol se aplica una derivación que conduce a tener como una frase preposicional, *pacientes con influenza.*

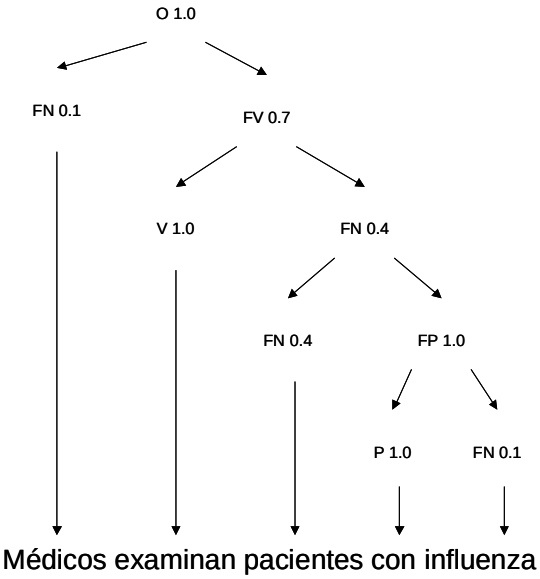

Figura 11. Árbol sintáctico t_1 de la frase *médicos examinan pacientes con influenza*

Tenemos la derivación:

O $\rightarrow$ FN FV	1.0
FN $\rightarrow$ *médicos*	0.1
FV $\rightarrow$ V FN	0.7
V $\rightarrow$ *examinan*	1.0
FN $\rightarrow$ FN FP	0.4
FN $\rightarrow$ *pacientes*	0.4
FP $\rightarrow$ P FN	1.0
P $\rightarrow$ *con*	1.0
FN $\rightarrow$ *influenza*	0.1

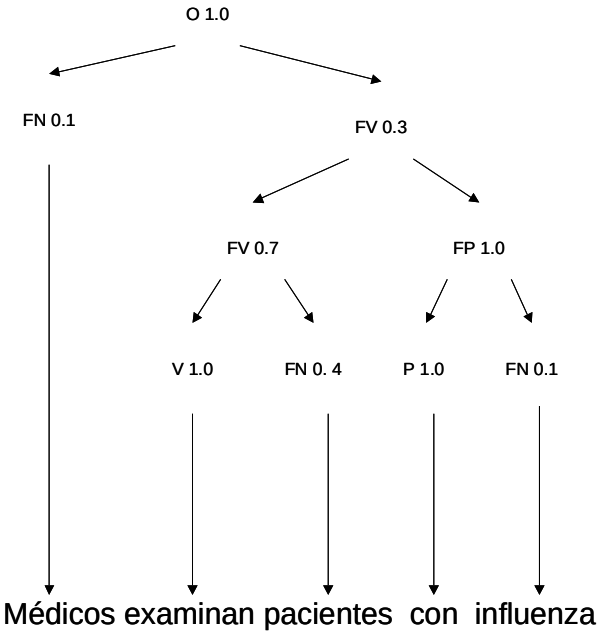

Figura 12. Árbol sintáctico t_2 de la frase *médicos examinan pacientes con influenza*.

O $\rightarrow$ FN FV	1.0
FN $\rightarrow$ *médicos*	0.1
FV $\rightarrow$ V FN	0.7
V $\rightarrow$ *examinan*	1.0
FN $\rightarrow$ FN FP	0.4
FN $\rightarrow$ *pacientes*	0.4
FP $\rightarrow$ P FN	1.0
P $\rightarrow$ *con*	1.0
FN $\rightarrow$ *influenza*	0.1

Considerando las derivaciones, las probabilidades de elegibilidad de los árboles son:

$P(t_1) = 1.0 \times 0.1 \times 0.7 \times 1.0 \times 0.4 \times 0.04 \times 1.0 \times 1.0 \times 0.18$

$= 0.0002016$

$P(t_2) = 1.0 \times 0.1 \times 0.3 \times 0.7 \times 1.0 \times 1.0 \times 0.04 \times 1.0 \times 0.18$

$= 0.0001512$

Considerando la gramática descrita, se tienen dos posibles árboles para la frase *médicos examinan pacientes con influenza*, junto con las probabilidades asignadas, dadas las reglas. La probabilidad resultante es mayor en t_1.

Debido a que las gramáticas se expanden para dar cobertura a corpus grandes y diversos, estas llegan a ser ambiguas. Una PCFG da una idea de elegibilidad a diferentes análisis sintácticos.

Adicional a esto, presentan robustez, al dar a oraciones no elegibles, una menor probabilidad. Las PCFGs son buenas en inducción gramatical. Esto es, aunque para el aprendizaje automático de una gramática a partir de corpus se requieren ejemplos no gramaticales, con estas gramáticas es posible *aprender* a partir de construcciones gramaticales.

Para [MANNING, 00], la inducción gramatical consiste en *agrupar* o reconocer estructuras unitarias de mayor nivel que permitan empaquetar la descripción de una oración. El aprendizaje de una gramática a partir de las estructuras que se encuentran, es inducción gramatical.

Las PCFGs no son modelos buenos por si solas, pero en combinación con otros métodos como proximidad semántica, son modelos fuertes para la desambiguación.

En el análisis sintáctico se presentan diversas formas de ambigüedad. La principal está relacionada con el WSD, y se da cuando no se puede establecer la estructura

sintáctica correcta debido a que las palabras tienen más de una categoría.

Otro tipo de ambigüedad se refiere a los complementos circunstanciales, por ejemplo en *La niña con vestido rojo juega a saltar la cuerda con nudos*, el grupo *con nudos*, se refiere a *la cuerda* y no a *la niña*. Esto es claro para un hablante nativo, pero no para la computadora.

De acuerdo a [WINOGRAD, 83], se dice que una gramática libre de contexto es ambigua si puede ser usada para derivar dos árboles diferentes que tienen la misma secuencia de nodos hojas.

Un analizador sintáctico puede tomar dos enfoques diferentes respecto a la ambigüedad: puede buscar la primera interpretación y detenerse cuando la encuentra o devolver todas las posibles interpretaciones.

2.4.2 Métodos probabilísticos de desambiguación sintáctica

Uno de los usos más comunes de las probabilidades en el análisis sintáctico, es la desambiguación. El propósito para un sistema que pretende desambiguar es elegir un árbol sintáctico de una oración particular, dado un conjunto de árboles sintácticamente posibles de la misma.

Para determinar el significado de una oración, es necesario determinar el significado de las unidades significativas y la forma en que se relacionan.

Generalmente en un análisis sintáctico se obtienen más árboles sintácticos que los deseados. Esto en los enfoques clásicos, puede verse como un hueco en la gramática. Sin embargo, en enfoques probabilísticos se considera y se busca solo el que la distribución de las probabilidades en los diferentes árboles, sea la mejor.

De un análisis sintáctico esperamos que sea capaz de tomar una oración *s* y generar árboles sintácticos de acuerdo a una gramática *G*. En el análisis sintáctico

probabilístico, se desea establecer un orden en los posibles análisis sintácticos basado en la elegibilidad de cada uno, esto es, un modelo probabilístico de análisis sintáctico debe encontrar la probabilidad en árboles *t* para una oración *s.*

P(*t*|s, *G*) donde $\sum_{t} P(t \mid s, G) = 1$

En este tipo de análisis sintáctico para desambiguación se utilizan PCFGs y su enfoque es del tipo *top down* o descendente.

2.5 Compilación de diccionarios

Según [GALICIA, 00], el uso del léxico implementado en computadora, lleva a una mayor convergencia de la teoría léxica y la práctica lexicográfica, ya que puede proveer información estadística y permite la manipulación de información en forma más rápida, esto además de facilitar el trabajo del lexicográfico, le permite tomar mejores decisiones.

Una de las principales propuestas para la compilación de diccionarios por computadora es que puedan incluirse métodos lexicográficos que realicen algunas de las tareas de los expertos, para reducir el tiempo de análisis (oración por oración) de un corpus de textos.

Según [HERNÁNDEZ, 04], la principal motivación para el uso y desarrollo de los diccionarios computacionales es que la mayor parte de los diccionarios académicos tiene solamente algún tipo de información sobre las palabras, así que se necesitan diferentes diccionarios para procesar la información.

Los diccionarios computacionales fácilmente pueden combinarse unos con otros. Sin embargo esta tarea presenta otras complicaciones debido a que los diferentes diccionarios contienen conjuntos de palabras que no coinciden entre sí y al combinarse pueden dar como resultado, información redundante o “huecos” de

información.

En adición a esto, la combinación de los diferentes diccionarios solo resulta significativa si los distintos sentidos que contienen las palabras homónimas y polisemánticas se combinan correctamente, puesto que frecuentemente no se puede reconocer que sentido de una palabra corresponde a que sentido de la misma en otro diccionario [BOLSHAKOV, 04].

A pesar de las dificultades expuestas, los diccionarios electrónicos continúan desarrollándose, y en los últimos años se han presentado diccionarios terminológicos, monolingües, bilingües, etc. [HERNÁNDEZ, 04].

Un término que es conveniente aclarar es el de *colocaciones*. Las colocaciones lingüísticas son combinaciones de dos o más palabras, que presentan unidad semántica y sintáctica. Ejemplo de estos objetos lingüísticos son: *paciencia infinita*, *mesa redonda*, etcétera.

El uso de las colocaciones es claro cuando se diseña un algoritmo de traducción automática de textos, en el que traducir las palabras miembro de una colocación aisladamente, resulta en una pérdida del sentido o información semántica del texto.

Algunos ejemplos se muestran en las secciones 2.4.1 hasta 2.4.4.

2.5.1 Oxford Collocations Dictionary

Este diccionario (OCD), contiene más de 150,000 colocaciones. Muestra las palabras dependientes que tienen un uso común con su palabra cabecera como sustantivos, verbos, adjetivos, adverbios y preposiciones así como también frases comunes.

Este diccionario está basado en el British National Corpus, que contiene 100

millones de palabras y usa búsquedas en Internet para asegurar el uso actualizado de términos en áreas como la computación.

Contiene más de 50,000 ejemplos que muestran la forma en la que se usan las colocaciones en cada contexto, con información gramática y de la referencia, donde se necesita.


New search: Search

Select entry: notwithstanding

not•with•stand•ing /ˌnɒtwɪðˈstændɪŋ; wɪð-; *NAmE* ˌnɑːt/ prep., adv.

■ *prep.* (*formal*) (also used following the noun it refers to) without being affected by sth; despite sth: *Notwithstanding some major financial problems, the school has had a successful year.* ◇ *The bad weather notwithstanding, the event was a great success.*

■ *adv.* (*formal*) despite this SYN HOWEVER, NEVERTHELESS: *Notwithstanding, the problem is a significant one.*

©Oxford University Press, 2005.

Figura 13. Ejemplo del diccionario en línea del OCD.

En este diccionario los grupos de colocaciones son distribuidos usando la combinación de parte del habla (*part-of-speech*) y su significado, lo que ayuda al usuario a encontrar rápidamente la palabra cabecera, el sentido y la colocación requerida.

2.5.2 English CrossLexica

El sistema CrossLexica (Bolshakov, 1994; Bolshakov y Gelbukh, 2001) es uno de

los diccionarios más grandes de colocaciones y combinaciones de palabras libres.

El sistema está formado por un tesauro y un diccionario de combinaciones de palabras en inglés, con una interfaz amigable al usuario y con varios modos de búsqueda. Proporciona las combinaciones de palabras como *pay attention*, *strong tea*, etcétera.

El sistema inteligente tiene la posibilidad de deducir millones de combinaciones probables no codificadas directamente en el diccionario.

Este diccionario proporciona las siguientes facilidades:

- Los sustantivos, adverbios y verbos se pueden combinar con el verbo dado
- Los sustantivos pueden modificarse por el adjetivo dado o modificarlo,
- Los modelos de subcategorización de verbos, sustantivos y adjetivos,
- Formas morfológicas de palabras,
- Traducciones de combinaciones como “té cargado”: strong tea,
- Sinónimos, antónimos, hipónimos, hiperónimos de las palabras,
- Marcas estilísticas del uso de las palabras.

El sistema es multilingüe, maneja los idiomas inglés y ruso y por su robustez y organización, es de los más reconocidos y utilizados en el PLN.

2.5.3 CrossLexica Española

El diccionario CrossLexica Española (Miranda-Jiménez, 2003; Miranda-Jiménez y Bolshakov, 2002) es un diccionario especial de español que incluye combinaciones sintácticas entre palabras tales como *mesa redonda*, así como

información semántica *limón – árbol*, como complemento a las colocaciones.

El diccionario CrossLexica Española está basado en el sistema English CrossLexica.

2.5.4 WordNet

[MILLER, 93] La base de datos léxica WordNet, es considerada como el recurso más importante para los investigadores en lingüística computacional, análisis de textos y muchas áreas relacionadas. Su diseño se inspiró en las teorías computacionales y psicolingüísticas actuales del léxico humano.

El idioma que maneja es el inglés, y contiene sustantivos, verbos, adjetivos, y adverbios que están organizados en conjuntos de sinónimos, cada uno representando un concepto léxico.

La principal motivación de desarrollar WordNet fue compilar un diccionario que abarcara un vocabulario más amplio al de las herramientas disponibles.

WordNet presenta aproximadamente 95,600 formas de palabras diferentes (51,500 palabras simples y 44,100 colocaciones) organizadas en aproximadamente 70,100 significados de palabras, o conjuntos de sinónimos.

La diferencia más obvia entre WordNet y los diccionarios estándar es que WordNet divide el lexicón en estas categorías: sustantivos, verbos, adjetivos, y palabras función. De hecho, WordNet solo contiene sustantivos, verbos, adjetivos y adverbios debido al tamaño del conjunto de palabras función en el idioma inglés.

Las palabras función, también llamadas forma de palabra u operadores (en inglés *functor*) son preposiciones, conjunciones o artículos, que tienen poco contenido

semántico propio y básicamente indican una relación gramatical.

La alta redundancia que presenta WordNet, permite que se explote mejor la información semántica de las palabras. Una palabra está contenida en más de una categoría sintáctica.

WordNet intenta organizar la información léxica en términos de significados de palabras, más que en formas de palabras. Por tanto, se considera a WordNet más cercano a un *tesauro* que a un diccionario.

La semántica léxica comienza con el reconocimiento de que una palabra es una asociación convencional entre un concepto léxico o representación lexicalizada y su vocalización, que desempeña un rol [MILLER, 93]. La definición de "palabra" conlleva al menos tres problemas para la investigación: ¿qué tipo de vocalizaciones entran en estas asociaciones léxicas? ¿cuál es la naturaleza y organización que los conceptos lexicalizados pueden expresar? ¿qué roles sintácticos desempeñan las palabras?

Según [MILLER, 93], una matriz léxica se representa con dos dimensiones, una de ellas será la "forma de palabra", esto es, la representación física de la palabra y por otra parte el "significado de palabra" que se refiere al concepto lexicalizado que una forma puede usar para expresarse. Entonces el punto de inicio para la semántica léxica sería el mapeo entre las formas y los significados [MILLER, 93].

En el modelo de matriz léxica, las categorías sintácticas pueden tener diferentes tipos de mapeos, esto significa que una celda de la matriz contendría la siguiente asociación: la forma en la columna puede ser usada en algún contexto determinado para expresar el significado en ese renglón.

Tabla 6. Ejemplo del modelo de matriz léxica

Significado de palabra	Formas de palabras				
	F_1	F_2	F_3	…	F_n
M_1	$E_{1,1}$	$E_{1,2}$			
M_2			$E_{2,2}$		
M_3			$E_{3,3}$		
…				...	
M_n					$E_{m,n}$

En la tabla 6, F_1 y F_2 son sinónimos y F_3 es polisémica.

2.6 Aprendizaje automático de la base de batos estadística de combinaciones de palabras en español

En esta sección se presenta una breve descripción de la tesis desarrollada.

2.6.1 Uso de combinaciones de palabras vs. patrones de manejo en el diccionario

Las estructuras que forman el diccionario son combinaciones de palabras en Español. De estas combinaciones tienen especial relevancia las colocaciones, en especial para tareas como traducción de textos.

Una colocación es una expresión consistente de dos o más palabras que

corresponden a un modo convencional de referirse a algo, en un idioma dado. Las colocaciones corresponden a expresiones como *leche cortada*, *boca calle*, *cara de niño*, etc.

Según [MANNING, 00], las colocaciones se caracterizan por su limitada *composicionalidad*. Una expresión en lenguaje natural es *composicional* si el significado de la expresión puede predecirse por sus partes.

Las características de las colocaciones son:

- No composicionalidad: El significado de una composición no puede ser determinado directamente por el significado de sus partes.
- No sustituibilidad: No es posible sustituir con otras palabras los constituyentes de la colocación y obtener el mismo significado, aunque estas sean sinónimos.
- No modificabilidad: No es trivial modificar las colocaciones con información léxica adicional o debido a transformaciones gramáticas.

Las combinaciones de palabras que forman el diccionario que se compilará en esta tesis están ampliamente relacionadas con las colocaciones, sin embargo son diferentes a las colocaciones respecto a que tienen alta composicionalidad y como característica tendrían únicamente su frecuencia en el uso común del lenguaje.

La prueba básica para comprobar si una combinación de palabras es una colocación consiste en tratar de hacer una traducción a otro lenguaje. Si no es posible traducir la frase palabra por palabra, entonces con mucha probabilidad se trata de una colocación.

Respecto a los patrones de manejo, [MEL'CUK, 88] en su Teoría Texto ⇔ Significado (*Meaning⇔Text Theory,* MTT), describe la diátesis de cada verbo, que

es la correspondencia entre los actores semánticos y los de la sintaxis superficial. La MTT describe la subcategorización para cada verbo, y para los distintos usos de un mismo verbo.

Según [GALICIA, 99] “En los formalismos basados en constituyentes, esta separación no existe, por lo que pueden incluirse predicados cuya ocurrencia es obligatoria en el contexto local de la frase pero que no son seleccionados semánticamente por el verbo. Al no considerarse la información de subcategorización de una forma específica para cada verbo, generalmente se realiza una clasificación y entonces cada clase (marco de subcategorización) es un patrón de composición de complementos que puede ser compartido por varios verbos. Bajo este esquema la alternación de la diátesis considera que el verbo puede aparecer en una diversidad de marcos de subcategorización.”

En la teoría Teoría Texto ⇔ Significado se introducen los *Government Patterns* (patrones de manejo, *PM*) para la descripción de los objetos de los verbos. Para ello se usa una tabla de PM con la información entre las valencias semánticas y sintácticas de la palabra núcleo o palabra cabeza, con las descripciones del uso de las valencias sintácticas e información como la *optatividad* de cada actuante.

Adicional a ello se presentan las restricciones y ejemplos de cada uno de los casos.

Dada una breve descripción de una y otra estructuras lingüísticas se tienen las características descritas en las siguientes secciones.

2.6.1.1 Uso de patrones de manejo sintáctico en el método Galicia-Haro, et al.

– Se tiene un marco, como un conjunto de subcategorías y después se intenta clasificar la diversidad total de verbos para ese marco. Esta

aproximación es suficientemente buena cuando el número total de subcategorías es pequeño, pero no así en lenguajes donde casi cada verbo presenta su propia subcategoría específica, como en español. Generalmente no intentan establecer correspondencia entre valencias sintácticas y semánticas.

– La separación ente complementos del verbo y complementos circunstanciales no existe, por lo que pueden incluirse predicados cuya ocurrencia es obligatoria en el contexto de la frase pero que no son seleccionados semánticamente por el verbo.

– Usualmente, en cada subcategoría, las valencias sintácticas se consideran en un orden fijo predeterminado, si se cambia el orden la regla falla y será necesario incluir nuevas reglas.

2.6.1.2 Uso de combinaciones de palabras en el método Galicia-Haro et al.

– Dado un conjunto de árboles sintácticos generados por la herramienta PARSER del Laboratorio de Lenguaje Natural del CIC, se construye el prototipo de diccionario extrayendo de estos árboles todos los pares de palabras sintácticamente dependientes una de la otra, a los cuales llamaremos combinaciones de palabras, o en caso de ramas del árbol que contienen preposiciones, las combinaciones de palabras del tipo: verbo – preposición – sustantivo ó sustantivo – preposición – sustantivo.

– No existen como en los patrones de manejo, consideraciones referentes a los complementos de un verbo dado, por lo que no se discriminan aquellas combinaciones de palabras que se extraigan de árboles sintácticos gramaticalmente correctos pero semánticamente incorrectos.

– Debido a que en el método Galicia-Haro et al. se asignan pesos muy

pequeños a todas las combinaciones al inicio del algoritmo iterativo, estas combinaciones de palabras semánticamente incorrectas pueden introducir ruido cuando el diccionario de construye a partir de corpus pequeños.

– El orden de las palabras que formen una combinación en esta variante del modelo Galicia-Haro et al. no es relevante, ya que los diferentes órdenes en que se presente una frase o cláusula generan para el diccionario de diferentes palabras y se calcula el peso estadístico de acuerdo a su frecuencia.

El método presentado en [GALICIA, 00], se basa principalmente en tres herramientas: Base de datos de combinaciones de palabras en español (diccionario) con pesos estadísticos, reglas ponderadas y proximidad semántica.

En esta variante del método, solo se utiliza la asignación de pesos estadísticos.

En el método Galicia-Haro, et al. cada uno de los módulos da una medida cuantitativa de la probabilidad de una u otra variante de estructura, y finalmente el sistema elige las variantes con los valores más altos de esas evaluaciones estadísticas [GALICIA, 00].

El modelo de PMA se refiere a estructuras lingüísticas, o combinaciones de palabras que adquieren los hablantes nativos de una lengua durante el aprendizaje de su lenguaje.

El conocimiento descrito en los patrones de manejo es la información léxica de verbos, adjetivos y algunos sustantivos del español. Según [GALICIA, 00], no es posible establecer ese conocimiento mediante reglas o algoritmos pero es posible obtener tal información a partir de un corpus.

El modelo de proximidad semántica contiene conocimiento semántico que se

requiere en la desambiguación en oraciones con estructuras sintácticas correctas. En él se trata de reconocer las palabras que están relacionadas o que son "semánticamente compatibles" [GALICIA, 00].

Por último el módulo de votación elige la mejor variante de acuerdo a los pesos que se asignaron en cada uno de los módulos anteriores.

Una vez descrito brevemente el método Galicia-Haro et al., se debe aclarar que en esta tesis no se evalúan los métodos alternos para cálculo de los pesos estadísticos, como las reglas ponderadas ni el módulo de proximidad semántica.

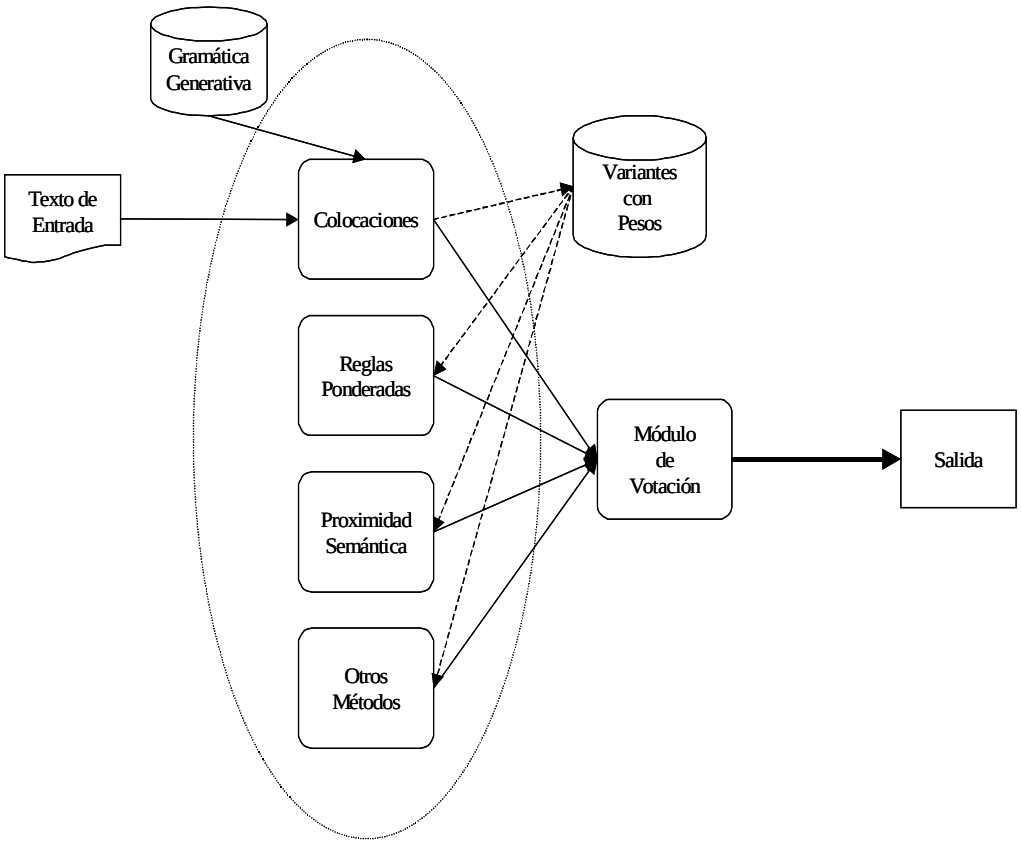

Figura 14. Estructura del analizador con resolución de ambigüedad basado en patrones de manejo sintáctico.

2.6.2 Trabajos relacionados

[LÚCIO, 02], en su sistema ATA utiliza una combinación de métodos estadísticos y métodos con conocimiento lingüístico para determinar si una combinación de palabras es un *término*. En este documento, *término* es una representación lingüística de un concepto por medio de un sustantivo simple o una frase sustantival. El uso de esta herramienta es la construcción semiautomática de índices terminológicos, sistemas de extracción de términos.

[ATLE, 06], utiliza un método de extracción de frases-clave, en inglés *key phrases* para construir y mantener ontologías para usarse en sistemas avanzados de recuperación de información. En este método los componentes son: preproceso de datos, generación de frases candidatas y cálculo de pesos y selección de frases. En este último componente, se calculan pesos para las frases de acuerdo a su utilidad esperada y se produce una lista de frases-clave para después seleccionar la lista final.

[BOUILLON, 00], presenta una herramienta para desambiguación léxica para Recuperación de Información de textos médicos. El sistema convierte la información léxica a marcas sintácticas y entonces se entrena el texto con marcas sintácticas en tres etapas: El modelo sintáctico se construye automáticamente a partir de textos ambiguos, siguiendo el algoritmo Baum-Welch. Después las matrices son refinadas con un conjunto de reglas sintácticas (bigramas) y por último una pequeña parte del texto (5000 palabras) es usada para reestimar el modelo.

[ALDEZABAL, 01], presenta un trabajo de extracción automática de información verbal, a partir de un análisis sintáctico parcial utilizando una gramática de estado finito, expresada en expresiones y relaciones regulares mediante autómatas y transductores de estado finito.

En [HERNÁNDEZ, 04], se presenta un algoritmo para compilación de un diccionario de colocaciones basadas en frases preposicionales y de coordinaciones conjuntivas.

CAPÍTULO 3 COMPILACIÓN DE LA BASE DE DATOS ESTADÍSTICA DE COMBINACIONES DE PALABRAS EN ESPAÑOL

En este capítulo presentamos el método utilizado [GALICIA, 00].

3.1 Método utilizado

El método inicia con el análisis sintáctico de un corpus. El corpus utilizado está marcado de acuerdo a las categorías gramaticales que utiliza el LEXESP.

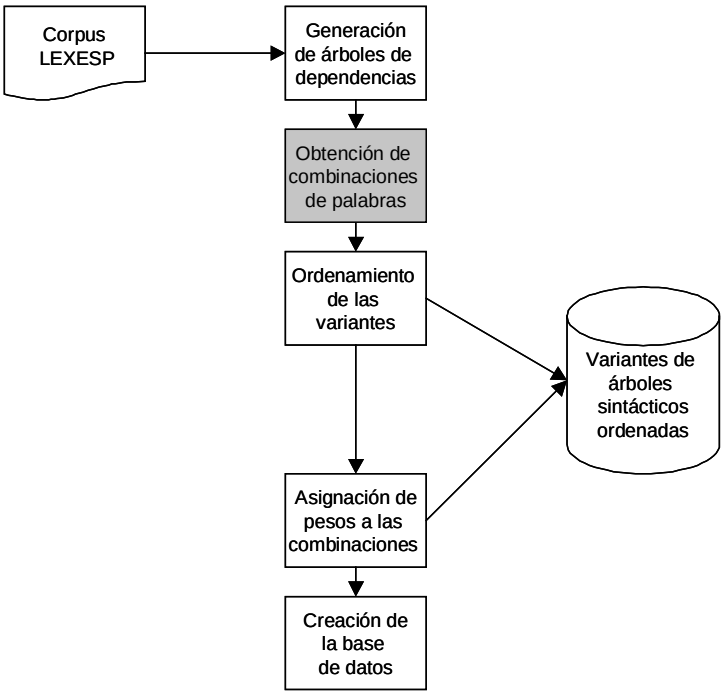

Figura 15. Diagrama general del método utilizado en el aprendizaje automático de la base de datos de combinaciones de palabras en español.

Con la ayuda del la herramienta PARSER del Laboratorio de Lenguaje Natural se realiza el análisis sintáctico y se generan las variantes de los árboles, en modo de árboles de constituyentes. Estos se convierten a árboles de dependencias.

Después se extraen las combinaciones de palabras de estos árboles. Estas combinaciones de palabras, que son variantes del prototipo de diccionario agrupadas en combinaciones por variante, junto con las variantes por oración y los totales, son la entrada para calcular los pesos.

El resultado final del método es una tabla con las combinaciones de palabras o variantes obtenidas con los pesos calculados.

3.2 Modelo matemático del método Galicia-Haro et al.

Desde el punto de vista matemático, el problema y su solución son los siguientes, este modelo fue tomado de [GALICIA, 00] y se incluye para completitud de la tesis presentada:

Sea :

F el conjunto de características de un árbol sintáctico, tales como "*poner* + objeto + *en* + lugar", "objeto + *para* + función", denotadas como $f_1, f_2,\ldots,f_n$.

P una frase tal que ***P*** $\in$ ***F***, y ***P*** = { f_{n_1} , f_{n_2} ,..., f_{n_k} }

El modelo de generación de pesos estadísticos de las diferentes variantes de árboles sintácticos de una frase ***P***, una fuente ***S*** que contiene la característica $f_i \in$ ***F*** incluye la característica f_i junto con su probabilidad p_i en el modelo. Por ejemplo, el generador *S* puede incluir la característica "bueno + para + corazón" en una de cada mil frases con la correspondiente probabilidad p_i= 0.001.

Entonces el método se basa en el conjunto de frecuencias p_i de combinaciones

individuales $f_i \in \boldsymbol{F}$.

A diferencia del modelo presentado en [GALICIA, 00] “el modelo de generación opera de la siguiente manera: para generar una frase ***P***, una fuente ***S*** conteniendo la característica $f_i \in \boldsymbol{F}$, decide si esta característica f_i será incluida o no en la frase ***P”***, en este modelo se incluyen todas las combinaciones significativas encontradas, dependiendo de la relación que etiquete a esa característica. Con esto se pretende compilar la información necesaria respecto a frases preposicionales y verbos, conjunciones, adverbios y relaciones significativas para la desambiguación, su uso en español y la frecuencia de cada variante de su uso, dado un corpus de más de cien mil frases marcadas.

Entonces, en base a estas consideraciones y en el hecho de que las combinaciones y su frecuencia son independientes entre sí, tenemos que las probabilidades o pesos de las mismas se calculan de la siguiente manera:

$$\alpha_n^{k,r} = \begin{cases} p_n^r \\ q_n^r \end{cases} \qquad (1)$$

donde:

p es la probabilidad de que la combinación se seleccione

q es la probabilidad de que la combinación no se seleccione y su valor es: $q = 1 - p$

k es el número de la variante

r es 1 si corresponde a la variante correcta (se representa con “+”)

n es el número de combinaciones

Así que se tienen las siguientes probabilidades:

p_n^+, si $f_n \in \boldsymbol{P}$ y k es la variante correcta

q_n^+, si $f_n \notin \boldsymbol{P}$ and k es la variante correcta

p_n^{--}, si $f_n \in \boldsymbol{P}$ and k es la variante incorrecta

q_n^{--}, si $f_n \notin \boldsymbol{P}$ and k es la variante incorrecta

Entonces la probabilidad de ***P*** es:

$$P(\boldsymbol{P}) = \prod \alpha_n^{k,r} \tag{2}$$

dado que cada característica está incluida en la frase ***P*** con las probabilidades α, r denota:

$$r = \delta_i^k \begin{cases} 1 & \text{si } k = i \text{ (variante correcta)} \\ 0 & \text{si } k \neq i \text{ (variante incorrecta)} \end{cases}$$

Por lo tanto las probabilidades pueden verse como una matriz V con k filas, una fila para cada variante, y n columnas, una columna para cada combinación. Entonces los valores en la matriz son:

$$\alpha_n^{k,r} = \begin{cases} p_n^{\delta_i^k} & V_k[n] > 0 \\ q_n^{\delta_i^k} & V_k[n] = 0 \end{cases} \tag{3}$$

Donde $V_k[n]$ representan los valores de las probabilidades de ocurrencia de las combinaciones presentes, $n \in V_k$. Si la combinación está presente, entonces $V_k[n]$

> 0, si no $V_k[n] = 0$.

Entonces es posible usar la fórmula (2) para la desambiguación, dado el conjunto $\boldsymbol{V}=\{V_1,...,V_N\}$ de variantes. Supóngase que solo una variante es correcta, entonces sea H_j la hipótesis de que la variante V_j es la correcta, y sea ξ el evento de obtención de exactamente el conjunto $\boldsymbol{V}$, como resultado del análisis sintáctico entonces, usando la fórmula de Bayes, se tiene que:

$$P(H_j \mid \xi) = P(\xi \mid H_j)\frac{P(H_j)}{P(\xi)} \tag{4}$$

Para abreviar se denota $P(H_j \mid \xi) \equiv P_j$, la probabilidad de que la variante V_j sea la correcta. Puesto que se hizo la suposición de tener solo una variante correcta, tenemos que:

$$\sum_{V_j \in \boldsymbol{V}} Pj = 1 \tag{5}$$

Para calcular el valor de $P(H_j \mid \xi)$, se asume que:

- No se tiene información a priori sobre las probabilidades de hipótesis individuales
- Todas las variantes son ruido excepto una que es la correcta.

Puesto que el evento ξ no depende por completo de j, podemos ignorar $P(\xi)$, y como no tenemos información a priori de las probabilidades de las hipótesis individuales, consideramos que todas tienen la misma probabilidad, tanto la correcta como las erróneas así que $P(H_j)$ es una constante, entonces (4) puede rescribirse como (6):

$$P_j \sim P(\xi \mid H_j) \tag{6}$$

Donde ~ significa *proporcional*, es decir, P_j = C x P($\xi|H_j$), con una constante de normalización C determinada de (5).

Suponiendo que la hipótesis H_j es verdadera, es decir, que V_j y todas las otras variantes V_k, donde $k \neq j$, son ruido, entonces:

$$P(\xi \mid H_j) \sim \prod_{k=1}^{K} \prod_{n=1}^{N} \alpha_n^{k,\delta_i^k} \tag{7}$$

donde N es el número de combinaciones y K es el número de variantes.

Suponiendo que se tienen dos fuentes de información una que genera la variante correcta y otra que genera las variantes espurias o ruido, entonces podemos introducir un elemento unitario compuesto de las probabilidades q correspondientes a todas las combinaciones presentes en las variantes, es decir, para las p ($n \in V_k$) en toda la matriz, se tiene:

$$\prod_{k=1}^{K} \prod_{n=1}^{N} \alpha_n^{k,\delta_i^k} = \prod_{n=1}^{K} \left(\left(\prod_{n=1}^{N} \alpha_n^{k,\delta_i^k} \right) \left(\frac{\prod_{n \in V_k} p_n^{k,\delta_i^k}}{\prod_{n \in V_k} q_n^{k,\delta_i^k}} \right) \right)$$

$$= \prod^{K} \left(\left(\prod_{n=1}^{N} q_n^{k,\delta_i^k} \right) \left(\prod_{n \in V_k} \frac{p_n^{k,\delta_i^k}}{q_n^{k,\delta_i^k}} \right) \right)$$

donde $\prod_{k=1}^{K} \prod_{n=1}^{N} \alpha_n^{k,\delta_i^k}$ es la matriz de probabilidades q. En esta matriz, las probabilidades q^+ están en la fila correspondiente a la variante correcta y las probabilidades q^- en K-1 filas. Esta manipulación puede verse como la limitación

de la matriz a las combinaciones presentes en las variantes para la frase dada.

$$\prod_{k=1}^{K}\prod_{n=1}^{N}\alpha_n^{k,\delta_i^k} = \prod_{n=1}^{K}\prod_{n\in V_k}\frac{p_n^{k,\delta_i^k}}{q_n^{k,\delta_i^k}} \tag{8}$$

Nuevamente se manipula algebraicamente la fórmula anterior con el elemento unitario compuesto del cociente p^-/q^- para todas las combinaciones correctas presentes en la variante correcta *i*.

$$\prod_{n=1}^{K}\prod_{n\in V_k}\frac{p_n^{k,\delta_i^k}}{q_n^{k,\delta_i^k}} = \prod_{n=1}^{K}\left(\left(\prod_{n\in V_k}\frac{p_n^{k,\delta_i^k}}{q_n^{k,\delta_i^k}}\right)\left(\frac{\prod_{n\in V_i}\frac{p_n^-}{q_n^-}}{\prod_{n\in V_i}\frac{p_n^-}{q_n^-}}\right)\right)$$

$$= \prod_{n=1}^{K}\left(\left(\prod_{n\in V_k}\frac{p_n^-}{q_n^-}\right)\left(\frac{\prod_{n\in V_i}\frac{p_n^+}{q_n^+}}{\prod_{n\in V_i}\frac{p_n^-}{q_n^-}}\right)\right)$$

Esta manipulación corresponde ahora a limitar el espacio de eventos a la parte de las combinaciones presentes en la variante correcta. El factor $\prod_{n=1}^{K}\prod_{n\in V_k}\frac{p_n^-}{q_n^-}$ corresponde a todas las combinaciones que no están presentes en la variante correcta, así que se puede eliminar con cierta pérdida:

$$\prod_{k=1}^{K}\prod_{n=1}^{N}\alpha_n^{k,\delta_i^k} \approx \prod_{n\in V_i}\frac{p_n^+}{q_n^+}\frac{q_n^-}{p_n^-} = \prod\frac{p_n^+(1-p_n^-)}{p_n^-(1-p_n^+)} \tag{9}$$

Como p^- y p^+ son valores pequeños, entonces $(1-p^-)/(1-p^+)$ tiende a uno por lo que se obtiene finalmente:

$$\prod_{k=1}^{K}\prod_{n=1}^{N}\alpha_n^{k,\delta_i^k} \approx \prod_{n\in V_i}\frac{p_n^+}{p_n^-} \tag{10}$$

Así que para calcular el peso de la variante *j*-ésima, deben tomarse del diccionario ***F*** las frecuencias p_i^+ y p_i^- de todas las características f_i encontradas en esta variante V_j, y después aplicarse en la fórmula (10).

Según [GALICIA, 00], una causa de errores es que algunas combinaciones aparecen en muy pocas ocasiones, por lo que el valor del cociente p_i^+ / p_i^- se incrementa y causa problemas.

Los resultados son más estables al suprimir casos muy raros. La solución fue añadir artificialmente algún ruido adicional a las características con frecuencia muy baja. Los mejores resultados se obtienen con la siguiente expresión:

$$P_j \sim \prod_{f_i\in V_j}\frac{p_i^+}{p_i^- + \lambda} \tag{11}$$

Los valores que se toman para calcular los pesos son:

Valor de la constante lambda λ:

$$\lambda = \begin{cases} n_s - n_0 \\ n_v/(n_s - n_0) \end{cases} \tag{12}$$

Con n_S = número de sentencias, n_v = número de variantes y n_0 = número de oraciones sin variantes.

Cálculo del peso estadístico de las combinaciones:

$$peso\_estadístico = \begin{cases} p^+q^- / p^-q \\ p^+ / p^- \\ p^+ \\ p^+ / q^+ \end{cases} \quad (13)$$

Con las fórmulas (12) y (13) se obtienen los pesos estadísticos para el prototipo de diccionario de combinaciones de palabras en español.

3.3 Obtención de los árboles de dependencias

La compilación de la base de datos estadística de combinaciones de palabras en español, consiste en la obtención de árboles de constituyentes por medio del PARSER del Laboratorio de Lenguaje Natural para convertirlos en árboles de dependencias, a partir de los cuales se extraen las combinaciones de palabras.

3.3.1 Corpus LEXESP

Como entrada al modelo se tiene el corpus LEXESP. Este corpus tiene las categorías gramaticales en PAROLE. El corpus tiene cinco millones de palabras marcadas. Esta información fue tomada de [GALICIA, 00] y se incluye para completitud de la tesis presentada.

3.3.2 Uso del PARSER

Para generar las estructuras intermedias de análisis o árboles sintácticos de constituyentes, se utilizó la herramienta PARSER desarrollada en el Laboratorio de Lenguaje Natural en el Centro de Investigación en Computación, por A. Gelbukh y G. Sidorov, 1999. El PARSER es un programa que permite investigar la estructura morfológica y sintáctica de oraciones del Español utilizando una

gramática libre de contexto extendida.

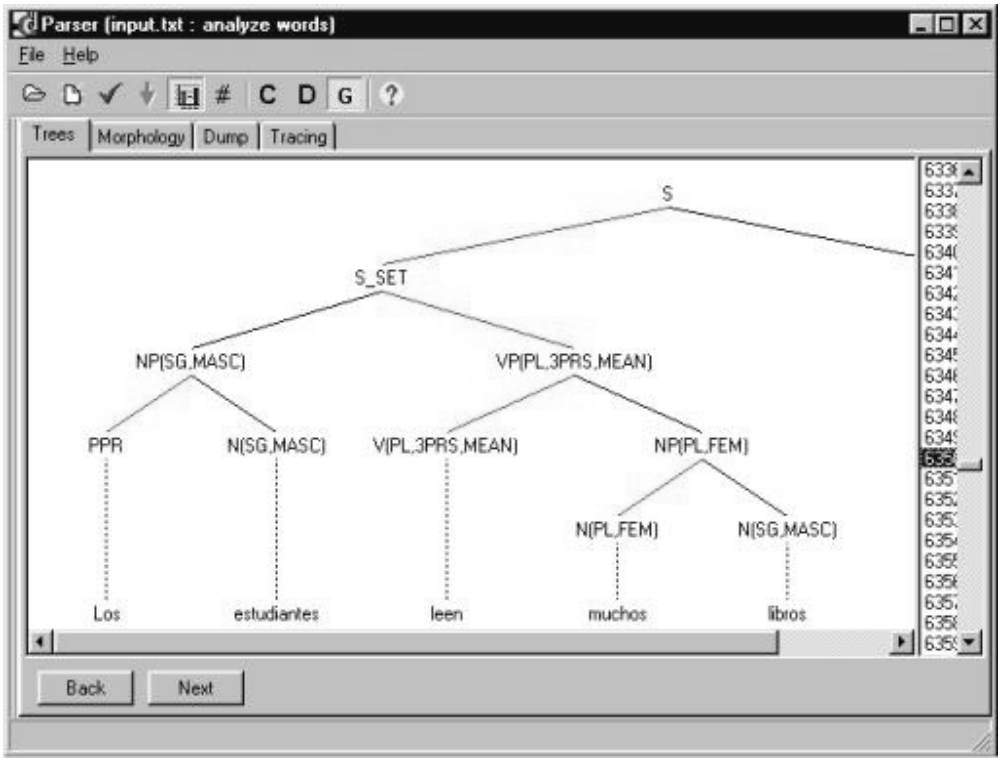

Figura 16. Pantalla principal de la herramienta PARSER.

Con esta herramienta es posible:

- Ver las variantes de la estructura sintáctica de las oraciones,
- Ver las variantes de análisis morfológico en las palabras en oraciones
- Investigar el protocolo del proceso de análisis sintáctico, para entender el trabajo interno del PARSER.

El PARSER realiza el análisis sintáctico usando la gramática descrita en el Anexo A. Gramática generativa. El análisis se hace en base a opciones configurables,

mismas que se describen en el Anexo B. Parámetros de análisis sintáctico del PARSER.

3.3.3 Algoritmo de conversión de árboles de constituyentes a árboles de dependencias

De acuerdo a [GALICIA, 00] para aplicar reglas con un solo núcleo como,:

PP $\rightarrow$ @:PR N ó CLAUSE $\rightarrow$ @:V NP, donde @ denota al elemento rector, se requiere la conversión de árboles de constituyentes (indican el elemento rector) a árboles de dependencias.

El algoritmo para hacer esta conversión es:

Convertir_a_dependencias

Para cada hijo **q** del nodo **n** (del árbol de constituyentes) que no cubre un terminal de izquierda a derecha, hacer

Convertir a dependencias

Asignar el nodo **m** (del árbol de dependencias) al elemento

rector de los hijos del nodo **n**

Para todos los hijos del nodo **n** (que no sean el elemento

rector) hacerlos dependientes de **m**

Trasladar las marcas de dependencias

Asignar como nodo superior de **m** al mismo nodo superior de **n** y

eliminar el nodo **n**

La representación de árboles de dependencias es exponencialmente más manejable que la representación de árboles constituyentes, aunque es más

natural la representación de constituyentes.

Esta la representación utilizada en el modelo [GALICIA, 00], por tanto aquí se busca exponer todos los detalles del modelo.

3.4 Obtención de combinaciones y ordenamiento de las variantes

El algoritmo de obtención de combinaciones de palabra en español es el siguiente:

Obtener_combinaciones

Para todas las frases en la fuente S

Para cada árbol de dependencias en el banco de árboles de una frase

Para cada hijo q del nodo n (del árbol de dependencias) de izquierda a derecha, hacer

Verificar la parte izquierda de la regla (PR -> 123)

Si parte_izquierda es PR (Preposición) entonces

Extrae la información del nodo y asígnala a la parte

Izquierda de la combinación de palabras

Si existe un hijo_izquierdo y NO existe un vecino_derecho entonces

asigna al complemento de la combinación la

relación y el lema del hijo_izquierdo

asigna al nodo actual

el nodo hijo_izquierdo

fin

fin

Asignar pesos a las combinaciones de palabras obtenidas

fin

Por último se escriben con el formato que utilizará el siguiente módulo del programa para generar los pesos de las variantes.

De acuerdo a [GALICIA, 00], La ecuación (10) permite obtener el peso de las variantes de análisis con los pesos anteriores de cada una de sus combinaciones y contribuir con esos nuevos valores para su reestimación.

El algoritmo de asignación de pesos estadísticos, tomado de [GALICIA, 00] es:

1. En el inicio todos los pesos son cero
2. Para cada frase de entrada, se construyen todas las variantes de análisis de acuerdo a la gramática que el analizador sintáctico emplea.

3. Para cada variante se estima su peso w_k, conforme a (10), es decir, el producto de las frecuencias de las combinaciones presentes en la variante.
4. Los pesos se normalizan.
5. Cada variante se separa en estructuras locales de los nodos. Estas estructuras se incorporan al diccionario.
6. Para cada nodo de cada variante, se adiciona el peso de la variante al peso p^+, y el cálculo (1-w) al peso p^-.
7. Se toma nuevamente el corpus y se sigue al paso 3.

CAPÍTULO 4 EXPERIMENTOS Y RESULTADOS

En este capitulo se presenta en la sección 4.1 el método que se empleó en el desarrollo de las pruebas de la base de datos de combinaciones de palabras en español.

Se presentan también en la sección 4.2 los resultados experimentales.

4.1 Método de evaluación del sistema

La evaluación del sistema se basó en los experimentos presentados en [GALICIA, 00].

4.1.1 Delimitación del propósito y uso del sistema

De acuerdo a [SPARCK, 96] el sistema a evaluar abarca todas las herramientas: interfaces, equipo, sistema operativo, y el conjunto de módulos o subprocesos que lo componen.

Haciendo un ejercicio de este tipo aplicado a esta tesis, obtendríamos la siguiente información:

Tabla 7. Delimitación del propósito y uso del sistema.

Característica:	Valor:
Tarea:	Desambiguación de textos mediante el prototipo de diccionario de combinaciones de palabras en español con pesos estadísticos.

Característica:	Valor:
Aplicaciones dentro del dominio:	Traducción automática, corrección de errores gramaticales, corrección de estilo, recuperación de información, resumen de información, extracción de datos a partir de textos.
Aplicaciones fuera del dominio:	Como parte de los sistemas que utilicen herramientas basadas en PLN, como recuperación de información, minería de datos, sistemas expertos.
Sistemas:	Compilación del diccionario de combinaciones de palabras en español.
Sistema híbrido:	Uso del PARSER.
Subsistema-L (lingüístico):	Compilación de la base de datos de combinaciones de palabras en español.
Subsistema-N (no lingüístico):	Algoritmo iterativo de asignación de pesos estadísticos.

Esta información, nos será útil para determinar en base a los diferentes tipos de evaluación de sistemas de PLN, el mejor método, más claro y más completo para este sistema particular.

4.1.2 Niveles de evaluación

Basado en [SPARCK, 96], deben establecerse criterios formales en los métodos

de prueba y la forma de medir los resultados. El criterio aplicado para llevar a cabo la evaluación puede ser intrínseco y extrínseco: i.e. criterios intrínsecos son aquellos que evalúan los objetivos del sistema y extrínsecos los que evalúan su función.

La evaluación de los objetivos de la base de datos de combinaciones de palabras en español respecto al aprendizaje automático, se realizará conforme a los siguientes criterios:

Dado un corpus marcado, el proceso *S* que es la fuente de las variantes de árboles sintácticos, un conjunto de variantes de árboles sintácticos, y un conjunto de combinaciones de palabras con pesos estadísticos obtenidas a partir de esas variantes, observaremos las siguientes variables:

1. Oraciones en el corpus
2. Número de variantes obtenidas (árboles sintácticos)
3. Tiempo de ejecución
4. Número de combinaciones de palabras
5. Número de combinaciones “útiles”
6. Número de combinaciones “descartadas”

La evaluación de la parametrización se realizará de forma intrínseca, el método de referencia en [GALICIA, 00] usa una base de datos de PMA contra las combinaciones de palabras o colocaciones que se utilizaron en esta tesis. Al evaluar los resultados, evaluaremos si esta nueva parametrización resultó mejor para el método.

Para esto tomaremos como referencia la tabla de resultados de la aplicación de

los pesos de combinaciones en el analizador presentada en [GALICIA, 00]. Para evaluar el método Galicia-Haro, et al., se compiló el diccionario basado en patrones de manejo utilizando el corpus LEXESP, en nuestro caso con este mismo corpus, se genera el prototipo de diccionario de combinaciones de palabras en español. Una vez construido el diccionario, se adaptó la herramienta PARSER para utilizar los pesos estadísticos para asignarle un peso a cada árbol sintáctico generado.

Con esta información se ordenan las variantes y se calcula el rango medio de la variante correcta. Mientras este rango medio sea menor, más cercana se encuentra la variante correcta, de la variante elegida por el PARSER utilizando el diccionario de combinaciones de palabras.

Para ello se utilizaron las oraciones que se especifican en el ANEXO C. Oraciones utilizadas en la evaluación.

La evaluación de la función del sistema se realizará conforme a los siguientes criterios:

Dada una matriz conteniendo un número de la oración, número de las variantes y pesos estadísticos, y el orden en que aparecieron, se observarán las siguientes diferencias contra el método que se tomo como referencia para desarrollar el actual:

1. Porcentaje de oraciones bien evaluadas
2. Porcentaje de oraciones mal evaluadas
3. Porcentaje de oraciones que están bien evaluadas en el método actual y mal evaluadas en el método tomado como referencia [GALICIA, 00]
4. Porcentaje de oraciones que están mal evaluadas en el método utilizado y

bien evaluadas en el método tomado como referencia.

4.2 Resultados experimentales

De la evaluación de los objetivos de la base de datos de combinaciones de palabras en español respecto al aprendizaje automático, se obtuvo:

Oraciones en el corpus:	238563
Oraciones con variantes:	173241
Número de variantes obtenidas (árboles sintácticos):	12682610
Combinaciones de palabras:	412578
Tiempo de ejecución:	1:14 hrs en promedio

Evaluación de los objetivos de la base de datos de combinaciones de palabras en español respecto al aprendizaje automático:

- Número de combinaciones de palabras
- Número de combinaciones “útiles”
- Número de combinaciones “descartadas”

Respecto de la función lambda y el cálculo de pp/pm.

El algoritmo descarta las combinaciones cuyo valor pp/pm resulta menor a $1e^{-7}$.

Tabla 8. Evaluación de los objetivos de la base de datos de combinaciones de palabras en español respecto al aprendizaje automático.

Combinación de parámetros	Función lambda	p^+	# Combinaciones de palabras = combinaciones útiles	Media	Desviación estándar	# Combinaciones de palabras descartadas
0 –1 1	nS – n,0	p^+q^-/p^-q^+	412578	0.0044	0.1356	0
0 –1 2	nS – n0	p^+/p^-	239693	0.0075	0.183	172885
0 –1 3	nS – n0	p^+	74541	0.0014	0.3025	338037
0 –1 4	nS – n0	p^+/q^+	108825	0.016	0.2712	303753
0 –2 1	nV / (nS – n0)	p^+q^-/p^-q^+	412578	0.393	7.7934	0
0 –2 2	nV / (nS – n0)	p^+/p^-	275848	0.6328	11.1462	136730

Combinación de parámetros	Función lambda	p^+	# Combinaciones de palabras = combinaciones útiles	Media	Desviación estándar	# Combinaciones de palabras descartadas
0 –2 3	nV / (nS – n0)	p^+	74541	0.0014	0.3025	338037
0 –2 4	nV / (nS – n0)	p^+/q^+	142870	1.2918	16.1619	269708

El valor importante a observar, tomado de [GALICIA, 00], es el rango medio de la variante correcta. Para determinar este valor se considera que:

- Si la posición *x* de la variante correcta es igual a la posición número 1, entonces el rango medio es 0.
- Si la posición *x* de la variante correcta es igual a la posición *n*, con *n* = número de variantes obtenidas en el análisis de una oración *s*, entonces el rango medio es 1.
- Para todos los demás valores de 1 hasta *n*, donde aparezca la variante correcta, respecto al conjunto de variantes, el rango medio de la variante con posición *x* es $(x-1)/(n-1)$.

$$rango\_medio = \begin{cases} 0 & x = 1 \\ 1 & x = n \\ (x-1)/(n-1) & 1 < x < n \end{cases} \quad (14)$$

Tabla 9. Resultados obtenidos con la combinación lambda=n_S-n_0 y p^+q^- / p^-q^+.

Oración	Posición variante correcta [GALICIA,00]	Rango medio [GALICIA,00]	#Total de variantes [GALICIA,00]	Posición variante correcta	Rango medio	#Total de variantes
1	2	50%	3	1	0%	7
2	1	0%	14	1	0%	40
3	4	15%	20	---	---	mal analizada
4	5	9%	44	---	---	mal analizada
5	5	26%	14	---	---	mal analizada
6	1	0%	2	1	0%	4
7	1	0%	169	3	0%	386
8	3	100%	3	2	50%	4
9	669	40%	1660	---	---	mal analizada
10	25	5%	480	---	---	mal analizada
11	73	61%	118	9	1%	552
12	---	---	mal analizada	---	---	mal analizada
13	441	13%	3144	---	---	mal analizada
14	555	59%	936	52	2%	2340
15	3	4%	48	---	---	mal analizada
16	2	33%	4	1	0%	6
17	---	--	mal analizada	8	43%	17

Oración	Posición variante correcta [GALICIA,00]	Rango medio [GALICIA,00]	#Total de variantes [GALICIA,00]	Posición variante correcta	Rango medio	#Total de variantes
18	1	0%	42	14	62%	22
19	1	0%	10	4	100%	4
20	1	0%	288	105	52%	200
21	1	0%	6	1	0%	4
22	28	31%	88	4	2%	122
23	25	14%	170	25	42%	58
24	17	41%	40	---	---	mal analizada
25	---	---	160200	2	14%	8
26	1	0%	12	---	---	mal analizada
27	1	0%	6	2	7%	15
28	1	0%	15	---	---	mal analizada
29	---	---	mal analizada	1	0%	18
30	---	---	mal analizada	1	0%	36
31	4	42%	8	3	66%	4
32	---	---	mal analizada	---	---	mal analizada
33	---	---	mal analizada	---	---	mal analizada
34	---	---	mal analizada	---	---	mal analizada
35	---	---	una variante	101	75%	134
36	---	---	mal analizada	---	---	mal analizada
37	1	0%	18	6	4%	112
38	1	0%	11	7	30%	21
39	---	---	una variante	---	----	una variante
40	5	12%	32	1	0%	96
41	---	---	mal analizada	---	---	mal analizada
42	---	---	mal analizada	1	0%	12

Oración	Posición variante correcta [GALICIA,00]	Rango medio [GALICIA,00]	#Total de variantes [GALICIA,00]	Posición variante correcta	Rango medio	#Total de variantes
43	---	---	mal analizada	---	---	mal analizada
44	---	---	mal analizada	---	---	una variante
45	1	0%	26	1	0%	23
46	5	17%	24	10	91%	40
47	---	---	mal analizada	---	---	mal analizada
48	5	57%	8	3	66%	4
49	1	0%	16	1	0%	6
50	---	---	mal analizada	---	---	mal analizada
51	1	0%	4	5	57%	8
52	---	---	mal analizada	---	---	mal analizada
53	19	56%	33	5	23%	18

Entonces, en este primer conjunto de datos, obtenemos un promedio de rango medio de colocación del 26%.

Porcentaje de oraciones mal evaluadas:	73%
Porcentaje de oraciones que están bien evaluadas en el método actual y mal evaluadas en [GALICIA, 00]:	22%
Porcentaje de oraciones que están mal evaluadas en el método utilizado y bien evaluadas en el método tomado como referencia:	41%

Es conveniente mencionar que de las oraciones que resultaron mal analizadas:

– Veintidós oraciones resultaron en análisis sintáctico fallido, sin embargo por

consistencia entre esta variante del método y el método original, se incluyen en la tabla de resultados. Por lo tanto consideramos que el algoritmo puede alcanzar mejores rangos medios.

– Dos oraciones generaron únicamente una variante, por tanto no requieren desambiguación.

CAPÍTULO 5 CONCLUSIONES Y TRABAJO FUTURO

En este capítulo se dan las conclusiones y algunas sugerencias para trabajo futuro.

5.1 Conclusiones y aportaciones

Se modificó el método propuesto por S. Galicia-Haro, para considerar colocaciones en lugar de los patrones de manejo sintáctico como lo considera el método original. Con esta modificación, el algoritmo genera un diccionario de combinaciones de palabras en español en lugar de un diccionario de patrones de manejo sintáctico como el método original. Se hicieron experimentos para valorar los efectos de dicha modificación.

Respecto al uso del diccionario para la desambiguación de árboles sintácticos, se observó que existen algunas combinaciones que no son significativas para la desambiguación. Queda como trabajo futuro el evaluar el valor de la constante lambda para ajustar su valor hacia un valor que ayude a suavizar los valores en estos casos.

Respecto a la evaluación del método elegido para generar la información estadística aplicando el algoritmo para desambiguación sintáctica de las variantes de análisis, consideramos que se requiere refinar el criterio para generar las combinaciones de variantes pero el sistema presenta comportamiento parecido al del método tomado como referencia.

La principal aportación de esta tesis fue comparar el método Galicia-Haro, et al. con la modificación propuesta, donde observamos que el rango medio de colocación del árbol correcto es mejor utilizando patrones de manejo sintáctico.

5.2 Trabajo futuro

Además de presentar el método específico de ponderación de las variantes tomado como referencia en esta tesis, el trabajo de Galicia-Haro plantea un marco de trabajo con determinación del peso final de la variante por votación de diferentes métodos. Sin embargo, no presenta una implementación de esta idea ya que solamente cuenta con un votante. Se puede considerar el presente trabajo como un votante más, combinándolo (en un trabajo futuro) con el método original de Galicia-Haro.

Para mayor cobertura del diccionario, se requiere utilizar más de un corpus. Es una propuesta del trabajo futuro en esta tesis, robustecer el diccionario compilando también el corpus Cas3lb.

Una propuesta más ambiciosa es tomar el corpus desarrollado en [CHANONA, 02] ó algún corpus de mayor tamaño ó basado en Internet y generar un conjunto de variantes con mayor cobertura.

Es importante desarrollar un modelo unificado que considere la información léxica, sintáctica y semántica para la desambiguación.

El algoritmo depende en un gran porcentaje de cómo se formaron las combinaciones de palabras y los filtros que se aplicaron para considerar usarlas o descartarlas. Si estas opciones fueran parametrizables, entonces podrían generarse más espacios de búsqueda de entradas al diccionario y medir su utilidad respecto de porcentaje de oraciones correctamente analizadas respecto a la ambigüedad.

Aunque por la naturaleza del modelo no se requiere interacción, ya que se considera que la parametrización ayudaría en el espacio de búsqueda de modelos para la desambiguación se concluye que se requiere de una interfaz gráfica para

captura de los parámetros.

El uso de las diferentes fórmulas propuestas en [GALICIA, 00] para la asignación de pesos, nos acercaría a la delimitación del tipo de información estadística que es conveniente utilizar al evaluar métodos de aprendizaje automático orientados al PLN.

5.3 Publicaciones

Del presente trabajo se generó la siguiente publicación:

Tania Lugo-Garcia, Alexander Gelbukh, Grigori Sidorov. *Unsupervised Learning of Word Combinations for Syntactic Disambiguation*. Workshop on Human Language Technologies at the ENC-2004, 5th Mexican International Conference on Computer Science. Avances en la Ciencia de la Computación, ISBN 970-692-170-2, pp. 311–318.

BIBLIOGRAFÍA

[ALDEZABAL, 01] ALDEZABAL, I.,ARANZABE, M., ATUXTA, A., GONJEOLA, K. Y SARASOLA, K. *Extracción masiva de información sobre subcategorización verbal vasca a partir de corpus.*

[ALLEN, 95] ALLEN, J. *Natural Language Understanding*. Benjamin Cummings. E.E.U.U., 1995.

[ATLE, 06] ATLE-GULLA, J., OLAF-BORCH y H., ESPEN-INGVALDSEN, J. *Unsupervised Keyphrase Extraction for Search Ontologies*. In Proceedings of Natural Language Processing and Information Systems, Klagenfurt, Austria, 2006.

[BAEZA, 99] BAEZA-YATES, RICARDO y RIBEIRO-NETO, BERTHIER. *Modern Information Retrieval*. ACM Press, 1999.

[BOLSHAKOV, 04] BOLSHAKOV, IGOR y GELBUKH, ALEXANDER. *Computational Linguistics. Models, Resources, Applications.* INSTITUTO POLITÉCNICO NACIONAL, 2004.

[BOUILLON, 00] BOUILLON, P., BAUD, R., ROBERT, GILBERT y RUCH, PATRICK. *Indexing by statistical tagging.* Proceedings of 5es Journées Internationales d´Analyse Statistique des Données Textuelles. Geneve, Suisse, 2000.

[CASTILLO, 03] CASTILLO-VELÁSQUEZ, FRANCISCO A. *Sistema de Análisis Morfológico para el Español*. Tesis (Maestría en

Ciencias de la Computación). México, D.F., Instituto Politécnico Nacional, Centro de Investigación en Computación, 2003.

[CHANONA, 02] CHANONA-HERNÁNDEZ, LILIANA. *Compilación de un corpus representativo de palabras en español a través de Internet* Tesis (Maestría en Ciencias de la Computación). México, D.F., Instituto Politécnico Nacional, Centro de Investigación en Computación, 2002.

[CHARNIAK, 93] CHARNIAK, EUGENE. *Statistical Language Learning.* MIT Press, Massachussets, E.E.U.U., 1993.

[CRYSTAL, 91] CRYSTAL, D. *A Dictionary of Linguistics and Phonetics*. 3a. Ed. Blackwell, E.E.U.U., 1991.

[CORTÉS, 93] CORTÉS-GARCÍA, ULISES et al. *Inteligencia Artificial*. EDICIONES UPC, 1993.

[FRANZ, 96] FRANZ, ALEXANDER. *Automatic Ambiguity Resolution in Natural Language Processing.* Lecture Notes in Artificial Intelligence 1171. Springer, Alemania, 1996.

[GALICIA, 00] GALICIA-HARO, SOFÍA N. *Análisis sintáctico conducido por un diccionario de patrones de manejo sintáctico para lenguaje español* Tesis (Doctorado en Ciencias de la Computación). México, D.F., Instituto Politécnico Nacional, Centro de Investigación en Computación, 2000.

[GALICIA, 99] GALICIA-HARO, SOFIA N., BOLSHAKOV, IGOR y GELBUKH, ALEXANDER. *Aplicación del formalismo de la teoría Texto ⇔ Significado al análisis de textos en español, introduciendo análisis estadístico.* REDII-CONACYT, Centro de Investigación en Computación,

México, 1999.

[GELBUKH, 02] GELBUKH, ALEXANDER (ED.). *Computational Linguistics and Intelligent Text Processing*. Third International Conference, CICLing 2002. Mexico City, Mexico, Febrero 2002, Proceedings.Springer, México, D.F. 2002.

[HERNÁNDEZ, 04] HERNÁNDEZ-RUBIO, ERIKA. *Compilación automática del diccionario de colocaciones para el español usando estructuras sintácticas*. Tesis (Maestría en Ciencias de la Computación). México, D.F., Instituto Politécnico Nacional, Centro de Investigación en Computación, 2004.

[JACOBS, 93] JACOBS, PAUL S. y RAU, LISA F. *Innovations in text interpretation*. Artificial Intelligence, 1993.

[JURAFSKY, 00] JURAFSKY, DANIEL. y MARTIN, JAMES H. *Speech and Language Processing. An Introduction to Language Processing, Computational Linguistics, and Speech Recognition.* Prentice Hall, Upper Saddle River, New Jersey, 2000.

[KOP, 06] KOP, CHRISTIAN, FLIEDL, GÜNTHER, MAYR, HEINRICH C., MÉTAIS, ELISABETH (EDS.). *Natural Language Processing and Information Systems.* Springer, Klagenfurt, Austria, 2006.

[LEWIS, 98] LEWIS, HARRY R. *Elements of the theory of Computation*. PRENTICE HALL, Upper Saddle River, New Jersey, 1997.

[LÚCIO, 02] LÚCIO-PAOLO, J., CORREIA, M., MAMEDE, N. y HAGÉGE, C. *Using Morphological, Syntactical and*

Statistical Information for Automatic Term Acquisition. In Proceedings of Third International Conference, PorTAL 2002, Lisboa, Portugal.

[MANNING, 00] MANNING, CHRISTOPHER D. y SCHÜTZE, HINRICH. *Foundations of Statistical Natural Language Processing.* The MIT Press, Cambridge, Massachusetts, E.E.U.U. 2000.

[MEL'CUK, 88] MEL'CUK, I. A*. Dependency Syntax: Theory and Practice.* State University of New York Press, Albany, E.E.U.U., 1988.

[MILLER, 93] MILLER, GEORGE A., BECKWITH, RICHARD., FELLBAUM, CHRISTIANE., and MILLER, KATHERINE. *Introduction to WordNet: An On-line Lexical Database.* Five papers on WordNet, 1993.

[MOMPÍN, 87] MOMPÍN-POBLET, JOSÉ. *Inteligencia Artificial. Conceptos, técnicas y aplicaciones.* Editorial Marcombo, Barcelona, España, 1987.

[RANCHHOD, 02] RANCHHOD, ELISABETE y MAMEDE, NUNO J (EDS.). *Advances in Natural Language Processing. Tirad International Conference, PorTAL 2002 Proceedings*. Springer, Lisboa, Portugal, 2002.

[SCHIEBER, 89] SCHIEBER, FRANZ. *Introducción a los Formalismos Gramaticales de Unificación.* 1ª Edición, Editorial Teide, Barcelona, España, 1989.

[SPARCK, 96] SPARCK-JONES, KAREN y GALLIERS, JULIA R*. Evaluating Natural Language Processing Systems*. *An analysis and Review*. Springer-Verlag, Berlin Heiderberg,

Alemania, 1995.

[WINOGRAD, 83] WINOGRAD, TERRY*. Language as a cognitive process*. *Volume I: Syntax.* Addison-Wesley Publishing Company, E.E.U.U., 1983.

ANEXO A. GRAMÁTICA GENERATIVA USADA

La información aquí presentada fue tomada de [GALICIA,00].

Reglas de la gramática

S

-> [BEG_S] @:LIS_CLAUSE END_S *# una o más* CLAUSE

LIS_CLAUSE

-> [coor_conj:CONJ] @:CLAUSE [SEP_O coor_conj_LIS_CLAUSE]

 # ella dice, ella hace

-> coor_conj:LIS_CLAUSE [SEP_O] @:CONJ [SEP_O] coor_conj_LIS_CLAUSE

 # y ella busca

-> @:LIS_CLAUSE coor_conj:LIS_CLUSE

 # cuando llegaron el hecho estaba consumado

CLAUSE

-> [coor_conj:CONJ] @:CLAUSIN

-> @:CLAUSE [SEP_O] cir:CIR

 # El investigador descubre algunas cosas, de ves en cuando….

-> cir:CIR [SEP_O] @:CLAUSE

 # Entre semana, por decisión del jefe, estarán restringidos

CLAUSIN

-> [subj:LIS_NP(nmb,gnd,pers)] @:VP(nmb,pers,mean)

 # El invetigador descubre algunas cosas

-> [subj:LIS_NP(nmb,gnd,pers)] [SEP_O] [adver:ADVP [SEP_O] @:VP(nmb,pers,gnd,AUX)

 # Los sobres, comúnmente blancos, son ahora membretados

-> subj:LIS_NP(nmb,gnd,pers) SEP_O [adver:ADV [SEP_O] @:VP(nmb,pers,mean)

 # El investigador, frecuentemente descubre algunas cosas

SEP_O

->’,’|’:’|’;’|’…’|’(‘|’””|’-‘|’)’

END_S

->’-‘|’)’|’!’|’”’ | ‘?’|’.’

BEG_S

->’-‘ | ‘¿’| ‘¡’

CONJ

->CONJ_C *# y, o, sino, pero*

->CONJ_SUB *# si, porque sea, ya*

(10) -> @:CONJ_C CONJ_SUB *# sino porque*

->’…’

GERP

->@:GER *# caminando*

->@:GER dobj:NP(nmb,gnd,pers)[obj:PP] *# brincando una barda*

->@:GER obj:LIS_PP [dobj:NP(nmb,gnd,pers)] *#caminando por el patio*

LIS_GERP

-> @:GERP [‘,’coord_conj:GERP] *# caminando corriendo*

->LIS_GER @:CONJ coord_conj:GERP

 # caminado, corriendo y saltando

CIR

->@:ADVP *# mal, durante meses*

-> [mod:´todo´ @:NP_TIE(nmb,gnd,pers) *# toda esta semana*

-> @PR prep:NP_TIE(nmb,gnd,pers) *# y a los dos dias*

-> @:PR pre:HACE_TIE *# desde hace una semana*

-> @:LIS_GERP

 # …y maquinando trastadas en grupo

-> @_LIS_PP [mod:ADV]

 # En Okinawa, a finales de la segunda guerra, cuando…

(20) -> @:LIS_NP(nmb,gnd,pers)

 # Dos edificios antes, junto a una tienda, venden ..

HACE_TIE

-> @:'hace' NP_TIE(nmb,gnd,pers)

LIS_NP(nmb,gnd,pers)

-> @:NP(nmb,gnd,pers)

LIS_NP(PL,gnd,pers)

-> @:NP(nmb,gnd)','coord_conj:LIS_NP(nmb1,gnd1)*# bajo, gordo, rechoncho*

->LIS_NP(nmb1,gnd1) @:CONJ coord_conj:NP(nmb,dng)

 # La mezquindad, el afán crítico, y la envidia de sus semejantes

(10) ->LIS_NP(nmb1,gnd1,pers) @:CONJ &coor_conj:PP

 # La mezquindad, el afán crítico, y hasta la envidia de sus semejantes

NP(nmb,gnd,pers)

->[det:DETER(nmb,gnd)] @:NOM(nmb,gnd,pers)*# los científicos americanos*

-> @:PPR_ID(nmb,gnd,pers) [prep:PP] *#muchas | muchas de ellas*

-> @:PPR_IT(nmb,gnd,pers) [prep:PP] *#quién | quién de ellas*

-> @:PPR(nmb,gnd,pers) *# ella*

-> [det:DETER(nmb,gnd) @:'cual' *# lo cual | las cuales*

-> [det:DETER(nmb,gnd) @:PPR_PO(nmb,gnd,pers)*# lo suyo | las suyas*

-> [det:DETER(nmb,gnd) @:PPR_N(nmb,gnd,pers) *# la primera*

-> mod:'todo' @:NP(nmb,gnd,pers) *# todos los mercados*

->""@:NOM(nmb,gnd,pers)"" *# "feliz"*

(10) -> &det:DETER(nmb,gnd) @:N(nmb,gnd,pers) pred:PP &comp:AP(nmb,gnd,pers)

 # un libro de cuentos desgastado por los años – aceite de oliva con residuos

-> [&det:DETER(nmb,gnd)] @:NOM((nmb,gnd,pers) [','] pred:LIS_PP[',']

 # el primer día del sol y de viento

-> @:DETER((nmb,gnd)pred:PP *# el de las rosas*

-> mod:AP(nmb.gnd) @:NOM(nmb,gnd,pers) *#amplias zona de árboles*

(5) ->det:DETER(nmb,gnd) @:AP(nmb,gnd) [pred:PP] *# el rojo*

(20) -> @:NOM(nmb,gnd,pers) mod:NOM(nmb1,gnd1,pers1)*# pilas botón*

NOM(nmb,gnd,pers)

-> [num:NUM(nmb)] @:N(nmb,gnd,pers) *# 5000 años*

-> @:N(nmb,gnd,pers) [‘,’] mod:AP(nmb,gnd) [‘,’] *#noticiario, televisivo,*

->mod:AP(nmb,gnd) @:N(nmb,gnd,pers) *# alguna galaxia*

-> @:N(nmb,gnd,pers) pred:PP *# aceite de oliva*

(15) -> @:N(nmb,gnd,pers) comp:N(nmb,gnd,pers) [mod:AP(nmb,gnd)]

 # tiempos más lejanos

-> mod:AP(nmb,gnd) @:N(nmb,gnd,pers) mod:AP(nmb,gnd) *# única mano valida*

->NUM(nmb)*# 5000*

-> INFP *#comprar una torta, beber un jarrito y escuchar rock*

PPR(nmb,gnd,pers)

->PPR_D(nmb,gnd,pers) *# éste | estos*

->PPR_PE(nmb,gnd,pers) *# ello | él*

->PPR_R(nmb,gnd,pers) *# cuya | mismo*

DETER(nmb,gnd)

->DET(nmb,gnd) *# aquel*

->ART(nmb,gnd) *# el, un*

AP(nmb,gnd)

-> @:ADJ(nmb,gnd) comp:ADJ(nmb,gnd) *# antitelevisiva tradicional*

-> @:ADJ(nmb,gnd) adver:ADV *#racial extremadamente*

->mod:ADV @:ADJ(nmb,gnd *# muy feliz*

-> @:ADJ(nmb,gnd) [‘,’ comp.:AP(nmb,gnd) *# racial, sexual o fisica*

-> AP(nmb,gnd) pred:LIS_PP *# lleno de …*

PP

-> @:PR prep:LIS_NP(nmb,gnd, pers) *# de la tal señora*

> @:QUE

> @:PR pre:QUE *# de que se enojaba*

> @:PR pre:INFP *# de caminar una hora*

(10) -> @:PR pre:CLAUSE *# de no se que señora*

QUE

-> @:'que' pre:CLAUSE *# que se enojaba*

-> @:'que' pre:NP(nmb,gnd,pers) *# que la señora*

LIS_PP

-> @:PP [',' coord_conj:LIS_PP] *#en noticiarios televisivos, en diarios, en..*

-> @ LIS_PP @:CONJ coord_conj:PP *# al patio trasero y a la escalera*

(30) -> @:CONJ coord_conj:PP ',' *# ni en espectáculos, ni en conseguir que….*

ADVP

-> ADV *#bueno | malo*

-> @:PR adver:ADV [mod:ADV] *# por atrás*

-> @:ADV comp:NP_TIE(nmb,gnd,pers1) *# durante meses*

-> @:ADV mod:ADJ(nmb,gnd) *# tanto major*

-> @:HACE_TIE *# hace un año*

-> @:ADV adver:ADV *# incluso mas*

(10) -> @:ADV comp:NP(nmb,gnd,pers) comp:QUE_NP *#más bajo que alto*

-> @:ADV mod:PP *# incluso ese día*

(10) -> @:PP *# a decir verdad*

(10) -> @:ADV comp:NP(nmb,gnd,pers) *# como un rosario*

(20) -> @:ADJ *# feliz*

QUE_NP

-> @:'que' prep:NP(nmb,gnd,pers) *# que aquel hombre*

NP_TIE(nmb,gnd,pers)

-> [[mod:'tod'] det:DETER(nmb,gnd)] @NOM_TIE(nmb,gnd,pers) # *todo el día*

-> det:DETER(nmb,gnd) @:NOM_TIE(nmb,gnd,pers) prep:PP # *el día de la bandera*

NOM_TIE(nmb,gnd,pers)

-> cuant:NUM(nmb)[mod:AP(nmb,gnd)] @:N_TIE(nmb,gnd,pers) [mod:AP(nmb,gnd)] # *dos largos años grises*

N_TIE(nmb,FEM,3PRS)

-> @:'semana'| @:'hora'| @:'mañana'| @:'tarde'|@:'noche'

N_TIE(nmb,MASC,3PRS)

-> @:'día'|@:'año'| @:'mes' | @:'ayer' | @:'siglo'|@:'minuto'|@:'milenio| @:'decenio'

-> @:'lunes'| @:'martes'|@:'miércoles'|@:'jueves'|@;'sabado'|@:'domingo'

-> @:'febrero'| @:'enero'|@:'marzo'|@:'abril'|@;'mayo'|@:'junio'

-> @:'julio'| @:'agosto'|@:'septiembre'|@:'octubre'|@;'noviembre'| @:'diciembre'

#**********

 # Grupo del verbo

#**********

VP_MODS

-> ADVP

-> @:LIS_GERP

VP(nmb,pers,gnd,AUX)

-> [clit:PPR_C(nmb1,gnd1,pers1)] @:VERB(nmb,pers,AUX) [mod:ADV] [dobj_suj:NP(nmb,gnd,pers)]

 # era pariente de

-> [clit:PPR_C(nmb1,gnd1,pers1)]@:VERB(nmb,pers,AUX) [mod:ADV] dobj:AP(nmb,gnd)

 # es fatal

-> @:VERB(nmb,pers,AUX) [mod:ADV] dobj:N(nmb,gnd,pers) obj:PP

 # hay vida en alguna

-> @:VERB(nmb,pers,AUX) [mod:ADV] obj:PP dobj:N(nmb,gnd,pers)

 # hay en algún lugar una escuela…

VERB(nmb,pers,AUX)

-> VIN(nmb,pers,AUX)| VCO(nmb,pers,AUX)| VSJ (nmb,pers,AUX)

-> [clit:PPR_C(nmb1,gnd1,pers1)] @:'haber' [adver:ADVP] PART(SG,MASC,AUX) PART (nmb,gnd)

 # le había sido visto

-> @:'haber' aux:NP(nmb,gnd,3prs) *# había testigos*

VP(nmb,pers,mean)

-> VP_DOB(nmb,pers,mean)

-> VP_OBJS(nmb,pers,mean)

VP_DOBJ(nmb,pers,mean)

-> @:VP_OBJS(nmb,pers,mean) obj:LIS_NP(nmb1,gnd1,pers1)

 # claban sus dardos

-> @:VP_DOBJ(nmb,pers,mean) comp:LIS_PP

 # trasladó su fábrica a la frontera

-> @:VP_DOBJ(nmb,pers,mean) mod:VP_MODS

 # ordenó una fila moviendo la sillas

SUJ_DOBJ

-> @:'al' prep:NP(nmb,gnd,pers)

-> @:'a' prep: :NP(nmb,gnd,pers)

-> @:NP(nmb,gnd,pers)

VP_OBJS(nmb,pers,mean)

-> [adver:ADV] @:VP_V(nmb,pers,mean) [mod:VP_MODS] *#provocaban en su mente*

-> [adver:ADV] @:VP_V(nmb,pers,mean) [obj:LIS_PP] *#salieron del corral*

-> @:VP_OBJS(nmb,pers,mean) obj:LIS_PP

 # clavaban sus dardos por todo el cuerpo

-> @:VP_OBJS(nmb,pers,mean) mod:VP_MODS

 # jugaban el ultimo partido provocándose a cada momento

VP_V(nmb,pers,mean)

-> [clit:PPR_C(nmb1,gnd1,pers1)[clit:PPR_C(nmb2,gnd,pers2)] @:VP_SV(nmb,pers,mean)

 # se le llamase, se les haya dicho

VP_SV(nmb,pers,mean)

-> @:VERB(nmb,pers,mean) *# creo*

-> @:'haber'(nmb,pers) [adver:ADVP] PART(SG,MASC) *# había incluso dudado*

-> @:'estar' [&adver:ADVP] AP(nmb,gnd) *# estaba contento*

-> @:'estar'(nmb,pers) [adver:ADVP] PART (nmb,gnd) *# está mal visto*

-> @:'ser'(nmb,pers) [adver:ADVP] PART(nmb,gnd)

 # es folicularmente handicapado

PPR_PE(nmb,gnd,3PRS)

(10) ->'usted'

VERB(nmb,pers,mean)

->VIN(nmb,pers,mean)| VCO(nmb,pers,mean)| VSJ(nmb,pers,mean)

PPR_PE(NMB,GND,3PRS)

->'usted'

#**********

INFP

-> @:VP_INF[SEP_O coord_conj:INFP] *# cantar, reir*

-> INFP@:CONJ coord_conj:VP_INF *# vivir morir*

-> [adver:ADV] @:V(INF,AUX)[adver:ADV] *# morir tambien*

VP_INF

-> @:VP_INF_DOBJ *# convertir la bandera de los rayos en oficial*

-> @VP_INF_OBJS *# ir a la cárcel*

VP_INF_DOBJ

-> @:VP_INF_OBJS [‘,’] dobj_suj:SUJ_DOBJ[dobj_suj:SUJ_DOBJ]

 # dar su consentimiento

-> @:VP_INF_DOBJ[‘,’] obj:LIS_PP

 # introducir unos centímetros en su interior

-> @:VP_INF_DOBJ[‘,’] mod:VP_MODS

 # decir una palabra negando su sentir

VP_INF_OBJS

-> @:V_INF *# esperar pacientemente*

-> @:VP_INF_OBJS [‘,’] obj:LIS_PP *# marchar hasta...*

-> @:VP_INF_OBJS [‘,’] mod:VP_MODS *#marchar torciendo…*

V_INF

-> [adver:ADV] @:V(INF,mean) [adver:ADV] *# no estar hoy*

-> [adver:ADV] @:’haber’(INF) [adver:ADV] PART(SG,MASC) [adver:ADV]

 # no haber presentado puntualmente, había siempre quedado..

-> [adver:ADV] @:’ser’(INF) [adver:ADVP] PART (nmb,gnd) [adver:ADV]

 # ser entrevistada

ANEXO B. PARÁMETROS DE ANÁLISIS SINTÁCTICO DEL PARSER

Para generar el análisis sintáctico se utiliza la herramienta PARSER. Esta herramienta realiza el análisis en base a la gramática descrita en el Anexo A. Gramática generativa.

Las opciones configurables del PARSER están en el menú file->options, y la interfaz es la siguiente:

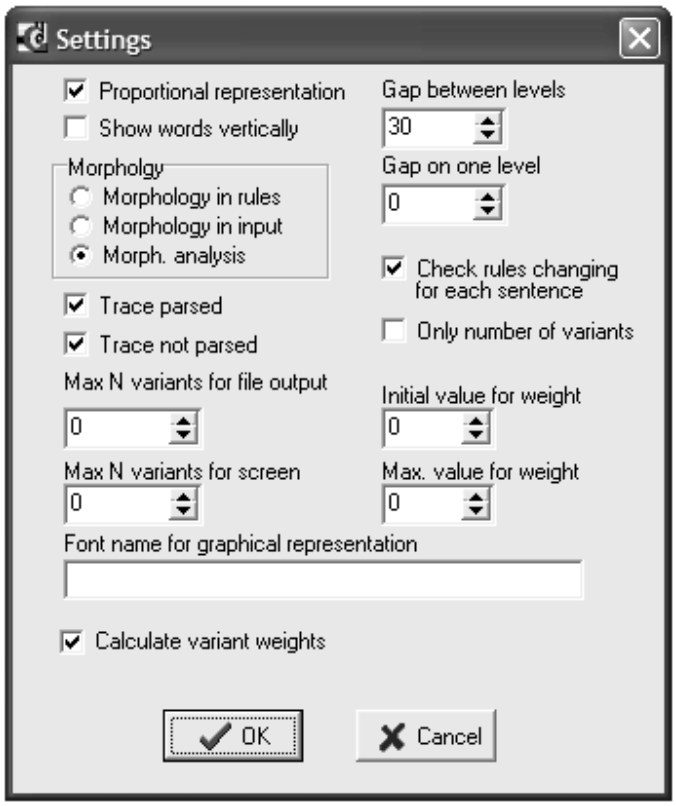

Figura 17. Opciones configurables del PARSER

Como dato importante se describe la configuración de las opciones del PARSER, que son las descritas a continuación.

- Representación proporcional: Afecta la forma en que se centran los nodos del árbol en el modo gráfico.
- Mostrar palabras verticalmente: Esta opción afecta la forma en que las palabras se presentan en el árbol en el modo gráfico.
- Configuración de morfología: Determina el formato esperado del archivo de texto de entrada. Existen tres formatos:

Morfología en las reglas: Se espera que todas las palabras del archivo sean nodos terminales de la gramática, es decir, no se utiliza ningún marcado morfológico.

Morfología en la entrada: El archivo de entrada tiene una forma estructurada, con los códigos morfológicos explícitamente asignados a cada palabra.

Morfología en el análisis: En este modo la entrada es un texto llano, y el programa lo analizará morfológicamente.

- Dibujar análisis sintáctico: Si la opción dibujar análisis está seleccionada, las sentencias analizadas exitosamente serán dibujadas en la página de dibujo del sistema.
- No dibujar análisis sintáctico: Si la opción no dibujar análisis está seleccionada, las sentencias para las cuales el análisis falla serán dibujadas en la página de dibujo.
- Niveles entre intervalos: Esta opción afecta la forma en que los nodos del árbol se colocan en el modo gráfico.

- Nivel en un intervalo: Esta opción afecta la manera en que se colocan los nodos en el modo gráfico.

- Verificar reglas en cada cambio de frase: Permiten cambiar la gramática sin volver a cargar el programa.

- Número único de variantes: Permite saltarse la fase de cargar las variantes encontradas dentro del visor del programa.

- Máximo número de variantes en entrada: En lugar de cargar las variantes encontradas dentro del visor del programa, solo se cargan el número de variantes dado para cada sentencia.

Esta información fue tomada de [HERNÁNDEZ, 04].

ANEXO C. ORACIONES UTILIZADAS EN LA EVALUACIÓN

¡ Llamaré a la policía!

Decidió que haría pintar la casa.

Pero Irene la detuvo con un gesto.

No le hace mal a nadie - sonrió.

Beatriz abandonó su puesto de observación mordiéndose los labios.

Este negocio no ha resultado ninguna maravilla.

Voy a entrevistar una especie de santa.

Dicen que hace milagros.

Beatriz suspiró sin dar muestras de apreciar el humor de su hija.

Tenía el hábito de hablar con Dios.

¿ No podía hacerlo en silencio y sin mover los labios ?

Así sucedía en todas las familias.

No quería dar la impresión de haberla descuidado , porque la gente murmuraría a sus espaldas.

Era un período de reposo , descansaban los campos , los días parecían más cortos , amanecía más_tarde.

Siempre lo dijo , pero nadie le prestó atención.

Tenía un carácter galante.

Al volver los hombres el hecho estaba consumado y debieron aceptarlo.

Luego la envió de regreso a su cama.

¿ Qué pensaría su marido al verla ?

Marchaba a su lado con paso firme en las manifestaciones callejeras.

En íntima colaboración criaron a sus hijos.

Esa criatura rubia de ojos claros tal_vez significaba algo en su destino.

Por allí dicen que se comprarán un tractor.

Aunque vivían a escasa distancia tenían pocas ocasiones de encontrarse , pues

sus vidas eran muy aisladas.

Cumplía múltiples ocupaciones bajo la tienda.

Ella también lo prefería así.

Su mujer nunca pudo recibirlo con naturalidad.

A_diferencia_de otros campesinos , se casaron enamorados y por amor engendraron hijos.

Nada se botaba ni perdía.

Nada podemos hacer.

Su madre recordaba con exactitud el comienzo de la desgracia.

Entretanto los batracios formaron filas compactas y emprendieron marcha ordenadamente.

La crisis duró pocos minutos y dejó a Evangelina extenuada , a la madre y al hermano aterrorizados.

Nos vamos a arruinar.

Pero todo había sido en_vano.

En su presencia se sentía repudiado.

El joven parecía tener las ideas claras y éstas no coincidían con las suyas.

En ese sentido era muy cuidadosa.

Sus abundantes batallas fortalecieron el odio.

Dejaron la perra en la casa , subieron en la motocicleta y partieron.

Apretaban los dientes y aguantaban callados.

Sacó por fin la voz y se presentó.

Poco después apareció Irene_Beltrán y pudo verla de cuerpo entero.

Resultó tal_como la imaginaba.

Irene no terminó el postre , dejando un trozo en el plato.

Pero no fue así.

En sus labios esta investigación adquiría una alba pátina de inocencia.

Nadie en la editorial sospechó del nuevo fotógrafo.

Parecía un hombre tranquilo.

Ni_siquiera Irene supo de su vida secreta , aunque algunos indicios leves estimulaban su curiosidad.

En los meses siguientes se estrechó su relación.

El hombre se puso lentamente de pie y las invitó al interior de su morada.