UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INFORMÁTICOS

Método para la Construcción Automática de
Ontologías Basado en Patrones Lingüísticos

Tesis Doctoral

Miguel Ángel de la Villa Moreno

Licenciado en Informática

2016

DEPARTAMENTO DE INTELIGENCIA ARTIFICIAL
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INFORMÁTICOS

Método para la Construcción Automática de
Ontologías Basado en Patrones Lingüísticos

Autor: **Miguel Ángel de la Villa Moreno**

Licenciado en Informática

Director: **Jesús Cardeñosa Lera**

Doctor en Informática

2016

Tribunal nombrado por el Magfco. y Excmo. Sr. Rector de la Universidad Politécnica de Madrid, el día ____ de _____ de 20__

Presidente _____

Vocal _____

Vocal _____

Vocal _____

Secretario _____

Suplente _____

Suplente _____

Realizado el acto de defensa y lectura de la Tesis el día ____ de _____ de 20__ en la Escuela Técnica Superior de Ingenieros Informáticos.

Calificación: _____

EL PRESIDENTE

LOS VOCALES

EL SECRETARIO

ÍNDICE GENERAL

ÍNDICE DE FIGURAS	III
ÍNDICE DE TABLAS	VI
RESUMEN.....	VIII
ABSTRACT	X
1 ESTADO DEL ARTE.....	1
1.1 INTRODUCCIÓN.....	1
1.1.1 Construcción automática de ontologías	3
1.2 MÉTODOS PARA LA OBTENCIÓN AUTOMÁTICA DE CONCEPTOS.....	3
1.2.1 Reconocimiento de términos	5
1.2.2 Determinación de conceptos	14
1.2.3 Concepto vs Instancia	15
1.3 MÉTODOS PARA LA OBTENCIÓN AUTOMÁTICA DE RELACIONES TAXONÓMICAS	16
1.3.1 Métodos basados en la semejanza semántica	17
1.3.2 Patrones léxico – sintácticos.....	25
1.3.3 Composición de términos.....	31
1.4 OBTENCIÓN DE RELACIONES NO TAXONÓMICAS.....	32
1.4.1 Reglas de asociación.....	32
1.4.2 Patrones léxico – sintácticos.....	35
1.4.3 Preferencias de selección	36
1.4.4 Reglas lingüísticas.....	39
1.5 DEEP ONTOLOGY LEARNING.....	40
1.5.1 Interlinguas.....	41
1.5.2 Patrones lingüísticos en UNL	43
1.6 CONCLUSIONES.....	47
2 PLANTEAMIENTO DEL PROBLEMA	49
3 HIPÓTESIS DE TRABAJO	50
4 MÉTODO PROPUESTO	51

4.1	MARCO FORMAL.....	53
4.1.1	<i>Grafo dirigido con arcos etiquetados.....</i>	53
4.1.2	<i>Grafo dirigido anidado.....</i>	62
4.1.3	<i>Expresión UNL como grafo anidado</i>	67
4.1.4	<i>Camino y distancia en una expresión UNL. Coordinación lingüística.....</i>	69
4.1.5	<i>Patrón lingüístico.....</i>	72
4.2	RESOLUCIÓN DE LA ANÁFORA PRONOMINAL	77
4.2.1	<i>Anáfora en UNL</i>	77
4.2.2	<i>Algoritmo de resolución.....</i>	82
4.3	PREPARACIÓN DE LAS EXPRESIONES.....	107
4.3.1	<i>Incorporación relaciones deducibles.....</i>	107
4.3.2	<i>Reducción de anidamiento</i>	111
4.3.3	<i>Eliminación de stopwords</i>	115
4.4	CONSTRUCCIÓN DE LA ONTOLOGÍA	116
4.4.1	<i>Reconocimiento de términos</i>	117
4.4.2	<i>Reconocimiento de entidades.....</i>	129
4.4.3	<i>Selección de conceptos</i>	133
4.4.4	<i>Relaciones taxonómicas.....</i>	134
4.5	EJEMPLO PASO A PASO	135
5	EXPERIMENTACIÓN	140
6	COMENTARIOS FINALES.....	149
6.1	CONCLUSIONES.....	149
6.2	LÍNEAS FUTURAS	151
7	BIBLIOGRAFÍA.....	152
ANEXO I.	UNIVERSAL NETWORKING LANGUAGE.....	160
ANEXO II.	EJEMPLO DE APLICACIÓN DE ALGORITMO DE RESOLUCIÓN DE ANÁFORA	164
ANEXO III.	LENGUAJE DE ESPECIFICACIÓN DE PATRONES LINGÜÍSTICOS	181
	<i>Especificación del lenguaje en BNF.....</i>	<i>184</i>
ANEXO IV.	PATRONES LINGÜÍSTICOS ESPECIFICADOS EN TEXTO	187
ANEXO V.	RESULTADOS.....	190

ÍNDICE DE FIGURAS

Figura 1. Tipos de ontologías.....	2
Figura 2. Triángulo semiótico de Ogden	4
Figura 3. Instancia - concepto.....	15
Figura 4. Determinación de concepto e instancia	16
Figura 5. Jerarquía de conceptos.....	18
Figura 6. Secuencia de pasos en algoritmo de clústering.....	21
Figura 7. Retículo de conceptos.....	23
Figura 8. Jerarquía de conceptos resultante.....	24
Figura 9. Precisión vs cobertura de patrones léxico-sintácticos.....	31
Figura 10. Jerarquía de términos construida por composición léxica.....	32
Figura 11. Representación de una pregunta en UNL	45
Figura 12. Localización de la respuesta en una expresión UNL	45
Figura 13. Conocimiento factual en expresión UNL.....	46
Figura 14. Patrón lingüístico relacionando personas y ciudades.....	47
Figura 15. Ejemplo de grafo dirigido.....	53
Figura 16. Subgrafos de G	54
Figura 17. Homomorfismo entre grafos.....	56
Figura 18. Subgrafo imagen del homomorfismo de la Figura 17.....	57
Figura 19. Diferentes tipos de homomorfismos	58
Figura 20. Ejemplo de grafo de encaje	60
Figura 21. Ejemplo de encaje entre M_G y G	61
Figura 22. Ejemplo de grafo anidado.....	63

Figura 23. Grafo raíz de un grafo anidado.....	64
Figura 24. Ejemplo de encaje sobre un grafo recursivo	67
Figura 25. Ejemplo de expresión UNL.....	68
Figura 26. Estructuras de coordinación en UNL.....	70
Figura 27. Patrón lingüístico.....	73
Figura 28. Notación simplificada de patrón lingüístico	74
Figura 29. Notación simplificada de restricciones de adyacencia.....	75
Figura 30. Ejemplo de encaje de un patrón lingüístico	76
Figura 31. Ejemplo de encaje de un patrón lingüístico (II)	76
Figura 32. Encaje no válido según restricciones de adyacencia.....	77
Figura 33. Expresión UNL con relación de posesivo	78
Figura 34. Expresión UNL con anáfora	79
Figura 35. Eliminación de pronombre anafórico.....	79
Figura 36. Aparición de pronombre anafórico por inserción de <i>scope</i>	80
Figura 37. Expresión UNL	81
Figura 38. Diferentes codificaciones UNL de la misma oración	86
Figura 39. Relaciones de c-comando en un árbol de ejemplo	88
Figura 40. Correspondencias entre el árbol sintáctico y la expresión UNL.....	92
Figura 41. Ejemplo de coordinación lingüística (caso 1)	108
Figura 42. Ejemplo de coordinación lingüística (caso 2)	108
Figura 43. Transformación de la coordinación lingüística (caso I)	109
Figura 44. Transformación de la coordinación lingüística (caso II).....	110
Figura 45. Expresiones transformadas.....	111
Figura 46. Patrón instanciado únicamente en un <i>scope</i>	112
Figura 47. Instanciación de patrón lingüístico tras reducir el anidamiento.....	112
Figura 48. Esquema general de la operación de reducción de anidamiento	114
Figura 49. Ejemplos de términos en UNL	118

Figura 50. Instanciaciones posibles del patrón N + V	122
Figura 51. Patrón con restricciones de adyacencia	123
Figura 52. Núcleos de sintagmas nominales máximos.....	124
Figura 53. Patrón lingüístico con restricción de atributo auxiliar de núcleo	124
Figura 54. Instanciación de patrón en término mayor	125
Figura 55. Patrón lingüístico completamente especificado	126
Figura 56. Patrones lingüísticos para selección de términos.....	133
Figura 57. Ejemplo de descomposición de un término.....	134
Figura 58. Ejemplo de expresión UNL.....	135
Figura 59. Expresión tras aplicar el preprocesamiento.....	136
Figura 60. Identificación de términos y entidades	137
Figura 61. Ontología obtenida de la expresión.....	139
Figura 62. Arquitectura del sistema implementado	141
Figura 63. Fragmento de la ontología obtenida en la experimentación	148
Figura 64. Restricciones y argumentos para las UWs	161
Figura 65. Grafo UNL.....	163
Figura 66. Ejemplo de resolución de anáfora (expresión 1).....	165
Figura 67. Ejemplo de resolución de anáfora (expresión 2).....	169
Figura 68. Ejemplo de resolución de anáfora (expresión 3).....	175
Figura 69. Ejemplo de resolución de anáfora (expresión 4).....	177
Figura 70. Patrón lingüístico de ejemplo (I).....	182
Figura 71. Patrón lingüístico de ejemplo (II).....	183
Figura 72. Patrón lingüístico de ejemplo (III).....	184

ÍNDICE DE TABLAS

Tabla 1. Clasificación de métodos de clústering	20
Tabla 2. Matriz objeto - atributo.....	22
Tabla 3. Patrones léxico - sintácticos.....	26
Tabla 4. Patrones léxico - sintácticos de hiponimia	27
Tabla 5. Patrones léxico - sintácticos de meronimia.....	27
Tabla 6. Patrones lexico – sintácticos para la meronimia.....	35
Tabla 7. Sustantivos lematizados ordenados por relevancia	36
Tabla 8. Clases de la ontología más relevantes tras la etiquetación y el reconocimiento de entidades del corpus	37
Tabla 9. Verbos lematizados ordenados por relevancia.....	37
Tabla 10. Preferencias de selección para los verbos "to cross" y "to test"	38
Tabla 11. Ejemplos de relaciones ontológicas obtenidas.....	38
Tabla 12. Capacidades de UNL como formalismo de representación del conocimiento	44
Tabla 13. Condiciones abreviadas en patrones lingüísticos	74
Tabla 14. Definición de función $I(x)$	91
Tabla 15. Ejemplos de argumentos principales de verbos.....	97
Tabla 16. Ejemplos de argumentos secundarios de verbos.....	98
Tabla 17. Ejemplos de relaciones de aposición.....	99
Tabla 18. Ejemplos de argumentos nominales.....	99
Tabla 19. Ejemplo de nodos con dos funciones.....	100
Tabla 20. Puntuación del candidato según su categoría funcional	100
Tabla 21. Transiciones posibles del foco	104

Tabla 22. Ejemplos de términos en español e inglés	117
Tabla 23. Ejemplos de correspondencias entre patrones de formación de términos.....	120
Tabla 24. Patrones lingüísticos para reconocer términos	127
Tabla 25. Patrones para obtener instancias	130
Tabla 26. Patrones para obtener fechas.....	130
Tabla 27. Patrón para obtener nombres o títulos	131
Tabla 28. Patrón para obtener nombres propios en scopes.....	132
Tabla 29. Estadísticas del corpus relacionadas con la anáfora pronominal.....	143
Tabla 30. Resultados de la resolución de la anáfora	143
Tabla 31. Estadísticas del corpus relacionadas con la preparación de expresiones	144
Tabla 32. Resultados de la preparación de expresiones	144
Tabla 33. Resultados de la identificación de términos y entidades	145
Tabla 34. Términos seleccionados.....	146
Tabla 35. Recuento de componentes de la ontología final	147
Tabla 36. Antecedentes de los pronombres del ejemplo	164

La tesis que se presenta tiene como propósito la construcción automática de ontologías a partir de textos, enmarcándose en el área denominada *Ontology Learning*. Esta disciplina tiene como objetivo automatizar la elaboración de modelos de dominio a partir de fuentes de información estructurada o no estructurada, y tuvo su origen con el comienzo del milenio, a raíz del crecimiento exponencial del volumen de información accesible en Internet. Debido a que la mayoría de la información se presenta en la web en forma de texto, el *aprendizaje automático de ontologías* se ha centrado en el análisis de este tipo de fuente, nutriéndose a lo largo de los años de técnicas muy diversas provenientes de áreas como la Recuperación de Información, Extracción de Información, Sumarización y, en general, de áreas relacionadas con el procesamiento del lenguaje natural.

La principal contribución de esta tesis consiste en que, a diferencia de la mayoría de las técnicas actuales, el método que se propone no analiza la estructura sintáctica superficial del lenguaje, sino que estudia su nivel semántico profundo. Su objetivo, por tanto, es tratar de deducir el modelo del dominio a partir de la forma con la que se articulan los significados de las oraciones en lenguaje natural. Debido a que el nivel semántico profundo es independiente de la lengua, el método permitirá operar en escenarios multilingües, en los que es necesario combinar información proveniente de textos en diferentes idiomas.

Para acceder a este nivel del lenguaje, el método utiliza el modelo de las *interlinguas*. Estos formalismos, provenientes del área de la traducción automática, permiten representar el significado de las oraciones de forma independiente de la lengua. Se utilizará en concreto UNL (Universal Networking Language), considerado como la única interlingua de propósito general que está normalizada.

La aproximación utilizada en esta tesis supone la continuación de trabajos previos realizados tanto por su autor como por el equipo de investigación del que forma parte, en los que se estudió cómo utilizar el modelo de las interlinguas en las áreas de extracción y recuperación de información multilingüe.

Básicamente, el procedimiento definido en el método trata de identificar, en la representación UNL de los textos, ciertas regularidades que permiten deducir las piezas de la ontología del dominio. Debido a que UNL es un formalismo basado en redes semánticas, estas regularidades se presentan en forma de grafos, generalizándose en estructuras denominadas *patrones lingüísticos*.

Por otra parte, UNL aún conserva ciertos mecanismos de cohesión del discurso procedentes de los lenguajes naturales, como el fenómeno de la anáfora. Con el fin de aumentar la efectividad en la comprensión de las expresiones, el método provee, como otra contribución relevante, la definición de un algoritmo para la resolución de la anáfora pronominal circunscrita al modelo de la interlingua, limitada al caso de pronombres personales de tercera persona cuando su antecedente es un nombre propio.

El método propuesto se sustenta en la definición de un marco formal, que ha debido elaborarse adaptando ciertas definiciones provenientes de la teoría de grafos e incorporando otras nuevas, con el objetivo de ubicar las nociones de expresión UNL, patrón lingüístico y las operaciones de encaje de patrones, que son la base de los procesos del método.

Tanto el marco formal como todos los procesos que define el método se han implementado con el fin de realizar la experimentación, aplicándose sobre un artículo de la colección EOLSS “Encyclopedia of Life Support Systems” de la UNESCO.

ABSTRACT

The purpose of this thesis is the automatic construction of ontologies from texts. This thesis is set within the area of *Ontology Learning*. This discipline aims to automatize domain models from structured or unstructured information sources, and had its origin with the beginning of the millennium, as a result of the exponential growth in the volume of information accessible on the Internet. Since most information is presented on the web in the form of text, the *automatic ontology learning* is focused on the analysis of this type of source, nourished over the years by very different techniques from areas such as Information Retrieval, Information Extraction, Summarization and, in general, by areas related to natural language processing.

The main contribution of this thesis consists of, in contrast with the majority of current techniques, the fact that the method proposed does not analyze the syntactic surface structure of the language, but explores his deep semantic level. Its objective, therefore, is trying to infer the domain model from the way the meanings of the sentences are articulated in natural language. Since the deep semantic level does not depend on the language, the method will allow to operate in multilingual scenarios, where it is necessary to combine information from texts in different languages.

To access to this level of the language, the method uses the *interlingua* model. These formalisms, coming from the area of machine translation, allow to represent the meaning of the sentences independently of the language. In this particular case, UNL (Universal Networking Language) will be used, which considered to be the only interlingua of general purpose that is standardized.

The approach used in this thesis corresponds to the continuation of previous works carried out both by the author of this thesis and by the research group of which he is part, in which it is studied how to use the interlingua model in the areas of multilingual information extraction and retrieval.

Basically, the procedure defined in the method tries to identify certain regularities at the UNL representation of texts that allow the deduction of the parts of the ontology of the domain.

Since UNL is a formalism based on semantic networks, these regularities are presented in the form of graphs, generalizing in structures called *linguistic patterns*.

On the other hand, UNL still preserves certain mechanisms of discourse cohesion from natural languages, such as the phenomenon of the anaphora. In order to increase the effectiveness in the understanding of expressions, the method provides, as another significant contribution, the definition of an algorithm for the resolution of pronominal anaphora limited to the model of the interlingua, in the case of third person personal pronouns when its antecedent is a proper noun.

The proposed method is based on the definition of a formal framework, adapting some definitions from Graph Theory and incorporating new ones, in order to locate the notions of UNL expression and linguistic pattern, as well as the operations of pattern matching, which are the basis of the method processes.

Both the formal framework and all the processes that define the method have been implemented in order to carry out the experimentation, applying on an article of the "Encyclopedia of Life Support Systems" of the UNESCO-EOLSS collection.

1.1 Introducción

La sobrecarga de información que ha provocado el enorme y rápido crecimiento de internet, ha tenido como consecuencia el aumento de la investigación en sistemas de extracción de información que procesen el contenido de la Web de forma eficiente y práctica. Sin embargo, el inmenso volumen de datos que se trata de manejar convierte a esta tarea en un reto difícil de abordar. Según una encuesta de mercado realizada por IDC [47], entre 2009 y 2020 la cantidad de información digital crecerá en un factor de 44, mientras que el número de profesionales de IT solo aumentará en un factor de 1.4.

La primera aproximación en acometer este desafío se realizó a comienzos del milenio, planteando el desarrollo de la *Web Semántica*, definida por Berners-Lee y Fichetti [8] como una extensión de la web en la que la información se puede encontrar con un significado bien definido, permitiendo tanto a las personas como a los computadores trabajar en cooperación.

Esta cooperación implica necesariamente el desarrollo de componentes que compartan conocimiento, convirtiendo a las *ontologías* en una pieza clave. Según la definición más aceptada “*an ontology is a formal, explicit specification of a shared conceptualization*” [51].

Con *conceptualización* se quiere decir que una ontología es un modelo abstracto acerca de cómo las personas piensan sobre las cosas del mundo, restringido a un dominio o área particular. Una *especificación explícita* significa que los conceptos y relaciones del modelo abstracto están representados por nombres y definiciones, estas últimas reflejando cómo un término se relaciona con los demás. *Formal* quiere decir que el modelo está especificado mediante un lenguaje con propiedades formales bien definidas. Generalmente se utilizan lenguajes de representación del conocimiento basados en lógica, como *OWL* [73]. Por último,

que sea *compartido* significa que el propósito de una ontología es ser reutilizada por diferentes aplicaciones y comunidades.

En resumen, una ontología intenta especificar, empleando un lenguaje formal, un modelo abstracto que contenga las definiciones consensuadas de los conceptos relevantes de un determinado dominio, junto con las relaciones que se pueden establecer entre ellos. Se encargan de formalizar los aspectos intensionales de un dominio, en el que la parte extensional está constituida por una base de conocimientos [15].

Los conceptos de una ontología generalmente están organizados en una taxonomía de clases y subclases. Cada clase está asociada con varias propiedades, denominadas atributos o *slots*, que describen sus características.

Existen diversos tipos de ontologías, que se diferencian en la manera de especificar los términos [101]. Varían entre las que no especifican en absoluto su significado (*lightweight ontologies*) [49] hasta las que formalizan con rigor teorías lógicas (*heavyweight ontologies*) [43]. La siguiente figura, adaptada de [49], muestra gráficamente la escala de tipos de ontologías:

Figura 1. Tipos de ontologías

Las ontologías han encontrado desde hace tiempo aplicaciones en áreas muy diversas, como la extracción de información [16, 55, 74, 94], la recuperación de información [39, 52, 66, 72], *question answering* [67, 83, 87] o la traducción automática [6, 77, 108].

1.1.1 Construcción automática de ontologías

Tradicionalmente, se ha contemplado la construcción de una ontología como un proceso manual, complicado y tedioso, debido a que se requiere extraer conocimiento de un experto al que, generalmente, le resulta complicado transmitir lo que sabe, y más aun de la manera adecuada para formalizar un modelo.

Por esta razón, durante los últimos 15 años se ha intentado automatizar la construcción de ontologías, utilizando el texto escrito como fuente de información. Esto no es extraño, teniendo en cuenta que el lenguaje natural es la vía que más se ha utilizado para representar el conocimiento (la Web, en su inmensa mayoría, contiene la información en forma de texto).

Con este objetivo, se han recopilado técnicas muy diversas provenientes de diferentes áreas de la IA, como son la extracción de información, el aprendizaje automático, el procesamiento del lenguaje natural, etc., enfocándolas en la construcción automática de ontologías. La combinación de estas técnicas con este nuevo objetivo dio lugar a la disciplina denominada *Ontology Learning* [70].

El resto de secciones tratan de proporcionar una visión general del conjunto de métodos que se emplean para la construcción automática de ontologías. Dada su heterogeneidad, se ha optado por clasificarlos primero por los elementos de la ontología que tratan de descubrir (conceptos, relaciones taxonómicas y relaciones no taxonómicas), y después por la naturaleza de la técnica (clústering, estadístico, lingüístico, etc.).

1.2 Métodos para la obtención automática de conceptos

El primer paso en la construcción de una ontología consiste en determinar los conceptos que la forman. No es de extrañar, por tanto, que la totalidad de métodos para la obtención automática de ontologías existentes aborden esta tarea en primer lugar y con cierta atención, ya que la calidad de la ontología final depende en gran medida de una elección correcta de sus conceptos. Pero antes de describir el abanico de métodos existentes, es necesario antes plantearse qué es un concepto.

Posiblemente por el hecho de haberse propuesto, a lo largo de la historia, una multitud de definiciones de la palabra “concepto” desde diferentes campos (filosofía, lingüística, etc.), la respuesta a la pregunta ¿qué es? no queda del todo clara. Sin embargo, y aunque la cuestión siga de hecho abierta, las disciplinas de carácter pragmático (como lo es la IA) que manejan

la idea de concepto adoptan una definición operativa del término, evitando discusiones de carácter filosófico.

Dentro del marco de las ontologías, se puede considerar un concepto como *aquello de lo que se puede decir algo*. En palabras de Corcho y Gómez Pérez [30], *“a concept can be anything about which something is said, and, therefore, could also be the description of a task, function, action, strategy, reasoning process, etc.”*. A pesar de la sencillez y generalidad de la definición, pueden observarse claras influencias lingüísticas que se evidencian en el apoyo sobre el lenguaje natural.

La conexión entre el lenguaje natural y la idea de concepto es estudiada por la semiótica, a través de la relación entre el signo y el concepto (véase el triángulo semiótico de Ogden [80]):

Figura 2. Triángulo semiótico de Ogden

Esta conexión es la que nos permite asegurar, de hecho, que es posible obtener conceptos a partir de textos ya que, según esto, vienen representados por las palabras que contienen.

En concreto, los métodos para obtener conceptos del dominio a partir de textos se apoyan en la noción de término, en lugar de palabra. Un término se define, precisamente, como una expresión o conjunto de palabras que determinan unívocamente un concepto del dominio. Según Cherfi y Toussaint [31], *“a term consists in one or more words considered together as a single syntactic construction (i.e. an indivisible unit). This term makes sense only in the context in which it is used (trade association, technical or scientific domains, etc.). This context is called domain of speciality. A term denotes an object (abstract or concrete) of the domain of speciality”*.

Se puede afirmar, con carácter general, que en cualquier método de obtención de conceptos a partir de un texto se distinguen dos etapas:

- Reconocimiento de los términos del texto.
- Generación o determinación de los conceptos a partir del conjunto de términos.

Hay que tener en cuenta, por otra parte, que no todos los términos de la ontología final estarán representados por términos encontrados explícitamente en el texto. Es decir, al conjunto de

conceptos que se obtienen a partir de las técnicas que se describirán en este capítulo, se suelen sumar otros, que serán descubiertos más tarde, generalmente durante la construcción de la jerarquía conceptual (ver *clustering*, sección 1.3.1).

1.2.1 Reconocimiento de términos

Como se puede deducir de lo comentado anteriormente, un término puede estar formado por una palabra o por varias. Así, tendríamos que “bacteria”, “Banco de España” o “televisión de plasma” podrían considerarse como términos.

Tal y como apuntan Kageura y Umino [59], se puede evaluar si una expresión es o no un término estudiando dos propiedades:

- *Termhood*, definido como el grado en que una determinada expresión representa un concepto del dominio.
- *Unithood*, definido como el grado de lo que se podría denominar *cohesión sintáctica* de la expresión a lo largo del texto.

Las técnicas de identificación de términos se caracterizan por analizar el texto de entrada utilizando un analizador sencillo para, posteriormente, obtener una lista de candidatos a término, ordenada siguiendo algún criterio de relevancia relacionado con alguna de las características anteriores. Un umbral prefijado o un experto se encargarán de descartar candidatos.

Estos métodos se dividen en dos tipos: métodos lingüísticos y estadísticos.

Generalmente, los métodos lingüísticos se utilizan para realizar una primera selección de expresiones candidatas, mientras que los estadísticos se encargan de ordenarlas. Lo normal es, por tanto, verlos funcionar conjuntamente [106]. A continuación se discutirán brevemente estas dos aproximaciones.

A) Métodos lingüísticos

Los métodos lingüísticos se basan en asumir que existen ciertas características comunes en la estructura gramatical de los términos. Con esta idea, se trata de encontrar patrones mediante los que determinar un conjunto de expresiones candidatas.

Este tipo de técnicas se han empleado con frecuencia en los dominios de la química, medicina y biología, disciplinas cuyo vocabulario se somete a unas reglas de formación rígidas (por

ejemplo, nombres de proteínas: *transaminasa* y *4-dendroprolina*), como queda demostrado en los trabajos de [42, 44, 58].

Sin embargo, salvo en dominios muy concretos como los anteriores, los términos no siguen estas reglas tan específicas. Por lo general, las palabras que componen los términos se presentan formando determinados patrones, modelados en función de las categorías gramaticales de las palabras. Por ejemplo, en [85] se identifican secuencias de palabras candidatas a términos comprobando si siguen patrones definidos mediante expresiones regulares como la siguiente:

$$((\text{DET})^* (\text{ADJ}|\text{NOUN}|\text{POS})^* (\text{NOUN}))$$

Que significa: secuencias de palabras que comienzan con 0 o más determinantes, 0 o más adjetivos, nombres o posesivos en cualquier orden, y finalizadas por un nombre.

En la identificación de los términos también suelen considerarse sus diferentes variantes formales. Por ejemplo, en [34] se utilizan técnicas para determinar que las secuencias “review of books” y “book review” se corresponden con el mismo término, seleccionando una de ellas como su forma canónica.

Otra técnica que suele emplearse es realizar una primera criba de palabras que se sabe de antemano que no formarán parte de ningún término. Estas palabras filtradas son las denominadas palabras vacías o *stopwords* – concepto acuñado por Luhn [68] –, caracterizadas por representar poca o nula información. Por ejemplo, Blake y Pratt [9] utilizaron una lista de 417 *stopwords* accesibles en [86]¹.

B) Métodos estadísticos

Los métodos estadísticos se han utilizado tanto para identificar los términos según la cohesión sintáctica de las palabras que lo forman (*unithood*) como para determinar la relevancia de un término con respecto al dominio del texto (*termhood*).

En el primer caso, se tratan de identificar palabras que tienden a aparecer juntas, formando una expresión – lo que se denomina *colocación* –, mientras en el segundo caso, se utiliza la

¹ El conjunto de *stopwords* listadas en el recurso web referenciado ha variado con el tiempo. En el momento de publicarse el trabajo de Blake y Pratt contenía 417 palabras.

frecuencia estadística de acuerdo a los siguientes criterios, derivados de la caracterización de los términos índices (*index terms*) que hacen Kageura y Umino [59]:

- Una expresión que aparezca frecuentemente en un dominio, es probable que sea un término del dominio.
- Una expresión que aparezca sólo en un dominio, es probable que sea un término de ese dominio.
- Una expresión que aparezca relativamente más veces en un texto de un dominio específico que en un texto general, es probable que sea un término de ese dominio.
- Una expresión que aparece relativamente más veces en un texto de un dominio con respecto a textos de otros dominios, es probable que sea un término de ese dominio.

A continuación se enumeran las técnicas estadísticas más empleadas en la identificación de términos.

B.1) Colocaciones

Según Manning y Schütze [71], una colocación (*collocation*) es “*an expression consisting of two or more words that correspond to some conventional way of saying things*”.

Ejemplos son “hormigón armado”, “vino blanco”, “banco central”, etc...

Estas secuencias de palabras poseen tres características fundamentales:

- Son expresiones *no composicionales*: es decir, el significado de la expresión completa no es simplemente la suma de los significados de cada palabra.
- Las palabras que forman la expresión no son sustituibles: no se pueden sustituir por otras palabras, aunque tengan el mismo significado dentro del contexto. Por ejemplo, no se puede decir “vino amarillo”, aunque amarillo se refiera un color como la palabra “blanco”. En “vino blanco”, no se refiere a un color sino a una clase de vino.
- La expresión no es modificable: no se pueden añadir más palabras, o realizar transformaciones gramaticales. No se puede decir, por ejemplo: “banco centralizado”.

La idea es que estas secuencias de palabras ocurren con una frecuencia lo suficientemente elevada como para hacer pensar que la expresión no se ha formado por pura casualidad, pudiéndose considerar como términos compuestos (formados por más de una palabra).

Un procedimiento para comprobar si una secuencia de palabras $(w_1, \dots, w_n)$ forma una expresión de este tipo es el siguiente:

- Se establece la hipótesis H , mediante la que se asume que las palabras no están asociadas, es decir la secuencia se produce de manera casual.
- Se calcula la distribución de probabilidad de aparición de la expresión si H fuera cierta. Asumiendo la independencia, sería:
$$P(w_1, \dots, w_n) = P(w_1) P(w_2) \dots P(w_n)$$
- Se obtiene la distribución de probabilidad real en una colección de textos.
- Si la distribución de probabilidad obtenida en la muestra es mucho mayor que la calculada para la hipótesis H , se rechaza la hipótesis y se asume que la expresión forma un término.

Los indicadores más utilizados para comparar dos distribuciones de probabilidad son *Test-t* y *Chi-cuadrado*.

El Test-t o prueba t de Student es utilizada por Manning y Schutze [71] para determinar de esta forma si una combinación de palabras forman o no una colocación. Esencialmente, el test mide lo probable que es que una muestra, caracterizada mediante una media y una varianza, se haya obtenido de una distribución normal de media μ . La fórmula es la siguiente:

$$Test - t = \frac{\bar{x} - \mu}{\sqrt{\frac{s^2}{N}}}$$

Donde:

$\bar{x}$ es la media de la muestra

s^2 es la varianza de la muestra

N es el tamaño de la muestra

μ es la media de la distribución.

El Test-t tiene el inconveniente de que asume una distribución normal de probabilidades, y esto generalmente no tiene por qué ser cierto. Una alternativa es la medida χ^2 , denominada Chi-cuadrado, que básicamente compara las frecuencias observadas en la muestra con las frecuencias esperadas, asumiendo la independencia (hipótesis H).

La fórmula es la siguiente:

$$\chi^2 = \sum \frac{(obs - exp)^2}{exp}$$

Donde:

obs es la frecuencia observada en la muestra

exp es frecuencia esperada

Otro método que se suele emplear para descubrir asociaciones de palabras, es el de la información mutua, propuesto por Church y Hanks [33]. Esta medida está basada en la teoría de la información. La fórmula es:

$$I(w_i, w_k) = \log_2 \frac{P(w_i, w_k)}{P(w_i)P(w_k)}$$

Donde:

w_i y w_k son palabras

$P(w_i)$ y $P(w_k)$ son las probabilidades de aparición² de las palabras w_i y w_k

$P(w_i, w_k)$ es la probabilidad de que w_i y w_k aparezcan juntas

Según la expresión anterior, la información mutua entre dos palabras será mayor si la ocurrencia de una palabra predice la ocurrencia de la otra. Cuanta más información exista entre dos palabras, más probable es que estemos ante un término compuesto.

B.2) TFIDF (Term Frequency - Inverse Document Frequency)

Es un sistema de pesos muy utilizado en el área de la recuperación de información para seleccionar los términos más relevantes de entre un conjunto de expresiones candidatas. Consiste en aplicar la siguiente fórmula simple:

² La probabilidad de una palabra (o secuencia de palabras) se estima dividiendo su frecuencia absoluta por el número de palabras del texto que se utiliza como muestra.

$$w_{i,j} = tf_{i,j} \cdot \log \left(\frac{N}{df_i} \right)$$

Donde:

$w_{i,j}$ es una palabra determinada i en un documento j

$tf_{i,j}$ es el número de ocasiones en las que aparece el término i en el documento j

df_i es el número de documentos que contienen el término i

N es el número total de documentos

Se observa que el peso o puntuación de un término, aplicando la fórmula anterior, es proporcional al número de veces que aparece en el documento y al número de documentos que contienen el término.

Como se ha comentado, el parámetro TFIDF se utiliza habitualmente en el campo de la recuperación de información [91], donde el objetivo es buscar documentos que cumplan unos criterios de búsqueda basados en términos introducidos por un usuario. Por lo general, se consideran términos relevantes de un documento aquellos que aparecen con una mayor frecuencia. Sin embargo, sería un error tomar la frecuencia absoluta como único criterio, puesto que en ese caso un documento que solo contuviera un término repetido muchas veces sería considerado como el documento buscado. Por esta razón, suelen emplearse otros criterios para puntuar la relevancia, siendo TFIDF uno de los más utilizados.

B.3) Valores C/NC

Este método, propuesto por Frantzi y Ananiadou [41] presenta un sistema de puntuación de relevancia que combina dos medidas: el *valor C* y el *valor NC*.

El valor C mide la frecuencia de aparición del término, teniendo en cuenta además si el término forma parte de otros más grandes. Es decir, se considera, por ejemplo, el caso en el que tanto “lente de contacto” como “lente de contacto blanda” sean ambos términos candidatos.

Para ello, utiliza los siguientes datos:

1. La frecuencia total de aparición de la cadena candidata
2. La frecuencia de la cadena candidata como parte de otros términos más largos

3. El número de esos términos candidatos más grandes
4. La longitud, en palabras, de la palabra candidata

La fórmula que calcula el valor C para una cadena (secuencia de palabras) candidata es la siguiente:

$$C - value(a) = \begin{cases} \log_2 |a| \cdot f(a) & \text{si } a \text{ no está anidado} \\ \log_2 |a| \left(f(a) - \frac{1}{P(T_a)} \sum_{b \in T_a} f(b) \right) & \text{en otro caso} \end{cases}$$

Donde:

a es la cadena candidata

$f(x)$ es la frecuencia absoluta de la cadena x

T_a es el conjunto de términos candidatos que contienen a a .

$P(T_a)$ es el número de esos términos candidatos

b es un término que contiene a a

Este indicador se complementa además con el valor NC , que tiene en cuenta la información que proporciona el contexto de los términos. Generalmente, un término suele estar rodeado de un conjunto restringido de palabras. Por ejemplo, “consistente” es una palabra que suele aparecer precediendo a un término del dominio de la medicina. Lo mismo ocurre, por ejemplo, con el verbo “demostrar”, en frases como “... demuestra un carcinoma basal celular”.

El sistema de puntuación propuesto utiliza esta información de contexto. Partiendo de los términos con un valor C más alto, se obtiene una lista de palabras (nombres, adjetivos y verbos) situados en su entorno contextual.

Después, se selecciona un subconjunto de esas palabras utilizando un peso definido de la siguiente forma:

$$weight(w) = \frac{t(w)}{n}$$

Donde:

w es la palabra de contexto (nombre, verbo o adjetivo) a la que se asignará un peso

$t(w)$ es el número de términos en cuyo contexto aparece la palabra w

n es el número total de términos considerados

$weight(w)$ es el peso asignado a w

Una vez obtenido el conjunto de palabras del entorno, se calcula el valor NC para una cadena candidata mediante la siguiente fórmula:

$$NC - value(a) = 0.8 \cdot C - value(a) + 0.2 \sum_{b \in C_a} f_a(b) weight(b)$$

Donde:

a es la cadena candidata

C_a es el conjunto de las palabras de contexto situadas en el entorno de a

b es una palabra de C_a

$f_a(b)$ es la frecuencia de b como palabra de contexto de a

$weight(b)$ es el peso de b como palabra de contexto de un término (fórmula anterior)

B.4) Domain Relevance – Domain consensus

Este método fue propuesto por Navigli y Velardi [76] específicamente para identificar términos relevantes para la construcción de una ontología del dominio. Se basa también en dos medidas: *DR* (*Domain Relevance*) y *DC* (*Domain Consensus*) y tiene en cuenta la distribución de los términos en diferentes dominios.

Se fundamenta en la idea de que los términos caracterizan a un dominio, por lo que deberán aparecer con mucha más frecuencia en un texto de ese dominio que en cualquier otro. Una expresión que no es un término aparecerá con una frecuencia similar en textos de diferentes dominios.

La medida $DR_{t,k}$ da una idea de la relevancia de un término t en un dominio determinado D_k . La fórmula que calcula el valor es la siguiente:

$$DR_{t,k} = \frac{P(t|D_k)}{\max_{1 \leq j \leq n} P(t|D_j)}$$

Donde:

$\{D_1, \dots, D_n\}$ es un conjunto de n dominios diferentes

$P(t|D_j)$ son las probabilidades condicionadas de aparición del término t en un texto del dominio D_j

Las probabilidades condicionadas $P(t|D_j)$ se estiman de la siguiente manera:

$$E(P(t|D_j)) = \frac{freq(t \text{ in } D_j)}{\sum_{i=1 \dots n} freq(t \text{ in } D_i)}$$

Donde $freq(t \text{ in } D_j)$ es la frecuencia del término t en el dominio D_j .

La segunda medida tiene en cuenta cómo se distribuye el término entre los documentos pertenecientes a un determinado dominio. Existen términos con muy alta frecuencia de aparición que sólo ocurren en un documento, mientras que en el resto de documentos pertenecientes al mismo dominio aparecen con poca o ninguna frecuencia.

La distribución de un término t en los documentos d_j de un dominio puede ser considerada como una variable aleatoria estimada a través de todos los d_j pertenecientes al dominio. El grado de consenso de t en el dominio D_k (valor $DC_{t,k}$) se puede obtener calculando la cantidad de información de esta distribución, aplicando la fórmula de la entropía:

$$DC_{t,k} = \sum_{d_j \in D_k} \left(P(t|d_j) \log \frac{1}{P(t|d_j)} \right)$$

Donde $P(t|d_j)$ es la probabilidad de aparición de un término t en un documento d , estimada de la siguiente forma:

$$E(P(t|d_j)) = \frac{freq(t \text{ in } d_j)}{\sum_{d_j \in D_k} freq(t \text{ in } d_j)}$$

Donde $freq(t \text{ in } d_j)$ es la frecuencia del término t en el documento d_j .

Finalmente, para calcular la puntuación final del término t en con respecto al dominio D_k ($TW_{t,k}$) se combinan ambas medidas utilizando la siguiente formula:

$$TW_{t,k} = \alpha DR_{t,k} + \beta DC_{t,k}^{norm}$$

Donde $DC_{t,k}^{norm}$ es una entropía normalizada, mientras que α y β son valores pertenecientes al intervalo (0,1) que se determinan experimentalmente. Típicamente, se suele asignar valores cercanos a 0.9 a α , mientras que el valore β depende del número N de documentos del dominio D_k . Para valores altos de N , el valor β adecuado está entre 0.35 y 0.25.

1.2.2 Determinación de conceptos

El paso del término al concepto que representa no es trivial. Un mismo concepto puede materializarse de formas diferentes a través de términos sinónimos, o bien mediante términos equivalentes en otros idiomas. La determinación de los conceptos pasa necesariamente por encontrar las diferentes variantes con las que se pueden presentar en los textos.

Parte de los trabajos dirigidos a este objetivo tratan de identificar el significado de los términos utilizando repositorios lingüísticos como *Wordnet* [38] y *EuroWordnet* [103], empleando técnicas clásicas de desambiguación de significados (*word sense disambiguation*) [76, 105, 109].

Otras aproximaciones emplean el agrupamiento o *clustering*, basándose en la hipótesis distribucional de Harris [53], de acuerdo a la cual, los términos similares tienden a compartir el mismo contexto sintáctico. Se encuentran trabajos en esta línea en [56, 64, 65], aparte de otros provenientes de áreas relacionadas como la inducción del significado de las palabras (*word sense induction*) [12, 13], y la recuperación de información [61, 89].

1.2.3 Concepto vs Instancia

Además de la idea de concepto, en los modelos conceptuales en general, y en las ontologías en particular, se maneja también la idea de *instancia*. Informalmente, una instancia es una ocurrencia material concreta de un concepto.

No es trivial, a menudo, determinar si algo es una instancia o un concepto dentro de una ontología.

Se puede ver con un ejemplo sencillo:

Figura 3. Instancia - concepto

En el ejemplo anterior es claro concluir que “Nôtre Dame” es una instancia del concepto “Catedral”, ya que los nombres propios, (como el caso de “Nôtre Dame”) son formas de identificar de manera unívoca a entidades concretas entre individuos de su misma clase³. A este tipo de instancias se las denomina *named entities*.

Sin embargo, en la jerarquía de la Figura 4, se podría llegar a dos conclusiones diferentes:

- “Lápiz” es una instancia de “Utensilio de escritura”.
- “Lápiz” es un concepto subclase de “Utensilio de escritura”. Se debería esperar como instancias algo más concreto, como “el lápiz número 27”.

³ En lo que resta, se considerará la palabra “clase” como sinónimo de “concepto”

A medida que descendemos en la jerarquía de clases, los conceptos son cada vez más específicos y llega un punto en el que podemos encontrar este tipo de dificultad.

Figura 4. Determinación de concepto e instancia

Generalmente, los métodos de construcción automática de ontologías resuelven la decisión anterior dependiendo de su propósito:

- Los métodos de población de ontologías (*ontology population* [81]) parten de una ontología ya definida, por lo que considerarán que se trata de una instancia.
- Los métodos de *ontology learning* construyen la definición misma de la ontología, por lo que considerarán que se trata de un concepto.

Por este motivo, las técnicas empleadas para la obtención de conceptos suelen utilizarse también para la obtención de instancias.

1.3 Métodos para la obtención automática de relaciones taxonómicas

Las relaciones taxonómicas, también denominadas “es un”, “hiperonimia” o “subclase de” son relaciones fundamentales en la IA, ya que permiten *clasificar* entidades. Decidir que un determinado objeto pertenece a una clase o a otra es una actividad que se relaciona claramente con un comportamiento inteligente.

Este tipo de relación antisimétrica, no reflexiva y transitiva, vertebró el conjunto de conceptos de una ontología, formando una jerarquía conceptual. No es de extrañar, por tanto, que la construcción de estas jerarquías haya sido el objetivo de muchas disciplinas dentro de la IA, como el procesamiento del lenguaje natural, la recuperación de información, clasificación, etc.

A grandes rasgos, existen tres clases de métodos para descubrir este tipo de relaciones a partir de textos:

- Métodos basados en la semejanza semántica
- Patrones léxico-sintácticos
- Composición de términos

A continuación se describirán estos tipos de aproximaciones.

1.3.1 Métodos basados en la semejanza semántica

Cuando en una jerarquía clasificatoria observamos que un determinado concepto se encuentra justo debajo de otro, decimos que el primero es *más específico* que el segundo. Dicho de otro modo, el concepto específico posee una definición más precisa, *añadiendo* sus propiedades particulares a la definición del concepto general. Esto implica que, aunque los conceptos son diferentes, ambos comparten ciertas propiedades, permitiendo afirmar que *se parecen*.

Por ejemplo, en la jerarquía conceptual de la Figura 5 se puede deducir intuitivamente que el concepto “gorrión” se parece más al concepto “ave” que al concepto “perro”, ya que comparte con aquél las propiedades {“está vivo”, “tiene alas”, “tiene pico”}, mientras que con “perro” sólo comparte la propiedad “está vivo”.

Con este pequeño caso se acaba de definir informalmente una *semejanza conceptual*, basada en la comparación de las propiedades que caracterizan intensionalmente los conceptos. Pero lo verdaderamente importante es que esta semejanza guarda relación con la posición de los conceptos en la clasificación: cuanto más parecidos sean los conceptos, más cerca se encontrarán en la jerarquía.

Esta última idea es la base sobre la que se fundamenta un conjunto de métodos para la construcción de jerarquías conceptuales a partir de textos. En todos ellos se define formalmente una medida de semejanza, que permita adivinar de alguna forma cuál debe ser la estructura de la jerarquía que clasifica un conjunto de conceptos.

Figura 5. Jerarquía de conceptos

Definición de semejanza semántica

Características como “está vivo”, “tiene pico”, etc. son atributos de conceptos que no se encuentran especificados directamente en el texto. A menos que se realice un análisis muy profundo y se disponga de una gran ontología previa, no se podrá contar con tales descriptores para definir una semejanza conceptual. En lugar de ello, y dado que los conceptos están representados en el texto mediante palabras, se suele emplear la denominada *semejanza semántica*, definida a partir de los significados de las palabras, que representan, a su vez, conceptos. Se dice que dos palabras son semánticamente similares si pueden sustituirse una por otra en el texto sin que éste pierda coherencia.

Lo dicho anteriormente sobre la cercanía en la clasificación de dos conceptos similares, sigue siendo válido utilizando la semejanza semántica. De hecho, a menudo suele definirse esta semejanza a partir del número de enlaces que separan a los términos en una jerarquía léxica tipo Wordnet, como en [3, 48, 57, 62].

Sin embargo, este tipo de semejanza sólo se puede determinar si ambos términos están presentes en la jerarquía, por lo que los métodos para la construcción de taxonomías a partir de textos utilizan otro tipo de semejanza semántica: la denominada *semejanza semántica distribucional*.

Como se ha comentado anteriormente, la *hipótesis distribucional* establecida por Harris [53], afirma que términos con significados similares tienden a aparecer en contextos lingüísticos similares. Utilizando esta idea, se puede caracterizar el significado de un término mediante la distribución de determinadas palabras clave que aparecen frecuentemente junto a él (*coocurren*, empleando terminología estadística) en un entorno contextual determinado.

Por ejemplo, Lin y Hovy [63] asocian a cada término lo que denominan *topic signature*, que es una estructura formada por el par {término, $\langle (t_1, w_1), (t_2, w_2), \dots, (t_n, w_n) \rangle$ } donde se asocia a cada término un vector de términos t_i que aparecen en el texto con mucha frecuencia cerca de él. Los valores w_i son pesos, que son calculados utilizando indicadores estadísticos.

También es frecuente observar cómo se establecen restricciones sobre las palabras clave más frecuentes, como por ejemplo su función sintáctica. En varios trabajos (por ejemplo [27, 36, 84]) se consideran como palabras clave los verbos cuyo objeto directo sea el término cuyo significado se quiere caracterizar.

Una vez caracterizado el significado de esta manera, se pueden definir métricas que cuantifiquen la semejanza semántica entre dos términos, en función de las distribuciones correspondientes.

Los métodos para construir jerarquías de conceptos a partir de textos que se basan en la semejanza semántica se pueden clasificar en dos tipos:

- Agrupación de términos o *clustering*.
- Clasificación de conceptos.

A) *Clustering*

Los métodos de *clustering* utilizan la distancia semántica para definir una operación de agrupación entre conceptos similares. La idea es que el concepto resultante de una agrupación de conceptos similares (denominado *clúster*) es *más general* que cada uno de éstos, pudiéndose afirmar que es su superclase.

Según Shamsfard y Barforoush [90] los métodos de *clustering*, en general, se caracterizan según los cuatro factores que se muestran en la Tabla 1.

Un ejemplo de *clustering* bottom-up es el *clustering* jerárquico acumulativo, propuesto por Duda et al. [35], mientras que se puede ver un *clustering* top-down en el algoritmo de las K-Medias con bisección en [93], utilizado también por Fortuna et al. [40].

Tabla 1. Clasificación de métodos de clústering

Modo	Online/Offline	<p>El clústering <i>online</i> o incremental presupone que los elementos de entrada (términos) se tratan secuencialmente. El algoritmo procesa cada elemento y reorganiza los clústers apropiadamente.</p> <p>En el clústering <i>offline</i> se parte del conjunto de elementos de entrada completo, procesándose todos a la vez durante la ejecución del algoritmo.</p>
	Jerárquico/no-jerárquico	<p>El clústering es <i>jerárquico</i> si la relación entre los clústers es de tipo “es un”, mientras que es <i>no-jerárquico</i> en otro caso. En esta sección se hace referencia únicamente al clústering jerárquico.</p>
	Clústering simple/Multi-clústering	<p>En un algoritmo de <i>multi-clústering</i>, un mismo elemento puede acabar perteneciendo simultáneamente a dos clústers distintos, mientras que el clústering simple sólo puede pertenecer a uno.</p>
Dirección	Top-down	<p>Se parte inicialmente de un clúster que contiene a todos los elementos. Este clúster se descompone en otros, manteniendo el criterio de máxima semejanza entre los elementos pertenecientes a un clúster. El proceso se repite hasta que cada clúster esté formado únicamente por un elemento.</p>
	Bottom-up	<p>Inicialmente cada elemento constituye un clúster. En cada iteración del algoritmo se agrupan los clústers más similares hasta llegar a la situación en la que un único clúster contiene todos los elementos.</p>
Medida de semejanza	Existen medidas de semejanza de diferente naturaleza. En la sección anterior se han mostrado ejemplos de cómo se define la semejanza semántica.	
Cálculo de semejanza entre dos clústers	Enlace simple	La distancia entre dos clústeres se corresponde con la distancia de sus elementos más cercanos.
	Enlace completo	La distancia entre dos clústeres se corresponde con la distancia de sus elementos más lejanos.
	Media del grupo	La distancia entre dos clústers se corresponde con la distancia media entre sus elementos.

La Figura 6 ilustra los pasos que sigue un algoritmo de clústering acumulativo para construir una jerarquía:

Figura 6. Secuencia de pasos en algoritmo de clústering

En cada paso se agrupan los clústeres más cercanos según la medida de distancia que se utilice, resultando en un nuevo clúster que pasa a ser el concepto padre de ambos. Una vez que han sido agrupados, los dos clústers son descartados en las agrupaciones que se realicen en iteraciones sucesivas. De esta forma, en el paso 2 de la figura, los clústers C1 y C2 son los más cercanos, formando el clúster C12 con el que se vinculan mediante la relación jerárquica. Ambos clústeres (C1 y C2) ya no intervendrán en la iteración posterior, de forma que en el paso 3 solo se calcularán las distancias entre C12, C3, C4 y C5, encontrándose con que los más semejantes son el C3 y C4. El proceso sigue hasta que solo queda un clúster, que constituirá la raíz de la jerarquía.

Set-Theoretic Clustering

Existe un tipo concreto de clústering en el que la semejanza semántica sirve de base para definir una *relación de orden* entre términos, en lugar de la habitual medida de distancia. Esta relación de orden junto al conjunto de términos forman un retículo, del que se puede deducir una taxonomía.

Este método, propuesto por Cimiano et al. [26], se basa en el denominado *Formal Concept Analysis* (FCA) [46], utilizado para el análisis de datos y la gestión de información.

La teoría se basa en la idea de *contexto formal*, definido como una terna (G, M, I) en la que G es un conjunto de objetos, M es un conjunto de atributos e I es la función de incidencia, que especifica una relación binaria entre objetos y atributos ($I \subseteq G \times M$). Intuitivamente, la función I asigna a los objetos de M un conjunto de características, definidas como atributos.

Los datos de entrada se contemplan como un conjunto de *conceptos formales*. Un concepto formal es un par (A, B) en el que A es un conjunto de objetos ($A \subseteq G$) y B es un conjunto de atributos ($B \subseteq M$) y además se cumplen las siguientes condiciones:

- El conjunto de todos los atributos compartidos por los objetos de A es igual a B .
- El conjunto de todos los conceptos que comparten los atributos de B es igual a A .

Los objetos de A son referidos como la *extensión* del concepto, mientras que los atributos son considerados como su *intensión*.

Una vez definidos de esta manera, se establece una relación de orden entre los conceptos (subconcepto - superconcepto) por la que un concepto (A_1, B_1) será menor ($\leq$) que otro (A_2, B_2) si $A_1 \subseteq A_2$ o $B_1 \subseteq B_2$.

Cimiano et al. [26] utilizan este marco formal para construir una taxonomía. El proceso comienza realizando un análisis morfológico y sintáctico del texto, extrayendo las estructuras verbo/sujeto, verbo/objeto y verbo/sintagma preposicional. Tras lematizar los verbos y los sustantivos obtenidos, se forma una lista de conceptos, definidos cada uno mediante un sustantivo y un vector con los verbos que aparecen junto a él en los contextos sintácticos.

Tomando el mismo ejemplo utilizado por el autor, correspondiente a un texto del dominio del turismo, se obtienen los sustantivos “apartment”, “car”, “motor-bike”, “excursión” y “trip”, y los verbos “to book”, “to rent”, “to drive”, “to ride” y “to join”. Esta información se traslada al marco del FCA, considerando los sustantivos como objetos y los verbos como atributos. La matriz objeto - atributo sería la siguiente:

Tabla 2. Matriz objeto - atributo

	bookable	rentable	driveable	rideable	joinable
apartment	x	x			
car	x	x	x		
motor-bike	x	x	x	x	
excursion	x				x
trip	x				x

(Para cada término, se marca con una cruz los verbos que aparecen en su contexto sintáctico).

A partir de la matriz, se definen los siguientes conceptos formales:

1. $(A_1 = \{\text{apartment, car, motor-bike, excursion, trip}\}, B_1 = \{\text{bookable}\})$
2. $(A_2 = \{\text{apartment, car, motor-bike}\}, B_2 = \{\text{bookable, rentable}\})$
3. $(A_3 = \{\text{car, motor-bike}\}, B_3 = \{\text{bookable, rentable, driveable}\})$
4. $(A_4 = \{\text{excursion, trip}\}, B_4 = \{\text{bookable, joinable}\})$
5. $(A_5 = \{\text{motor-bike}\}, B_5 = \{\text{bookable, rentable, driveable, rideable}\})$

Utilizando la relación de orden, los conceptos se estructuran formando el retículo de la siguiente figura:

Figura 7. Retículo de conceptos

Basándose en la representación anterior, se puede construir la jerarquía con facilidad, siguiendo los pasos siguientes:

- Por cada nodo del retículo, se crea un concepto etiquetado con uno de sus atributos. Si posee varios, se elige el que no esté presente en los conceptos mayores que él según la relación de orden.
- Los enlaces del retículo se trasladan a relaciones “es un” entre los conceptos. Es decir, por cada enlace entre dos nodos, se crea una relación taxonómica entre los conceptos correspondientes, de forma que el concepto menor es subclase del mayor.
- Por cada objeto se crea un concepto, ubicándolo en la jerarquía bajo el concepto que lo contiene como extensión. Si existen varios conceptos que contienen ese objeto

como extensión, se ubicará bajo el concepto menor según la relación de orden. Por ejemplo, el objeto “car” aparece como extensión de los conceptos formales 1, 2 y 3 definidos anteriormente. Se elegirá el concepto formal 3 al ser el menor de ellos, por lo que “car” será subclase del concepto de la jerarquía correspondiente al concepto formal 3.

Siguiendo estos pasos sobre el retículo de la Figura 7, se obtiene la siguiente taxonomía:

Figura 8. Jerarquía de conceptos resultante

B) Clasificación de conceptos

Otra aproximación para construir taxonomías mediante semejanza semántica la constituyen aquellos métodos cuyo principal cometido es incorporar nuevos conceptos a una ontología previa. Estos métodos surgieron para tratar de cubrir la carencia de recursos léxicos del tipo de Wordnet, aunque posteriormente se han incorporado al conjunto de herramientas de *Ontology Learning*.

En líneas generales, un procedimiento típico de clasificación toma como entrada un nuevo concepto y recorre los nodos de la taxonomía en un orden determinado, hasta encontrar aquél que se pueda considerar como superclase del nuevo concepto en función de una semejanza semántica previamente definida. Una vez encontrado el concepto superclase, se inserta el nuevo concepto en la taxonomía.

Ejemplos de estos métodos los encontramos en [2, 4, 104, 107].

1.3.2 Patrones léxico – sintácticos

A partir de textos de una temática totalmente desconocida, es posible poder llegar a deducir intuitivamente ciertas relaciones entre conceptos. Por ejemplo, dado el siguiente fragmento:

*“Hay dos clases principales de antidepresivos: tricíclicos, tales como la amitriptilina (Elavil) y la doxepina (Sinequan), e inhibidores selectivos de la recaptación de serotonina, o ISRS, **tales como** la fluoxetina (Prozac), la paroxetina (Paxil) y la sertralina (Zoloft).”*

Cualquier lector, sin saber a priori qué es la “amitriptilina” o la “doxepina”, podrá deducir que son un tipo de “antidepresivo tricíclico”. De la misma forma, también se puede deducir que la “fluoxetina”, la “paroxetina” y la “sertralina” son tipos de “ISRS”.

Es fácil ver que para ello ha bastado tan sólo identificar la siguiente estructura o patrón:

<término 1>”, **tales como** <término 2>”, <término 3>...<término n>

como un indicador de que los conceptos referidos por <término i>, i=2..n están relacionados de alguna manera con el <término 1>.

El ejemplo anterior sugiere la existencia de determinadas estructuras, que se suelen utilizar de manera recurrente en los textos, que son independientes del dominio, y que permiten inferir ciertas relaciones, requiriendo únicamente un análisis superficial de las oraciones.

Esta idea fue utilizada por Marti Hearst en 1992 [54] para explotar la información contenida en los textos. En concreto, sus investigaciones se centraron en la obtención de relaciones de hiponimia entre términos, con el fin de automatizar la construcción de lexicones de gran tamaño.

Para ello, trató de identificar patrones que pudieran indicar dicha relación, estableciendo los siguientes requisitos:

- Deben ocurrir con bastante frecuencia, y en textos de diferentes temas.
- Casi siempre indican la relación en cuestión (hiponimia).
- Se pueden reconocer con poco o ningún conocimiento previo.

En total, obtuvo los seis patrones siguientes (referidos al idioma inglés), a los que denominó *patrones léxico-sintácticos* (ver Tabla 3).

Realizó las experimentaciones empleando un analizador sintáctico sencillo aplicado a un gran corpus, procedente de una enciclopedia. Demostró que los dos primeros patrones de la tabla son los más eficaces a la hora de extraer relaciones, con una precisión del 55%. También observó que se presentan de forma muy dispersa en el texto: se requirió un corpus de más de 8 millones de palabras, para inferir tan sólo 152 relaciones.

Tabla 3. Patrones léxico - sintácticos

Patrón (NP = noun phrase)	Ejemplo	Relaciones inferidas (hip(A,B) → "A es un tipo de B")
NP such as {NP,* {(and or)} NP	"The bow lute, such as the Bambra ndang, is plucked and has an individual curved neck for each string."	hip("Bambra ndang" , "bow lute")
such NP as {NP,* {(and or)} NP	"... works by such authors as Herrick, Goldsmith, and Shakespeare."	hip("author" , "Herrick") hip("author" , "Goldsmith") hip("author" , "Shakespeare")
NP {, NP}* {,} or other NP	"Bruises, wounds, broken bones or other injuries..."	hip("bruise" , "injury") hip("wound" , "injury") hip("broken bone" , "injury")
NP {, NP}* {,} and other NP	"... temples, treasures, and other important civic buildings."	hip("temple" , "civic building") hip("treasure" , "civic building")
NP {,} including {NP,* {or and} NP	"All common-law countries, including Canada and England..."	hip("Canada" , "common-law country") hip("England" , "common-law country")
NP {,} especially {NP,* {or and} NP	"...most European countries, especially France, England and Spain."	hip("France" , "European country") hip("England" , "European country") hip("Spain" , "European country")

A) Mejora de resultados

Una forma posible forma de mejorar la precisión al inferir relaciones de esta forma consiste en restringir el tipo de lenguaje sobre el que aplicar los patrones. Así, Sunblad [96] buscó patrones que permitieran descubrir relaciones de hiponimia y meronimia en expresiones interrogativas.

Para la relación de hiponimia propuso los patrones de la Tabla 4:

Tabla 4. Patrones léxico - sintácticos de hiponimia

Patrón	Relación inferida
Who is/was X	Hip(X,person)
What is/was X best/most known/famous for?	Hip(X,person)
Where is X (located)?	Hip(X,location)
What is the location of X?	Hip(X,location)
What does X stand for?	Hip(X,acronym)
X is an abbreviation/acronym for what?	Hip(X,acronym), Hip(X,abbreviation)
What kind/type of Y is/was X?	Hip(X,Y)

Mientras que para relación de meronimia (“es parte de”) propuso los patrones de la Tabla 5:

Tabla 5. Patrones léxico - sintácticos de meronimia

What is/was the X of Y?
What is the X for Y?
What is X's Y?
How many X are in/on Y?

El lenguaje interrogativo es mucho más conciso y directo, demostrando ser muy útil a la hora de inferir personas y lugares (precisión de más del 83%).

Aunque la precisión de los patrones en general es mejorable, el verdadero problema lo constituye su cobertura (*recall*). Es decir, el empleo de patrones léxico-sintácticos para la extracción de hipónimos descubre solamente una pequeña parte de todos los que son deducibles en el texto.

Cederberg y Windows [25] fueron un paso más allá en el desarrollo de esta técnica, tratando de mejorar ambos factores (precisión y cobertura).

Para mejorar la precisión utilizaron un sistema que permitía puntuar cada relación de hiponimia deducida. Esta puntuación se expresa en función de una distancia semántica entre términos, apoyándose en la hipótesis de que debería existir cierta semejanza semántica entre un hipónimo y sus hiperónimos (conceptos subclase).

Para calcular la distancia emplearon el denominado *Latent Semantic Analysis* (LSA), que se basa en representar cada palabra mediante un vector, utilizando el *Vector Space Model*. Los vectores se construyen de manera siguiente:

- Se eligen las 1000 palabras más frecuentes y significativas (que caractericen el dominio del texto), y se forma una lista con ellas: $w_1, w_2, \dots, w_{1000}$.
- Para cada palabra w del texto, se calcula el número total de veces que aparece cada palabra w_i en un entorno cercano a w (15 palabras).
- Si el número de ocurrencias de w_i (en el entorno de w) lo denominamos f_i , el vector asociado a w será $(f_1, f_2, f_3, \dots, f_{1000})$.

Para simplificar los cálculos, redujeron el número de dimensiones de 1000 a 100 utilizando técnicas algebraicas.

Una vez trasladada cada palabra a un espacio vectorial, definieron la medida de distancia semántica entre dos palabras a partir del coseno del ángulo que forman sus respectivos vectores.

Aplicando esta métrica, ordenaron el conjunto de los pares de términos obtenidos mediante los patrones léxico-sintácticos, y establecieron un valor umbral a partir del cual descartar el par candidato. Esta técnica demostró un aumento de precisión de hasta el 58%.

Por otro lado, abordaron el problema de la cobertura de los patrones, apoyándose en la información que proporciona la coordinación lingüística entre términos. Por ejemplo, en el siguiente fragmento:

“No ocurre lo mismo con el azúcar, la miel, el mosto, el clavo y otras especias que incrementan su valor.”

Utilizando patrones léxico-sintácticos, se puede deducir que el clavo es un tipo de especia. Si posteriormente encontramos la siguiente frase:

“Barcos embarcados con nuez moscada o canela, clavo o cilantro lucharon una vez para retornar a casa con su preciada mercancía”

Podemos llegar a la conclusión de que la nuez moscada, la canela y el cilantro son también especias, aunque no se haya empleado para ello ninguno de los patrones que se han visto hasta ahora.

La idea es que, generalmente, en los textos podemos encontrar enumeraciones de términos semánticamente similares y situados en el mismo nivel ontológico. Es decir, nos resultará más fácil encontrar la expresión “frutas y verduras” o “zanahorias y patatas” que “fruta y patatas” o “manzanas y verduras”.

De hecho, en un texto normalmente aparece una mayor cantidad de estos patrones de coordinación que de patrones de hiponimia, por lo que utilizaron esta nueva fuente de información en las técnicas de obtención de relaciones.

Una vez obtenida una determinada relación de hiponimia “y es un x” mediante los patrones léxico-sintácticos habituales, el proceso continua obteniendo 10 palabras similares a y utilizando estos nuevos patrones de coordinación. Se comprobará entonces si las nuevas palabras también son hipónimos de x, utilizando la distancia semántica definida anteriormente. La aplicación de esta técnica tuvo como resultado un aumento en 5 veces de la cobertura.

B) Aprendizaje de patrones

En los experimentos comentados hasta ahora se han empleado una cantidad reducida de patrones descubiertos manualmente, siguiendo el procedimiento propuesto por Hearst [54]. En general, este procedimiento resulta bastante tedioso, y se ha investigado la manera de automatizarlo.

Por ejemplo, Snow et al. [92] describen un algoritmo de aprendizaje de patrones para obtención de relaciones de hiponimia. Para ello, primero formaliza un espacio de

representación que permita expresar cualquier patrón léxico-sintáctico, haciendo uso de los denominados *árboles de dependencia*.

Un árbol de dependencia se genera como salida de un analizador sintáctico, y representa las relaciones sintácticas entre las palabras dentro de una oración. Utilizando esta idea, Snow definió el espacio de patrones léxico-sintácticos como *“todos los caminos mínimos, de cuatro enlaces o menos, entre dos nombres situados un árbol de dependencia sintáctica”*.

Una vez definido el formalismo, presenta un algoritmo de aprendizaje de patrones que sigue los siguientes pasos:

1. Parsear un corpus de gran tamaño, utilizando un analizador sintáctico ligero.
2. Obtener todos los pares de nombres que ocurren dentro de cada oración.
3. Etiquetar cada par como hipónimo o no-hipónimo, según exista o no una relación de hiponimia entre ellos, según Wordnet.
4. Obtener, de forma automática, utilizando el corpus, caminos sintácticos que enlacen 5 o más de los pares de términos etiquetados como hipónimos encontrados en el paso 2.
5. Cada camino encontrado se corresponde con un patrón léxico-sintáctico.

En las experimentaciones, se hallaron casi 70000 patrones, entre los que estaban los seis patrones descubiertos por Hearst (ver Figura 9). Snow intentó aprovechar de alguna forma este enorme conjunto de patrones. Trató de construir un clasificador que permitiera decidir si entre un par de términos existe o no una relación de hiponimia, utilizando la información proporcionada por todos los patrones.

Para ello definió, para cada par de términos, un vector v con n componentes, siendo n el número total de patrones encontrados (≈ 70000). El valor de cada componente v_i será el número de veces que el patrón correspondiente se instancia con ese par de términos en todo el corpus.

Una vez representado el par de términos mediante un vector, entrenó un clasificador que permitiera decidir si un vector representa o no una relación de hiponimia entre palabras, atendiendo a los valores de sus componentes. Para entrenarlo, utilizó como información de partida un conjunto de pares de palabras previamente clasificados como hipónimos o no hipónimos según el criterio de Wordnet.

Los mejores resultados de los experimentos demostraron una mejora de este clasificador de un 132% sobre la puntuación *F-score* con respecto a un clasificador basado solo en los patrones de Hearst.

Figura 9. Precisión vs cobertura de patrones léxico-sintácticos

1.3.3 Composición de términos

Un método sencillo para obtener relaciones taxonómicas entre términos consiste en analizar su composición léxica. Generalmente, los términos compuestos son más específicos que los términos simples. Por ejemplo, el término "*field emission display*" es más específico que "*display*". Pudiéndose inferir una relación "subclase de" entre ambos. Hay que observar que en el caso del idioma inglés, el núcleo del término compuesto (es decir, el subtérmino que define un concepto más general) es el que está situado más a la derecha.

La Figura 10 muestra un ejemplo de cómo a partir de los términos "Field Emission display" y "Video image" puede construirse una jerarquía por composición.

Esta técnica se ha empleado satisfactoriamente en sistemas como OntoLearn [76], OntoLT [17] y Text2Onto [28].

Figura 10. Jerarquía de términos construída por composición léxica

1.4 Obtención de relaciones no taxonómicas

Las relaciones no taxonómicas representan cualquier asociación entre conceptos diferentes de la subclasificación. Ejemplos son la sinonimia, meronimia (“parte de”), antonimia, causalidad y otras relaciones.

En general, son más difíciles de obtener, porque dependen fuertemente del dominio, y no existe un consenso acerca de cuáles son y cómo se definen. A pesar de ello, se han producido varios intentos de identificarlas. A continuación se comentarán brevemente algunas aproximaciones en esta dirección.

1.4.1 Reglas de asociación

El aprendizaje de reglas de asociación [1] es una técnica que se utiliza para descubrir vínculos no explícitos entre variables en grandes conjuntos de datos, y ha sido empleado para obtener de relaciones no taxonómicas en los sistemas Text To Onto [69] y DOODLE-R [95].

El marco formal del algoritmo queda determinado por un conjunto de transacciones $T := \{t_i | i = 1 \dots n\}$, donde cada transacción está formada por un conjunto de elementos $t_i = \{a_{i,j} | j = 1 \dots m_i, a_{i,j} \in C\}$ y cada elemento $a_{i,j}$ pertenece a un conjunto de conceptos C .

El algoritmo tiene como objetivo determinar reglas de asociación del tipo:

$$X_k \Rightarrow Y_k$$

Donde:

X_k es el antecedente de la regla, $X_k \subset C$

Y_k es el consecuente de la regla, $Y_k \subset C$

Y además se cumple $X_k \cap Y_k = \emptyset$

Un ejemplo de regla que podría obtenerse de una base de datos de transacciones de un supermercado sería:

$$\{patatas fritas, snacks\} \Rightarrow \{cerveza\}$$

Que indicaría que existe una alta probabilidad de que alguien que haya comprado patatas fritas y snacks haya incluido también cerveza en su cesta de la compra.

En el contexto del *Ontology Learning*, los elementos a_{ij} serán términos, mientras que las transacciones t_i serán oraciones. Cuando dos términos concurren frecuentemente en una oración, se dice que existe una asociación entre ellos, manifestándose mediante la regla de asociación $X_k \Rightarrow Y_k$ correspondiente.

Existen dos medidas que se emplean para determinar la validez de una regla de asociación. Dichas medidas son el soporte y la confianza.

El soporte representa la proporción de oraciones del corpus que contienen a las palabras asociadas. Su fórmula de cálculo es la siguiente:

$$soporte(X_k \Rightarrow Y_k) = \frac{|\{t_i | X_k \cup Y_k \subseteq t_i\}|}{n}$$

Es decir, se divide el número de oraciones t_i que contienen todas las palabras de la regla (tanto las del antecedente como del consecuente) por el número total de oraciones n .

Por otro lado, la confianza es la proporción de transacciones que contienen tanto las palabras del antecedente como las del consecuente, frente a las que contienen sólo las del antecedente. La fórmula es la siguiente:

$$confianza(X_k \Rightarrow Y_k) = \frac{|\{t_i | X_k \cup Y_k \subseteq t_i\}|}{|\{t_i | X_k \subseteq t_i\}|}$$

Cuando los valores de estas dos medidas superen unos umbrales determinados⁴, se considera que existe una asociación significativa entre los términos, asumiendo que existe una relación de tipo no taxonómico entre los conceptos que representan.

⁴ Por ejemplo, en el sistema DOODLE-R se establecen experimentalmente umbrales de 0,4% para el soporte y 80% para la confianza.

En principio, las relaciones descubiertas mediante esta técnica son anónimas, quedando su etiquetación relegada al criterio de un usuario. Con el fin de evitar esto, Kavalec y Svátek [60] presentan un método para asignar automáticamente un nombre a estas relaciones. Este método se utilizó en el sistema OntoLearn [102].

Se parte de la base de que el nombre ha de ser necesariamente un verbo. Dados los conceptos c_1 y c_2 , de los que se sabe que están relacionados, se buscan verbos en el corpus que ocurran más frecuentemente entre ellos.

El método define el concepto de transacción $VCC(n)$ (*Verb Concept Concept*) entre un verbo v y dos conceptos c_1 y c_2 si y solo si ambos ocurren dentro de una ventana de n palabras alrededor de una ocurrencia del verbo v .

Buenos candidatos para etiquetar una relación no taxonómica entre dos conceptos serán los verbos que ocurran frecuentemente en transacciones $VCC(n)$ con esos conceptos para un valor razonable de n .

En los experimentos llevados a cabo, se estableció heurísticamente el valor 8 a n , teniendo en cuenta de esta forma la presencia de posibles artículos, preposiciones, adjetivos o cláusulas subordinadas en la oración.

Una vez obtenidas las transacciones $VCC(n)$ para un par de conceptos, se determina el grado de asociación con el verbo mediante la medida heurística denominada *above expectation* (AE), cuya fórmula es la siguiente:

$$AE(c_1 \wedge c_2 | v) = \frac{P(c_1 \wedge c_2 | v)}{P(c_1 | v)P(c_2 | v)}$$

Donde:

c_1 y c_2 son los conceptos

v es el verbo

$P(c_i | v)$ es la frecuencia con la que el concepto c_i se encuentra en la misma transacción que el verbo v

$P(c_1 \wedge c_2 | v)$ es la frecuencia condicionada con la que los conceptos c_1 y c_2 se encuentran en la misma transacción que el verbo v , calculándose de la siguiente forma:

$$P(c_1 \wedge c_2 | v) = \frac{|\{t_i | v, c_1, c_2 \in t_i\}|}{|\{t_i | v \in t_i\}|}$$

Donde t_i son las transacciones $VCC(n)$.

1.4.2 Patrones léxico – sintácticos

Hearst no sólo presentó los patrones para obtener la relación concreta de hiponimia, sino que también describió un método general con el fin de descubrir patrones que infirieran otros tipos de relaciones. Los pasos serían los siguientes:

1. Establecer el tipo de relación que se desea inferir.
2. Obtener pares de términos entre los que se sabe se produce dicha relación. Este paso puede ser realizado automáticamente, a partir de lexicones u otros recursos como Wordnet, si disponen de la relación que se pretende inferir.
3. Encontrar en el texto estos pares de términos, de tal manera que se sitúen dentro del mismo contexto sintáctico (por ejemplo, dentro de la misma oración).
4. Analizar los contextos sintácticos, tratando de obtener estructuras comunes que puedan conllevar la definición de un patrón.

Trató además, sin éxito, de aplicar este método para la relación de meronimia. Obtuvo patrones muy simples, que podrían representar otras muchas relaciones, y llegó a concluir que, en general, los patrones lexico-sintácticos son muy dependientes de la lengua sobre la que se aplican, siendo el inglés un idioma más adecuado para inferir la hiponimia que otras relaciones.

A pesar de este aparente fracaso, Berland y Charniak [7] retomaron las investigaciones sobre la relación de meronimia, empleando para ello el mismo método definido por Hearst. De esta forma descubrieron los patrones de la Tabla 6, que básicamente coincidían con los que obtuvo Hearst. La novedad que se introduce en este caso es que cada relación candidata se puntúa de acuerdo a una determinada métrica, de naturaleza estadística.

Tabla 6. Patrones lexico – sintácticos para la meronimia

Patrón	Ejemplo
Whole NN[-PL]'s POS part NN[-PL]	... building's basement...
Part NN[-PL] of PREP {the a} DET mods [JJ NN]* whole NN	...basement of a building...
Part NNin PREP {the a} DET mods [JJ NN]* whole NN	...basement in a building...
Parts NN-PL of PREP wholes NN-PL	...basements of building...
Parts NN-PL in PREP wholes NN-PL	...basements in building...

Con esta métrica, se pretende dar una idea de la certeza con la que se podría afirmar que la parte pertenece o no al todo. La inclusión de este sistema de puntuación permite evitar que se infiera una relación entre dos términos habiéndose instanciado solamente un patrón para ese par de términos.

Tras probarlo sobre un corpus de 100 millones de palabras, se obtuvo una precisión muy similar que para la relación de hiponimia (55%).

1.4.3 Preferencias de selección

El concepto de *preferencias de selección* (*selectional preferences* o *selection restrictions*) hace referencia al hecho de existen ciertas restricciones semánticas sobre los argumentos de determinados elementos lingüísticos. Por ejemplo, el verbo “comer” admite o *prefiere* como objeto argumentos que se refieran a cosas comestibles. Estas restricciones semánticas, sobre todo las que se aplican sobre los verbos, se han empleado para obtener relaciones no taxonómicas entre conceptos. Los sistemas ASIUM [37] y RelExt [88] constituyen aproximaciones representativas que explotan esta característica.

En el caso del sistema RelExt, trata de obtener las relaciones de tipo no taxonómico entre los conceptos de una ontología previamente construída sobre el dominio del fútbol. El proceso comienza con la etiquetación de un corpus formado por crónicas de futbol en alemán, asignando a cada sustantivo⁵ la correspondiente clase (concepto) de la ontología, cuando sea posible. Esta etiquetación se realiza automáticamente, requiriendo una serie de etapas de análisis morfológico y sintáctico, así como de reconocimiento de entidades.

Una vez etiquetado el corpus, se realiza un recuento estadístico de sustantivos, etiquetas y verbos utilizando el indicador χ^2 , obteniendo tres listas ordenadas por relevancia. La Tabla 7

Tabla 7. Sustantivos lematizados ordenados por relevancia

Rank	χ^2 -score	Headnoun	Frequency
1	125245.24	Ball (<i>ball</i>)	6849
2	121888.52	Tor (<i>goal</i>)	7767
3	95003.21	Meter (<i>meters</i>)	5967
4	64157.18	Schuss (<i>shot / drive</i>)	3575
5	57185.76	Eck (<i>corner</i>)	3132
6	45474.96	Strafraum (<i>penalty area</i>)	2298
7	34668.11	Freistoss (<i>freekick</i>)	1752
8	30017.75	Leder (<i>leather / ball</i>)	1561
9	27989.09	Flanke (<i>cross</i>)	1479
10	27414.66	Pfosten (<i>post</i>)	1457

⁵ No se etiqueta cualquier sustantivo, sino únicamente los que actúan como núcleos de sintagmas nominales.

muestra los 10 sustantivos con más puntuación que se encontraron, mientras que la Tabla 8 muestra las 10 primeras clases tras vincular los sustantivos a la ontología. La Tabla 9 muestra los verbos lematizados más relevantes utilizados en el dominio del fútbol.

Tabla 8. Clases de la ontología más relevantes tras la etiquetación y el reconocimiento de entidades del corpus

Rank	Class Label	Frequency
1	FOOTBALLPLAYER	28494
2	GOALOBJECT	8188
3	BALLOBJECT	7249
4	GOALKEEPER	6887
5	SHOOT	3578
6	TEAM	2477
7	PENALTYAREA	2298
8	FREEKICK	1752
9	WING	1482
10	POST	1457

Tabla 9. Verbos lematizados ordenados por relevancia

Rank	χ^2 -score	Predicate	Frequency
1	27167.41	flanken (<i>to cross / to centre</i>)	1373
2	22045.39	klaeren (<i>to clear</i>)	1435
3	21908.37	schieten (<i>to shot</i>)	1503
4	20439.09	koepfen (<i>to head</i>)	1033
5	16342.99	lassen (<i>to let / to leave</i>)	826
6	9563.41	ziehen (<i>to pull / to drag</i>)	1548
7	9468.57	passen (<i>to pass / to play</i>)	814
8	7752.84	spielen (<i>to play / to pass</i>)	1559
9	7653.68	lenken (<i>to divert</i>)	537
10	7637.45	parieren (<i>to parry / to save</i>)	405

Una vez determinados los elementos más relevantes, se examinan las estructuras verbo-argumento del corpus en las que coocuran verbos y sustantivos con una puntuación de relevancia alta. Solo se consideran los argumentos que cumplan la función de sujeto, objeto directo y objeto indirecto.

Sobre estas estructuras se vuelve a aplicar el indicador χ^2 , obteniendo una lista de pares verbo-argumento ordenadas por relevancia. Mediante este cálculo, se determinan las preferencias de selección de cada predicado, que son las restricciones semánticas que se aplican a los sustantivos que actúan como argumentos de cada verbo, según su función gramatical.

Por ejemplo, para los verbos “to cross”, y “to test” se obtienen las preferencias que aparecen en la tabla Tabla 10. Estas restricciones permiten encontrar las clases de los sujetos, objetos directos y objetos indirectos *preferidas* de los verbos.

Tabla 10. Preferencias de selección para los verbos "to cross" y "to test"

Predicate	ARG-CLASS	GF	χ^2
flanken (<i>to cross</i>)	FOOTBALLPLAYER	SUBJ	25.03
flanken	REFEREE	SUBJ	0.05
flanken	FOOTBALLPLAYER	DOBJ	34.77
flanken	REFEREE	DOBJ	0.01
flanken	FOOTBALLPLAYER	IOBJ	10.63
flanken	REFEREE	IOBJ	0.01
pruefen (<i>to try / to test</i>)	FOOTBALLPLAYER	SUBJ	3.09
pruefen	GOALKEEPER	SUBJ	0.63
pruefen	FOOTBALLPLAYER	DOBJ	0.20
pruefen	GOALKEEPER	DOBJ	20.60
pruefen	FOOTBALLPLAYER	IOBJ	0.96
pruefen	GOALKEEPER	IOBJ	7.69

A partir de la información anterior, se puede definir una relación entre dos clases de la ontología de la siguiente forma:

- El nombre de la relación es el verbo.
- El origen de la relación es la clase que suele actuar como sujeto (con una puntuación más alta)
- El destino de la relación es la clase que suele actuar como objeto directo u objeto indirecto (con una puntuación más alta)

Las relaciones que se obtendrían a partir de la Tabla 10 serían las siguientes⁶:

Tabla 11. Ejemplos de relaciones ontológicas obtenidas

Domain	Relation	Range
FOOTBALLPLAYER	flanken (<i>to cross</i>)	FOOTBALLPLAYER
FOOTBALLPLAYER	flanken_auf (<i>to cross to</i>)	FOOTBALLPLAYER
FOOTBALLPLAYER	flanken_zu (<i>to cross to</i>)	FOOTBALLPLAYER
FOOTBALLPLAYER	pruefen (<i>to test</i>)	GOALKEEPER

⁶ *Domain* indica la clase origen de la relación, mientras que *Range* indica la clase destino.

1.4.4 Reglas lingüísticas

Buitelaar et al. [17] proponen un sistema de aprendizaje de ontologías mediante reglas lingüísticas. El sistema en cuestión, denominado OntoLT, se ha implementado como un plugin de la herramienta de desarrollo de ontologías Protégé [78].

El sistema recibe como entrada un corpus de texto en formato XML, ya analizado sintácticamente y anotado, y aplica sobre él un conjunto de reglas especificadas mediante un lenguaje diseñado a tal efecto. El sistema posee un conjunto de reglas predefinidas, aunque se le da al usuario libertad para especificar otras.

En estas reglas, el antecedente es una expresión sobre la estructura sintáctica de la entrada, y el consecuente una acción que se realiza sobre la ontología (inserción de una determinada clase o atributo).

El usuario es libre de especificar las reglas con el propósito que desee: generar clases, añadir un *slot* a una clase, definir el rango de un *slot*, etc. (Protégé denomina a una relación no taxonómica como *slots* de las clases que relaciona).

Por ejemplo, la siguiente regla está predefinida en el sistema:

<p>Nombre: SubjectToClass_PredicateToSlot_DObjToRange</p> <p>Antecedente:</p> <p>containsPath(Predicate) AND containPath(Subject) AND containsPath(DirectObject)</p> <p>Consecuente:</p> <p>CreateCls(Subject) AND AddSlot(Subject, Predicate) AND FillSlot(Subject, Predicate, DirectObject)</p>
--

Mediante esta regla, si se encuentra en la entrada una estructura sintáctica de tipo Sujeto/Predicado(verbo)/Objeto directo, el sistema inserta automáticamente en la ontología una clase cuyo nombre coincide con el sujeto de la estructura, añadiéndole un *slot* (relación no taxonómica) etiquetado con el nombre del verbo y como dominio (destino de la relación) la clase de objetos etiquetada mediante el objeto directo.

Con el fin de utilizar solo información lingüística relevante para el dominio, esta aproximación incluye también un preprocesamiento estadístico en el que se computa la relevancia de las palabras utilizando el indicador χ^2 .

1.5 Deep ontology learning

En las secciones anteriores se han presentado diferentes aproximaciones a la búsqueda de conceptos y relaciones utilizando textos como fuente, pudiéndose clasificar en dos orientaciones. Una de ellas es la estadística, basada en el estudio de las frecuencias de fenómenos presentes en los textos, mientras que la otra es la lingüística, basada en el concepto de patrón lexico-sintactico, con el cual se buscan significados prototípicos a determinadas construcciones sintácticas.

Ambas llegan a resultados intermedios, requiriendo el manejo de textos de entrada bien escritos con el fin de encontrar los patrones con cierta garantía de fiabilidad. En un texto poco claro, el patrón será difícil de encontrar, debido a que depende estrictamente de una construcción léxico sintáctica correcta.

Hasta hoy, no se ha estudiado suficientemente ninguna aproximación orientada a lo que se podría denominar como *deep ontology learning*, encaminada a la identificación de los patrones semánticos que determinan el significado preciso de cualquier aserción escrita. El objetivo fundamental sería deducir los conceptos y relaciones analizando el significado profundo del lenguaje, y no solo la frecuencia estadística o la estructura sintáctica. Este significado profundo no puede estudiarse mediante patrones sintácticos, sino que requiere un análisis semántico dirigido al significado de las aserciones embebidas en el texto. Esta vía es la única que permitiría identificar correctamente los conceptos y las relaciones de cualquier tipo presentes en diferentes dominios.

Según esta visión, el concepto de patrón ya no es superficial (basado en sintaxis) sino profundo (basado en la semántica), orientando la búsqueda de los conceptos y relaciones en función del comportamiento, y no de la forma. Esto último sugiere que tampoco debería depender de la lengua, puesto que ésta se puede considerar como un componente de la forma.

Como se ha visto en secciones anteriores, la mayoría de los métodos presenta una fuerte dependencia de la lengua, quedando en entredicho su validez en entornos multilingües. La única aproximación que permite la representación con éxito de significados de manera independiente a la lengua son las denominadas *interlinguas*. A continuación se describirá este modelo de representación semántica de contenidos.

1.5.1 Interlinguas

Las interlinguas fueron creadas en el área de la traducción automática con el fin de reducir el problema del incremento de sistemas de traducción entre pares de lenguas. Las primeras que se definieron fueron ATLAS [98], propuesta por Fujitsu, y PIVOT [75], propuesta por NEC, ambas paradigmáticas diseñadas con propósito general.

Los sistemas de traducción basados en interlinguas no obtuvieron los resultados deseados, debido a la dificultad existente para representar todos los fenómenos lingüísticos de todos los idiomas, a pesar de lo cual se desarrollaron nuevas interlinguas, esta vez en el marco de los llamados sistemas de traducción automática basados en el conocimiento, como KANT [79] y Mikrokosmos [5].

Posteriormente, en 1996, la Universidad de las Naciones Unidas propuso a un conjunto de algo más de 100 investigadores, que representaban a 14 lenguas y al 98% de la población mundial, la definición formal de una interlingua que fuera capaz de eliminar las barreras del idioma en Internet derivadas tanto de la diversidad lingüística como del uso de una lengua dominante (inglés). Esta interlingua, denominada Universal Networking Language (UNL) [99], se considera la única interlingua normalizada de propósito general sobre la que se ha mantenido actividad investigadora hasta la actualidad.

Desde su concepción, UNL ha demostrado experimentalmente su eficacia en diferentes escenarios multilingües, también como apoyo en la traducción manual de textos. Por ejemplo, Boitet et al. [11] describen un experimento en el que se estudia el rendimiento de un traductor humano al traducir un sitio web a 4 idiomas, ayudándose de varios sistemas de traducción automática. Se constata que los sistemas de traducción mejoran el rendimiento del traductor humano, obteniendo un ratio entre 25 min/pág y 20 min/pág. Sin embargo, UNL permite ir más allá, alcanzándose un ratio de 10 min/pág, si el traductor se apoya en texto generado a partir de expresiones UNL correctas.

Por otra parte, las especificaciones de UNL no solo se limitan a definir una interlingua, sino que también describen un sistema mayor, el denominado *sistema UNL*, en el que la interlingua juega un rol central, y en torno a la cual se especifican los procesos de análisis y generación de lenguaje para cualquier idioma. La arquitectura modular de este sistema ha servido de base para definir metodologías de trabajo con interlinguas, como por ejemplo en [21], donde se propone un plan sistemático de trabajo para el análisis y generación de lenguaje aplicado a textos con lenguaje y vocabulario controlado, concretamente a folletos farmacéuticos, que

son un tipo de documentos caracterizados por un formato rígido y un léxico preciso (los idiomas considerados son el español y el árabe).

El flujo de trabajo que implican las interlinguas permite además hacer visibles ciertos procesos a los que se les ha prestado una menor atención en el área del procesamiento del lenguaje natural, como es el caso de la generación del lenguaje. Este asunto es analizado por Gallardo y Cardeñosa [45], donde se afirma que una de las posibles causas de ello es que la entrada de estos procesos no se ha estandarizado convenientemente, proponiendo a UNL como vía para poder hacerlo.

Pero más allá de su utilización en los escenarios para los que fueron diseñadas (es decir, la traducción automática), las interlinguas se han explotado también atendiendo a su capacidad para representar de contenidos, permitiendo su incorporación a un gran abanico de áreas de la Inteligencia Artificial relacionadas con el procesamiento del lenguaje natural, como la recuperación y extracción de información, sumarización, etc.

La especificación de UNL como formalismo basado en redes semánticas, junto a su condición de interlingua de propósito general, han facilitado su uso como lenguaje de representación del conocimiento. Por ejemplo, en los proyectos HEREIN [29] y AgroExplorer [97] se ha utilizado a UNL para representar colecciones de documentos de texto sobre patrimonio cultural y agricultura, respectivamente. Además, AgroExplorer permite realizar búsquedas de recuperación de información utilizando esta representación, trasladando las preguntas al plano de la interlingua y obteniendo así un buscador que opera directamente sobre el nivel semántico de los documentos.

Las posibilidades de análisis semántico en los textos representados con UNL han sido exploradas también en el área de la extracción de información. En esta línea se pueden encontrar trabajos como los de Cardeñosa et al. [18, 19], donde los autores describen cómo la naturaleza semántica de UNL y su soporte para la deducción permiten acceder a un conocimiento profundo del significado tanto de las preguntas como de los contenidos, haciendo posible la búsqueda de información precisa.

Por otro lado, el uso de la interlingua no solo aporta beneficios al representar contenidos, sino también cuando se usa para representar modelos de conocimiento. La interlingua permite aproximarlos al lenguaje natural, facilitando su vinculación con el texto con independencia del idioma en el que esté escrito. Esto tiene una evidente aplicación en sistemas CLIR (*Cross Language Information Retrieval*), dedicados a la recuperación de información en entornos multilingües.

Generalmente, este tipo de sistemas están basados en una doble arquitectura formada por sistemas a cargo de extraer información junto con los clásicos sistemas de traducción [50]. Cardeñosa et al. [24] presentan la manera en la que UNL puede servir para modelar el conocimiento en este tipo de sistemas dirigidos a dominios concretos.

Un ejemplo de sistema CLIR haciendo uso de interlingua es el proyecto Ontojuris [14], en el que se desarrolla un sistema de información multilingüe para la recuperación de información legal asociada con una ontología sencilla, en la que los conceptos están representados en UNL. El sistema acepta preguntas en un lenguaje, y permite la recuperación de documentos escritos en uno o varios idiomas utilizando medidas de similaridad. El sistema utiliza el modelo representado en UNL para la expansión de la pregunta de forma multilingüe. Esta expansión mejora la búsqueda realizada por el usuario, añadiendo términos adicionales que tienen significado similar a la pregunta original.

En suma, las interlinguas han dejado atrás su propósito original de traducción automática, sirviendo hoy en día para representar significados e implicando por extensión a todo tipo de relaciones existentes en un texto.

Ha habido ya trabajos previos, tanto de quien presenta esta tesis como del equipo de investigación en el que está integrado, que demuestran las capacidades de las interlinguas para representar contenidos, y cómo a partir de éstos se pueden realizar labores de extracción de información en sistemas CLIR y CLIE⁷. En dichos trabajos se constata que la aproximación interlingual permite identificar un nuevo tipo de patrones lingüísticos de carácter semántico, que implican un salto cualitativo y cuantitativo en la deducción de modelos de conocimiento, facilitando además su vinculación con el conocimiento expresado en texto no estructurado. Dichos trabajos se exponen brevemente a continuación.

1.5.2 Patrones lingüísticos en UNL

Los diferentes parámetros de UNL que determinan sus capacidades como lenguaje formal de representación del conocimiento son estudiados por Cardeñosa et al. [23] (ver Tabla 12, extraída del artículo), postulándose su utilización para especificar ontologías multilingües.

⁷ CLIE = Cross Lingual Information Extraction.

Tabla 12. Capacidades de UNL como formalismo de representación del conocimiento

Concepts	
Definition of concepts	yes
Organization into taxonomies	yes
Exhaustive decompositions	no
Disjoint decompositions	no
Not subclass of	yes
Attributes	
Instance attributes	yes
Class attributes	yes
Local/global scope	yes
Restrictions over values	+/-
Default values	no
Not subclass of	
Relations	
Definition of n-ary relations	yes
Constraints over arguments	yes
Type constraints	no
Necessary/optional arguments	+/-
Definition of n-ary relations	yes
Constraints over arguments	yes
Instances	
Instance of concept relation	yes
Facts	yes
Claims about facts	yes
Axioms	no
Functions	no
Reasoning capabilities	+/-

Además, este trabajo es el primero en plantear que pueden reconocerse, en los textos representados mediante UNL, ciertas regularidades en forma de patrones lingüísticos, a partir de los cuales se podrían obtener las piezas de un modelo ontológico. Este planteamiento ha constituido el punto de partida para el desarrollo de esta tesis.

En [22] se comienza a utilizar en la práctica el concepto de patrón lingüístico en UNL, aunque todavía en una concepción relajada del mismo. En dicho trabajo se presenta un método para implementar un sistema de question answering multilingüe, en el que se puede lanzar una pregunta de tipo factoid (Quién?, Dónde?, Cuándo?, ...) en cualquier lenguaje contra una representación en UNL de un conjunto de documentos, obteniendo la respuesta en el mismo idioma en el que se hace la pregunta.

El método se basa en trasladar la pregunta al plano de la interlingua, obteniendo un patrón de búsqueda con una incógnita (ver Figura 11), y utilizar este patrón para localizar la respuesta.

Question:

(EN): "Why was Aubert awarded with Caméré prize?"

(SP): "¿Por qué fue Aubert galardonado con el premio Caméré?"

Figura 11. Representación de una pregunta en UNL

El patrón se relaja eliminando las relaciones entre los nodos de la pregunta salvo la que está vinculada al nodo incógnita, denominada relación principal. La idea es utilizar estos elementos del grafo como coordenadas para localizar la respuesta, buscando en las expresiones UNL de los documentos aquellos subgrafos de tamaño mínimo que contengan al patrón.

La respuesta se determina partir de la instanciación del nodo incógnita (ver Figura 12).

Question: "Why was Aubert awarded with Caméré prize?"

Sentence: "Aubert was awarded the Caméré prize in 1934 by the Academy of Sciences for a new type of movable dam".

Figura 12. Localización de la respuesta en una expresión UNL

Debido a que se encuentra representada en UNL, se puede ofrecer al usuario en su propio idioma sin más que aplicar el proceso de generación correspondiente.

En otra aproximación posterior [20], se utiliza la idea de patrón lingüístico para implementar un sistema de extracción de información elaborada, que fuera capaz de responder con un informe completo a una pregunta de un usuario acerca de un tema o hecho particular. El objetivo es facilitar al usuario el acceso al conocimiento almacenado en una fuente de información, aunque no sepa exactamente qué es lo que debe o puede preguntar. El informe obtenido contendría conocimiento de tipo enciclopédico. El sistema parte igualmente de una colección de documentos representados en UNL, en la que se tratan de localizar estructuras que representan conocimiento de tipo factual. Para ello, se analizan las expresiones de la interlingua en busca de subredes en las que se puedan encontrar vinculadas dos o más entidades (ver Figura 13).

Tanto la forma como la composición de estas subredes están relacionadas con el tipo de entidades que contienen, sugiriendo que pueden ser generalizadas en forma de patrones lingüísticos. En este trabajo se provee además una primera definición de este concepto, circunscrita a UNL:

“A linguistic pattern is the conjunction of semantic relations (in our case, those from the UNL model) and nodes, representing the terms of a domain and linked by the semantic relations, that allows for the expression of events and situations in a semantically coherent way”.

Figura 13. Conocimiento factual en expresión UNL

Los patrones lingüísticos obtenidos mediante esta aproximación (ver Figura 14) son utilizados como herramientas que permiten vincular entre sí hechos concretos diseminados en colecciones de documentos, haciendo posible la composición de informes de conocimiento de tipo enciclopédico como los que se sugiere en esta aproximación.

Figura 14. Patrón lingüístico relacionando personas y ciudades

En resumen, estos trabajos ponen de manifiesto la vinculación de la representación semántica de contenidos con los modelos formales, permitiendo definir una nueva aproximación al *Ontology Learning* sin las limitaciones lingüísticas del uso de idiomas diversos, es decir, en un entorno de multilingüidad.

Se ha utilizado UNL por ser la única interlingua existente normalizada de propósito general, y es la que se tomará en este trabajo como modelo de representación de contenidos semánticos de forma independiente de la lengua. En el Anexo I puede verse una descripción de esta interlingua.

1.6 Conclusiones

Las aproximaciones que se han presentado en las secciones anteriores dirigidas a identificar conceptos y relaciones a partir de textos, no obtienen resultados satisfactorios cuando se aplican sobre dominios abiertos. Si además se requiere combinar conocimientos procedentes de textos escritos en diferentes lenguas, dichas aproximaciones resultan casi inviables en la práctica.

Esto sugiere que para la identificación de modelos de dominio en entornos multilingües es necesario disponer de un modelo de representación interlingual, que permita acceder al significado profundo del lenguaje mediante patrones lingüísticos. Apenas hay trabajos en esta dirección, por lo que esta tesis pretende abrir una nueva vía que permita acometer la

necesidad de crear repositorios de conocimiento sin tener que adecuarse las formas lingüísticas derivadas del uso de una u otra lengua.

Trabajar con una interlingua no evita, sin embargo, la aparición de ciertos fenómenos exclusivos de los lenguajes naturales, como la anáfora, cuya resolución requiere frecuentemente el contexto extralingüístico. Será necesario, por lo tanto, proponer además un modelo para tratar este fenómeno en el entorno de la interlingua.

Por otra parte, también habrá que evaluar hasta qué punto los patrones lingüísticos de carácter semántico son equiparables o compatibles con las estructuras léxico sintácticas estudiadas por otros autores.

Con esta tesis se intentará llegar tan lejos como sea posible en la consecución de estos objetivos, dentro de lo razonable en un trabajo de estas características, y acompañar las tesis con resultados plausibles tras la oportuna experimentación.

Siendo las interlinguas una representación comúnmente aceptada de la semántica, resulta un capricho de las modas del mundo científico el haber obviado su uso como modelo de representación de contenidos, esencialmente de su nivel semántico.

PLANTEAMIENTO DEL PROBLEMA

Como se ha visto en la sección anterior, la inteligencia artificial aborda el problema de la adquisición del conocimiento tratando de automatizar la construcción de conceptualizaciones formales (ontologías), aplicando técnicas avanzadas de análisis allá donde la información se encuentra en un mayor volumen, principalmente en colecciones de documentos de texto.

La gran mayoría de estas técnicas, provenientes de las áreas de la recuperación y extracción de información, tratan de explotar las regularidades superficiales del lenguaje, estudiando su fenomenología a través de la observación de frecuencias y estructuras sintácticas. Ambos tipos de análisis comparten, en mayor o menor medida, una dependencia significativa con el idioma, limitando enormemente sus prestaciones en entornos multilingües donde se requiere combinar los conocimientos deducibles de textos en diferentes idiomas.

El trabajo que se presenta en esta tesis constituye una nueva aproximación a la construcción automática de ontologías a partir de textos, en la que no se analiza la superficie del lenguaje sino su contenido semántico profundo.

Acceder a este nivel del lenguaje permitirá deducir los componentes del modelo formal estudiando regularidades semánticas, expresadas no mediante construcciones sintácticas sino por patrones lingüísticos de carácter semántico.

HIPÓTESIS DE TRABAJO

Las **interlinguas**, como formalismos de representación de significados, servirán de vía al método propuesto para acceder al nivel semántico profundo del lenguaje. En concreto, **se utilizará a UNL (Universal Networking Language) como base**, aunque sus principios podrían aplicarse a cualquier otra, siempre y cuando posea una naturaleza similar.

Las especificaciones de esta interlingua pueden encontrarse en [99], aunque en el Anexo I se realiza una breve introducción de sus características básicas.

La razón fundamental por la que se ha elegido esta interlingua es porque es la única de propósito general que está normalizada, permitiendo representar cualquier fenómeno lingüístico presente en un texto con independencia del dominio.

A pesar de ello, UNL ha sufrido a lo largo del tiempo cierta inestabilidad en sus especificaciones, dificultando su difusión mundial y provocado una presencia escasa en la red. El análisis masivo de contenidos representados en UNL, que ha debido realizarse para el desarrollo de esta tesis, se ha visto limitado justamente por la ausencia de corpus con los que trabajar. Las técnicas empleadas en el método son exclusivamente lingüísticas debido a este motivo, impidiendo su combinación con otros tipos de técnicas, como por ejemplo las estadísticas.

Por otra parte, en cuanto a la **ontología** que se pretende construir, en este trabajo se limita a la **obtención de conceptos y relaciones de tipo taxonómico**, quedando la identificación de relaciones no taxonómicas, atributos y axiomas para trabajos posteriores.

MÉTODO PROPUESTO

El objetivo del método que se presenta en esta tesis es construir automáticamente la ontología del dominio que subyace a una colección de textos, analizando el nivel semántico profundo del lenguaje.

Como se ha comentado, para acceder a este nivel semántico se utilizará el modelo de la interlingua, concretamente a UNL, de forma que el método no tomará como entrada el texto directamente sino su representación interlingual.

El estándar UNL provee, en el ámbito de la interlingua, el concepto de documento UNL como el equivalente a un documento de texto en lenguaje natural. El método tomará por tanto como entrada colecciones de estos documentos con el fin de obtener a partir de ellos el modelo del dominio.

Debido a que UNL es un lenguaje basado en redes semánticas, el proceso de análisis que se realizará en el método estará orientado a detectar las regularidades existentes en las expresiones de la interlingua, generalizándolas en forma de patrones lingüísticos. Estos patrones permitirán descubrir las piezas de la ontología, que en esta tesis se limitan a:

- Conceptos
- Relaciones taxonómicas
- Instancias

Por otra parte, a pesar de que UNL se ha utilizado como un lenguaje de representación del conocimiento (como se ha podido comprobar en el estado del arte), aun se encuentra lo suficientemente cerca de los lenguajes naturales como para presentar fenómenos cuya interpretación depende del contexto, como la anáfora.

En las expresiones anafóricas, como por ejemplo la oración “él me lo dijo”, existen determinadas palabras (“él” y “lo” en el ejemplo) cuyo objeto al que se refieren (antecedente) debe ser deducido por el lector a partir del contexto extralingüístico. Con el fin lograr una mayor comprensión de los contenidos expresados en los documentos y mejorar así la deducción del modelo ontológico, el método incorpora un procedimiento de resolución de la anáfora circunscrito al modelo de la interlingua. El objetivo de este procedimiento es reemplazar los pronombres anafóricos por sus respectivos antecedentes.

El método global propuesto en esta tesis realiza un proceso de análisis de los documentos UNL dividido en tres etapas:

1. Resolución de la anáfora: la resolución de este fenómeno constituye el primer paso del método. En él se realiza un recorrido secuencial de todas las expresiones UNL de entrada, sustituyendo cada pronombre anafórico por su antecedente, si existe. Esta resolución está limitada a los pronombres anafóricos “él” y “ella”, cuando su antecedente es un nombre propio de persona.
2. Preprocesamiento: el objetivo de esta etapa es preparar las expresiones para el reconocimiento de los patrones lingüísticos que se realizará posteriormente, durante la identificación de los componentes de la ontología. Esta preparación consiste en aplicar una serie de transformaciones a cada expresión, con el objetivo por un lado de simplificarlas, eliminando la información lingüística que no es necesaria para el reconocimiento de los elementos ontológicos, y por otro lado de hacer explícitas ciertas relaciones entre palabras deducibles a partir de las especificaciones de la interlingua.
3. Construcción del modelo: una vez preprocesadas, el conjunto de expresiones se recorrerá nuevamente, tratando de identificar en ellas un conjunto de patrones lingüísticos preestablecidos que permiten deducir los elementos del modelo: instancias, conceptos y relaciones taxonómicas.

Debido a que la noción de patrón lingüístico es básica para el método, antes de desarrollar cada una de las etapas anteriores es necesario en primer lugar definir qué se entenderá por patrón lingüístico en el ámbito de UNL, y cómo estos patrones pueden servir como herramientas para la extracción de información.

Con este fin, en la siguiente sección se establecerá un marco formal, adaptando conceptos conocidos de teoría de grafos a la noción de expresión UNL, como grafo dirigido anidado, y la de patrón lingüístico, como un grafo que define implícitamente una operación de encaje sobre expresiones UNL.

Las definiciones de este marco serán fundamentales tanto para la implementación del modelo como para introducir la terminología y la notación que se emplearán en las secciones posteriores, dedicadas a describir cada una de las etapas del método.

4.1 Marco formal

4.1.1 Grafo dirigido con arcos etiquetados

Un *grafo dirigido de arcos etiquetados* G se define como la terna (N, A, E) , donde N es el conjunto de nodos, $A \subseteq (N \times N \times E)$ es el conjunto de arcos y E es el conjunto de etiquetas. Cada arco lleva asociado una (y solo una) etiqueta del conjunto E .

La siguiente figura muestra ejemplo:

Figura 15. Ejemplo de grafo dirigido

El concepto de etiqueta utilizado aquí no equivale al de identificador. Los únicos elementos etiquetados del grafo son los arcos y, según se ve en el ejemplo, varios de ellos pueden contener la misma etiqueta, por lo que no pueden ser identificados exclusivamente por ella.

Lo contrario ocurre con los literales n_i , que no son etiquetas sino identificadores cuya única finalidad es referenciar unívocamente a los nodos.

Otro detalle a destacar de la definición es la mención explícita al conjunto E de etiquetas. Esta mención se debe, en primer lugar, a la necesidad de adaptar la definición al modelo UNL, asimilando la idea de etiqueta con el concepto de *tipo de relación* presente en las expresiones UNL y, en segundo lugar, a que este conjunto de etiquetas interviene en la definición de homomorfismo que se introducirá posteriormente, y que sirve de base para la idea de encaje y patrón lingüístico.

Dado que en lo que resta de sección solo se mencionarán este tipo de grafos, con el fin de simplificar, se emplearán indistintamente los términos “grafo” o “grafo dirigido” como sinónimos de “grafo dirigido con arcos etiquetados”.

Subgrafo

Un grafo $G' = (N', A', E')$ es *subgrafo* de $G = (N, A, E)$ si $N' \subseteq N$, $E' \subseteq E$, $A' \subseteq A$ y $A' \subseteq (N' \times N' \times E')$. Se dice además que es *subgrafo completo* si contiene todos los arcos de G que conectan los nodos de G' , es decir:

$$(u, v, e) \in A \wedge u, v \in N' \Rightarrow e \in E' \wedge (u, v, e) \in A'$$

Siguiendo con el ejemplo de la Figura 15, a continuación se muestran un subgrafo de G no completo (izquierda) y otro completo (derecha), resaltados con respecto al grafo que los contiene:

Figura 16. Subgrafos de G

Camino

Es una sucesión de nodos tal que desde cada uno de ellos existe al menos un arco que lo conecta con el nodo sucesor.

El primer nodo del camino se denomina nodo origen, mientras que el último se denomina nodo destino del camino.

El camino se denominará *dirigido* o *no dirigido*, dependiendo de si en la sucesión de nodos se tiene en cuenta o no el sentido de los arcos. En un camino dirigido, desde cada uno de los nodos debe partir un arco hacia el siguiente.

Ej.: (n_1, n_3, n_6, n_3) y (n_3, n_4, n_6) son ejemplos de caminos no dirigidos en el grafo de la Figura 15, mientras que (n_3, n_1, n_4) es un camino dirigido.

Grafo conexo

Un grafo es conexo si se puede formar un camino desde uno de sus nodos a cualquier otro.

Ej.: el grafo de la Figura 15 no es conexo, ya que desde n_5 y n_7 no pueden establecerse caminos hacia el resto de nodos.

Longitud de un camino

Número de arcos que se utilizan en el camino. Coincide con el número de nodos que contiene menos 1.

Ej.: El camino (n_1, n_3, n_6, n_3) tiene una longitud de 3.

Distancia entre nodos

Es la longitud del camino no dirigido más corto que contiene ambos nodos.

Si no se puede establecer un camino que contenga los dos nodos, la distancia entre ellos será infinita. Esto puede ocurrir en los grafos no conexos (por ejemplo, entre los nodos n_5 y n_6 del grafo de la Figura 15).

Ej.: La distancia entre el nodo n_1 y n_3 es 2.

Homomorfismo entre grafos

Dados los grafos $G = (N, A, E)$ y $G' = (N', A', E')$, se define un *homomorfismo* F desde G hasta G' (denotado como $F: G \rightarrow G'$) al par de correspondencias $f_N: N \rightarrow N'$ y $f_E: E \rightarrow E'$, tal que $(u, v, e) \in A$ implica $(f_N(u), f_N(v), f_E(e)) \in A'$.

Según esto, un homomorfismo es una correspondencia entre grafos que tiene como requisito fundamental la conservación de la adyacencia de los nodos del grafo origen en el grafo destino. La siguiente figura muestra un ejemplo:

Figura 17. Homomorfismo entre grafos

Subgrafo imagen de un homomorfismo

Dados los grafos $G = (N, A, E)$ y $G' = (N', A', E')$ y el homomorfismo $F: G \rightarrow G'$, se define el subgrafo imagen de F al subgrafo de G' , $ImF = (N_{ImF}, A_{ImF}, E_{ImF})$, en el que:

$$N_{ImF} = \{ n \in N' \mid \exists x \in N, f_N(x) = n \}$$

$$E_{ImF} = \{ e \in E' \mid \exists x \in E, f_E(x) = e \}$$

$$A_{ImF} = \{ (u, v, e) \in A' \mid \exists (x, y, z) \in A, f_N(x) = u, f_N(y) = v, f_E(e) = z \}$$

Es decir, el subgrafo imagen está formado por los nodos y arcos de G' que son destino de las correspondencias del homomorfismo. La siguiente figura muestra resaltado el subgrafo imagen del homomorfismo del ejemplo de la Figura 17:

Figura 18. Subgrafo imagen del homomorfismo de la Figura 17

Tipos de homomorfismo. Concepto de encaje.

Las características de las correspondencias f_N y f_E determinan diferentes tipos de homomorfismos. Desde el punto de vista de la formalización de las operaciones de encaje de patrones, tal y como veremos posteriormente, el criterio más importante de clasificación lo proporciona la clase de dichas correspondencias, según sean inyectivas, suprayectivas o biyectivas.

Recordando brevemente estos conceptos, una correspondencia f es inyectiva si a cada elemento del conjunto origen le corresponde un elemento distinto del conjunto destino, es decir:

$$f(x) = f(y) \Rightarrow x = y$$

Por otro lado, una correspondencia f es suprayectiva o sobreyectiva si todos los elementos del conjunto destino son imagen de al menos un elemento del conjunto origen:

$$\forall y \exists x \mid f(x) = y$$

Si una correspondencia es inyectiva y suprayectiva a la vez, se dice que es biyectiva.

En el homomorfismo de la Figura 17, f_N es inyectiva, mientras que f_E no lo es, ya que asigna mismo valor (t_4) a las dos etiquetas (s_1 y s_2) del grafo G .

A continuación se muestran otros ejemplos de homomorfismo, en los que las correspondencias son de diferentes clases:

Figura 19. Diferentes tipos de homomorfismos

De los ejemplos anteriores puede deducirse que cuando f_N no es inyectiva (caso de los homomorfismos F_1 y F_2), el subgrafo imagen (ImF) posee un número de nodos inferior al del grafo origen G , es decir $|N| < |N_{Imf}|$. La consecuencia visible de esto es que G no conserva su estructura una vez trasladado sobre G' , al contrario de lo que ocurre, por ejemplo, con F_2 o el homomorfismo de la Figura 17, donde el grafo origen puede superponerse sobre el destino.

Esta superposición no está condicionada por las etiquetas de los arcos, sino solo por la estructura de los grafos, esto es, los nodos y las relaciones de adyacencia entre ellos. Por esta razón, es una propiedad que no depende de f_E , sino exclusivamente de f_N .

Atendiendo a este criterio estructural, se distinguirán los siguientes tres tipos particulares de homomorfismos entre grafos, según las características de f_N :

- Monomorfismo: si f_N es inyectiva (p. ej. el homomorfismo de la Figura 17)
- Epimorfismo: si f_N es suprayectiva (p. ej. F_1 en la Figura 19)
- Isomorfismo: si f_N es biyectiva (p. ej. F_2 en la Figura 19)

Se prestará especial interés a los homomorfismos en los que f_N sea inyectiva, permitiendo conservar la estructura del grafo origen, es decir, a los monomorfismos, también denominados *encajes*.

Funciones de encaje

Debido a que la noción de patrón lingüístico lleva implícita la idea de operación de encaje, será necesario caracterizar estas operaciones convenientemente, introduciendo lo que se denominará como *función de encaje*.

Dado un homomorfismo entre grafos $F: G \rightarrow G'$, se dice que verifica las funciones $m_N: N \times N' \rightarrow \{1, 0\}$ y $m_E: E \times E' \rightarrow \{1, 0\}$ si se cumple:

$$f_N(u) = v \Rightarrow m_N(u, v) = 1$$

$$f_E(m) = n \Rightarrow m_E(m, n) = 1$$

Las funciones m_N y m_E se denominan funciones de encaje, y permitirán restringir las correspondencias que pueden establecerse entre los nodos y las etiquetas de los grafos G y G' , asignando a cada par el valor 1 si la correspondencia es posible, o bien 0 en caso contrario. Su definición determina implícitamente un conjunto de homomorfismos entre G y G' , formado solo por aquéllos que cumplen las restricciones (es decir, que verifican las funciones de encaje).

Igualdad entre grafos

Dados los grafos $G = (N, A, E)$ y $G' = (N', A', E')$, se dice que son *iguales* según el par de funciones $m_N: N \times N' \rightarrow \{1, 0\}$ y $m_E: E \times E' \rightarrow \{1, 0\}$ si existe un isomorfismo $I: G \rightarrow G'$ que verifica las funciones m_N y m_E .

Obsérvese que la relación de igualdad entre grafos no queda determinada solo por ser isomorfos, sino que también se deben cumplir ciertas restricciones sobre las correspondencias entre nodos y etiquetas, especificadas por las funciones m_N y m_E .

Por ejemplo, los grafos G y G' del segundo ejemplo de la Figura 19 son iguales respecto a las funciones m_N y m_E definidas de la siguiente manera:

$$m_N: \quad m_N(u_1, v_1) = m_N(u_2, v_2) = 1$$

$$m_N(u_i, v_j) = 0 \text{ si } i \neq j$$

$$m_E: \quad m_E(s_1, t_1) = 1$$

Si la función m_N se redefine de forma que $m_N(u_1, v_1) = 0$, entonces ya no se podría hablar de igualdad con respecto a esas funciones. Los grafos siguen siendo isomorfos, ya que puede establecerse un isomorfismo entre ellos, pero dicho isomorfismo no verificaría las funciones de encaje.

Grafo de encaje

Intuitivamente, puede verse un *grafo de encaje* como un grafo especial que lleva definidas implícitamente, en los nodos y las etiquetas, las correspondencias posibles con respecto a los nodos y las etiquetas de otro grafo.

Formalmente, un grafo de encaje M_G sobre un grafo $G = (N, A, E)$ es un grafo $M_G = (N_M, A_M, E_M)$ en el que:

- N_M : es el conjunto de nodos $\{nc_i\}$. Cada uno de ellos lleva asociada una función $h_i: N \rightarrow \{0, 1\}$, que representa una condición de encaje del nodo nc_i con los nodos de G .
- E_M : es el conjunto de etiquetas $\{ec_i\}$. Cada una de ellas lleva asociada una función $g_i: E \rightarrow \{0, 1\}$, que representa una condición de encaje de la etiqueta ec_i con las etiquetas de G .
- A_M : es el conjunto de arcos $A_M \subseteq (N_M \times N_M \times E_M)$.

La siguiente figura representa el esquema de un grafo de encaje, en el que las funciones g_i y h_i se han dejado indicadas:

Figura 20. Ejemplo de grafo de encaje

Un grafo M_G determina, a través de las condiciones que tiene definidas por las funciones h_i y g_i , el conjunto de homomorfismos que pueden establecerse desde M_G a G .

Dado M_G , se define un *encaje de M_G en G* como un homomorfismo $F: M_G \rightarrow G$ que verifica las funciones de encaje $m_N: N_M \times N \rightarrow \{1, 0\}$ y $m_E: E_M \times E \rightarrow \{1, 0\}$ definidas de la siguiente forma:

- $m_N(nc_i, u) = h_i(u)$
- $m_E(ec_i, m) = g_i(m)$

La siguiente figura muestra un ejemplo de grafo de encaje (M_G) sobre un grafo G . Las funciones h_i y g_i se definen en la parte inferior de la figura:

Figura 21. Ejemplo de encaje entre M_G y G

En este ejemplo, un encaje de M_G en G será un homomorfismo que verifique las funciones m_N y m_E definidas de la siguiente forma:

$$\begin{aligned}
 m_N: \quad & m_N(nc_1, v_3) = h_1(v_3) = 1 \\
 & m_N(nc_1, v_i) = h_1(v_i) = 0, i \neq 3 \\
 & m_N(nc_2, v_1) = h_2(v_1) = 1 \\
 & m_N(nc_2, v_2) = h_2(v_2) = 1 \\
 & m_N(nc_2, v_i) = h_2(v_i) = 0, i \neq 1, 2 \\
 & m_N(nc_3, v_4) = h_3(v_4) = 1 \\
 & m_N(nc_3, v_i) = h_3(v_i) = 0, i \neq 4 \\
 m_E: \quad & m_E(ec_1, t_4) = g_1(t_4) = 1 \\
 & m_E(ec_1, t_i) = g_1(t_i) = 0, i \neq 4 \\
 & m_E(ec_2, t_1) = g_2(t_1) = 1 \\
 & m_E(ec_2, t_3) = g_2(t_3) = 1 \\
 & m_E(ec_2, t_i) = g_2(t_i) = 0, i \neq 1, 3
 \end{aligned}$$

$$m_N(nc_3, v_4) = h_3(v_4) = 1$$

$$m_N(nc_3, v_i) = h_3(v_i) = 0, i \neq 4$$

Las flechas discontinuas de la figura representan las correspondencias posibles de los elementos de M_G en los de G , es decir, los pares a los que las funciones m_N y m_E asignan el valor 1.

Según la definición de ambas funciones, solo existen dos encajes posibles de M_G sobre G (el resto de posibilidades no conservan las adyacencias). La definición de ambos homomorfismos sería la siguiente:

1. $f_N(nc_1) = v_3$ $f_E(ec_1) = t_4$
 $f_N(nc_2) = v_1$ $f_E(ec_2) = t_1$
 $f_N(nc_3) = v_4$
2. $f_N(nc_1) = v_3$ $f_E(ec_1) = t_4$
 $f_N(nc_2) = v_2$ $f_E(ec_2) = t_3$
 $f_N(nc_3) = v_4$

El concepto de grafo de encaje se asimilará a la noción de patrón. Los diferentes homomorfismos que define se denominarán *instanciaciones* del patrón.

4.1.2 Grafo dirigido anidado

Debido a que UNL tiene naturaleza recursiva, es decir, un nodo puede contener, a su vez, una expresión, es necesario introducir un nuevo concepto que permita manejar este tipo de estructuras. Este concepto se denominará grafo dirigido anidado.

Formalmente, un *grafo dirigido anidado* G_A se define como la terna (N_A, A_A, E_A) , donde:

- N_A es el conjunto de nodos. Cada nodo puede contener a su vez un grafo dirigido anidado. Los nodos que contienen grafos se denominan *nodos contenedores*.
- $A_A \subseteq (N_A \times N_A \times E_A)$ es el conjunto de arcos
- E_A es el conjunto de etiquetas. Cada arco lleva asociado una (y sólo una) etiqueta del conjunto E_A .

Dado un nodo contenedor n , se denotará mediante $:n$ al grafo anidado que contiene.

La siguiente figura muestra un ejemplo de grafo dirigido anidado:

$G_A = (N, E, A)$	$:n_4 = (N, E, A)$	$:n_5 = (N, E, A)$	$:n_8 = (N, E, A)$
$N = \{n_1, n_2, n_3, n_4, n_5\}$	$N = \{n_6, n_7, n_8\}$	$N = \{n_9, n_{10}, n_{11}\}$	$N = \{n_{12}, n_{13}\}$
$E = \{e_1, e_2, e_3\}$	$E = \{e_4\}$	$E = \{e_5\}$	$E = \{e_6\}$
$A = \{(n_5, n_4, e_1), (n_2, n_4, e_1), (n_2, n_3, e_3), (n_2, n_1, e_2)\}$	$A = \{(n_6, n_7, e_4), (n_7, n_8, e_4)\}$	$A = \{(n_9, n_{10}, e_5)\}$	$A = \{(n_{13}, n_{12}, e_6)\}$

Figura 22. Ejemplo de grafo anidado

Se observa que el grafo G_A presenta hasta dos niveles de anidamiento. El nodo n_4 contiene al grafo $:n_4$, cuyo nodo n_8 contiene, a su vez, al grafo $:n_8$.

Para simplificar la notación de la figura, cada uno de los grafos se ha definido usando la misma terna (N, E, A) . Hay que tener en cuenta que cada uno de ellos es totalmente independiente del resto, es decir, el conjunto E del grafo $:n_4$ (por ejemplo) es distinto al conjunto N de $:n_5$.

Raíz de un grafo anidado

Dado un grafo anidado G_A , su *grafo raíz*, denotado como $\mathcal{R}(G_A)$, es el grafo no anidado obtenido a partir de él.

La siguiente figura muestra el grafo raíz del grafo anidado de la Figura 22:

Figura 23. Grafo raíz de un grafo anidado

Conjunto de nodos contenedores de un grafo anidado

Dado un grafo anidado $G_A = (N, A, E)$, el conjunto de nodos contenedores de G_A , denotado como $\mathcal{N}^c(G_A)$, está formado por los nodos de N que contienen, a su vez, un grafo anidado ($\mathcal{N}^c(G_A) \subseteq N$).

En el grafo anidado de la Figura 22, se definen los siguientes conjuntos de nodos contenedores:

$$\mathcal{N}^c(G_A) = \{n_4, n_5\} \quad \mathcal{N}^c(:n_4) = \{n_8\} \quad \mathcal{N}^c(:n_8) = \emptyset \quad \mathcal{N}^c(:n_5) = \emptyset$$

Conjunto completo de nodos de un grafo anidado

Dado un grafo recursivo $G_A = (N, A, E)$, se define su conjunto completo de nodos $\mathcal{N}^*(G_A)$ como:

$$\mathcal{N}^*(G_A) = N \cup \bigcup_{g \in \mathcal{N}^c(G_A)} \mathcal{N}^*(:g)$$

La definición anterior indica que el conjunto completo de nodos está formado por los nodos del grafo raíz más los nodos de todos los grafos que contiene G_A , ubicados en cualquier nivel de anidamiento.

Siguiendo con el ejemplo de la Figura 22, a continuación se muestran los conjuntos completos de nodos de los grafos anidados:

$$\mathcal{N}^*(:n_8) = \{n_{12}, n_{13}\}$$

$$\mathcal{N}^*(:n_5) = \{n_9, n_{10}, n_{11}\}$$

$$\mathcal{N}^*(:n_4) = \{n_6, n_7, n_8\} \cup \mathcal{N}^*(:n_8)$$

$$\mathcal{N}^*(G_A) = \{n_1, n_2, n_3, n_4, n_5\} \cup \mathcal{N}^*(:n_4) \cup \mathcal{N}^*(:n_5)$$

Conjunto completo de etiquetas de un grafo anidado

Dado un grafo recursivo $G_A = (N, A, E)$, su conjunto completo de etiquetas $\mathcal{E}^*(G_A)$ se define como:

$$E^*(G_A) = E \cup \bigcup_{g \in \mathcal{N}^c(G_A)} E^*(:g)$$

De la Figura 22 se deducen los siguientes conjuntos completos de etiquetas:

$$\mathcal{E}^*(G_A) = \{e_1, e_2, e_3, e_4, e_5, e_6\}$$

$$\mathcal{E}^*(:n_4) = \{e_4, e_6\}$$

$$\mathcal{E}^*(:n_5) = \{e_5\}$$

$$\mathcal{E}^*(:n_8) = \{e_6\}$$

Grafo anidado conexo

Un grafo anidado G_A es conexo si:

- $\mathcal{R}(G_A)$ es conexo
- $\forall x \in \mathcal{N}^c(G_A), :x$ es un grafo anidado conexo

Es decir, un grafo anidado es conexo si lo es su grafo raíz y también lo son todos los grafos que contiene.

Por ejemplo, el grafo anidado :n4 de la Figura 22 es conexo, mientras que G_A no lo es, debido a que :n5 no es conexo.

Encajes sobre grafos anidados

Según se vio en la sección anterior, un homomorfismo entre dos grafos se puede definir informalmente como una correspondencia entre los nodos y etiquetas de ambos grafos, de forma que el grafo de origen conserva las adyacencias una vez trasladado al grafo de destino.

Aunque esta misma idea podría extrapolarse también para grafos anidados, una definición de homomorfismo generalizada de esta forma no tendría utilidad para el objetivo de esta tesis, debido a que el método propuesto no requiere establecer correspondencias entre estructuras recursivas de este tipo.

No obstante, sí que es necesario especificar cómo se establecen correspondencias entre un grafo no anidado y un grafo anidado, para lo cual se reutilizará el concepto de grafo de encaje.

Un *grafo de encaje* sobre un grafo anidado $G_A = (N, A, E)$ se define como el grafo $M_{GA} = (N_M, A_M, E_M)$, no anidado, en el que:

- N_M : es el conjunto de nodos $\{nc_i\}$. Cada una de ellos lleva asociada una función $h_i: \mathcal{N}^*(G_A) \rightarrow \{0, 1\}$, que representa una condición de encaje de nc_i con los nodos de G_A
- E_M : es el conjunto de etiquetas $\{ec_i\}$. Cada una de ellas lleva asociada una función $g_i: \mathcal{E}^*(G_A) \rightarrow \{0, 1\}$, que representa una condición de encaje de ec_i con las etiquetas de G_A .
- A_M : es el conjunto de arcos $A_M \subseteq (N_M \times N_M \times E_M)$.

Un *encaje* de M_{GA} sobre el grafo anidado $G_A = (N, A, E)$ es cualquiera de los homomorfismos siguientes:

- Un encaje de M_{GA} sobre $\mathcal{R}(G_A)$
- Un encaje de M_{GA} sobre $n \mid n \in \mathcal{N}^c(G_A)$

Los aspectos principales de la definición anterior son los siguientes:

- El grafo de encaje establece, en sus nodos y etiquetas, condiciones de encaje sobre todos los nodos y etiquetas del grafo anidado, independientemente del nivel de anidamiento en el que se encuentren.
- Los encajes de M_{GA} sobre G_A se restringen a un nivel de anidamiento. Esto quiere decir que a dos nodos diferentes de M_{GA} no pueden corresponderles dos nodos de G_A ubicados en distintos niveles de anidamiento.

La siguiente figura muestra un ejemplo de grafo de encaje sobre un grafo anidado, junto a los encajes que pueden establecerse entre ambos según la definición de las funciones h_i y g_i :

Figura 24. Ejemplo de encaje sobre un grafo recursivo

Los dos homomorfismos posibles (F_1 y F_2) están representados mediante las flechas discontinuas. Ambos cumplen la condición de estar restringidos a un nivel de anidamiento: F_1 se establece sobre el grafo raíz $\mathcal{R}(G_A)$, mientras que F_2 se establece sobre el grafo raíz $\mathcal{R}(n_3)$.

4.1.3 Expresión UNL como grafo anidado

Según las especificaciones de UNL, las expresiones de la interlingua presentan una estructura recursiva que puede definirse en términos de un grafo anidado, tal y como se ha descrito en la sección anterior.

Dados los siguientes conjuntos:

- ATR_{UNL} : conjunto de atributos UNL
- VOC_{UNL} : conjunto de palabras universales de UNL (vocabulario)
- TR_{UNL} : conjunto de tipos de relaciones de UNL

Se define una expresión UNL, $e_{UNL} = (N, A, E, f_{atr}, f_{uw}, n_p)$, como un grafo anidado conexo sin bucles, donde:

- N : conjunto de nodos. Cada nodo puede contener, a su vez, una expresión UNL.
- E : conjunto de tipos de relaciones, $E \subseteq TR_{UNL}$.
- A : conjunto de arcos. $A \subseteq (N \times N \times E)$.
- $f_{atr}: N \rightarrow \mathcal{P}(ATR_{UNL})$ es una función que asigna un conjunto de atributos UNL a cada nodo.
- $f_{uw}: N - \mathcal{N}^c(e_{UNL}) \rightarrow VOC_{UNL}$ es una función que asigna una palabra universal a cada nodo que no contenga, a su vez, una expresión UNL.
- $n_p \in N$, es el denominado nodo principal de la expresión (denotado como $n_p(e_{UNL})$), y cumple las siguientes condiciones:
 - o '@entry' $\in f_{atr}(n_p)$
 - o $\exists! n \in N \mid "@entry" \in f_{atr}(n)$

Es decir, el nodo principal tiene asignado el atributo "@entry", y no existe ningún otro nodo de N que contenga dicho atributo.

La definición anterior añade tres elementos nuevos con respecto a la definición de grafo recursivo. Por un lado, las funciones f_{atr} y f_{uw} , que caracterizan a cada nodo con una palabra universal (perteneciente al vocabulario de UNL) y con un conjunto de atributos respectivamente, y por otro, el nodo n_p , que se denomina nodo principal y se garantiza que sea único.

Por ejemplo, en la siguiente expresión UNL:

Figura 25. Ejemplo de expresión UNL

Los elementos de la definición anterior son:

$$\begin{array}{ll}
e_{UNL} = (N, A, E, f_{atr}, f_{uw}, n_p) & :n_3 = (N, A, E, f_{atr}, f_{uw}, n_p) \\
N = \{n_1, n_2, n_3\} & N = \{n_4, n_5, n_6\} \\
E = \{agt, obj\} & E = \{agt, plc\} \\
A = \{(n_2, n_1, agt), (n_2, n_3, obj)\} & A = \{(n_4, n_5, plc), (n_4, n_6, agt)\} \\
f_{uw}: f_{uw}(n_1, \text{"police"}), f_{uw}(n_2, \text{"watch"}) & f_{uw}: f_{uw}(n_4, \text{"hide"}), f_{uw}(n_5, \text{"house"}), f_{uw}(n_6, \text{"criminal"}) \\
f_{atr}: f_{atr}(n_2) = \{\text{"@entry"}, \text{"@past"}\} & f_{atr}: f_{atr}(n_4) = \{\text{"@past"}, \text{"@complete"}\}, \\
n_p = n_2 & f_{atr}(n_5) = \{\text{"@entry"}, \text{"@def"}\}, \\
& f_{atr}(n_6) = \{\text{"@def"}\} \\
& n_p = n_5
\end{array}$$

4.1.4 Camino y distancia en una expresión UNL. Coordinación lingüística.

Recordando la definición de camino realizada para el caso de grafos no anidados, un camino es una sucesión de nodos tal que cada uno de ellos está conectado con el siguiente mediante un arco. Dado que en un grafo anidado no existen arcos que conecten nodos de niveles diferentes de anidamiento, los caminos en los grafos anidados quedan restringidos únicamente al grafo raíz.

Sin embargo, una expresión UNL muestra cómo se relacionan todas las palabras de una oración, por lo que parece razonable poder establecer caminos entre dos nodos cualesquiera, aunque estén ubicados en diferentes *scopes*.

Para poder realizar esto, se deberá extender la definición anterior de camino, particularizándola al caso de expresiones UNL:

Dada una expresión UNL e , se define un *camino* como una sucesión de nodos $n_1 \dots n_m$ ($n_i \in \mathcal{N}^e(e)$) en la que cada transición $n_i - n_{i+1}$ cumple cualquiera de las condiciones siguientes:

1. n_i está conectado con n_{i+1} mediante un arco
2. n_i está conectado con un nodo contenedor n_k mediante un arco, y $n_p(:n_k) = n_{i+1}$.
3. Siendo n_k un nodo contenedor, se cumple que $n_p(:n_k) = n_i$ y además n_k está conectado a n_{i+1} mediante un arco.

La primera condición se corresponde con la definición tradicional de camino para grafos no anidados, mientras que las nuevas condiciones 2 y 3 son las que hacen posible que los caminos puedan cruzar dos niveles contiguos de anidamiento, utilizando los nodos principales como enlaces.

Por ejemplo, en la expresión UNL de la Figura 25 puede establecerse el camino (n_1, n_2, n_5, n_4) , en el que las transiciones $n_1 - n_2$ y $n_5 - n_4$ cumplen la condición 1, mientras que la transición $n_2 - n_5$ cumple la condición 2. Obsérvese que el nodo contenedor n_3 , que es el que permite el enlace entre el nivel superior y el inferior, no forma parte del camino.

Los caminos definidos de esta forma podrán ser también dirigidos o no dirigidos, dependiendo de si en la sucesión de nodos se tiene en cuenta o no el sentido de los arcos.

Una vez definido el concepto de camino en expresiones UNL, se puede derivar de él la idea de distancia entre nodos, sin más que reutilizar la definición que se presentó en la sección 4.1.1:

Dados dos nodos u y v de una expresión UNL e ($u, v \in \mathcal{N}^*(e)$) se define la distancia entre ellos como la longitud del camino no dirigido más corto que los contiene, siendo la longitud de un camino el número de arcos que intervienen en él. Por ejemplo, la distancia entre los nodos n_1 y n_4 de la expresión de la Figura 25 es 4.

Sin embargo, la distancia definida de esta manera puede no ser apropiada en el contexto de las expresiones UNL. El problema es que existen ciertas normas de codificación UNL que permiten omitir relaciones entre nodos, si éstas pueden deducirse del significado de otras relaciones. Es el caso de la codificación de la coordinación.

Por ejemplo, las siguientes expresiones UNL representan dos casos de coordinación lingüística:

Figura 26. Estructuras de coordinación en UNL

En la expresión e_1 , se representa una coordinación disyuntiva de dos acciones (“go” y “buy”), ambas con el mismo agente (“Bill”). Por convención, puede especificarse que el agente es común a ambas acciones agrupándolas en un *scope* y añadiendo la relación “agt” desde el *scope* al agente, tal y como se ha hecho en este ejemplo⁸.

Si aplicamos la definición de distancia anterior, obtendremos que “Bill” se encuentra a distancia 2 de “restaurant” y a distancia 3 de “book”. Desde el punto de vista lingüístico, “book” y “restaurant” son argumentos de acciones iniciadas por el mismo agente “Bill”. Ambos deberían estar a la misma distancia, atendiendo a este criterio.

El caso de la expresión e_2 es similar, salvo que se trata de una disyunción. Tanto “Wednesday” como “Monday” se refieren al momento en el que se realiza la acción “come”. Sin embargo, solo se ha añadido la relación que lo indica, “tim”, con “Wednesday”. La norma permite deducir que “Monday” también actúa como complemento de tiempo de “come”, gracias a la presencia del nexo disyuntivo con “Wednesday”. A pesar de ello, esto no se refleja al calcular la distancia, ya que “he” está a distancia 2 de “Wednesday” y a 4 de “Monday”.

La forma más sencilla de solucionar esta inconveniencia consiste en no contabilizar los nexos a la hora de calcular la distancia entre dos nodos.

Las expresiones de la figura anterior muestran los dos tipos de nexos coordinativos que se pueden encontrar en UNL:

- Nexo simple: formado por una única relación de cualquiera de los dos tipos básicos que provee UNL para representar la coordinación: “and” u “or”. Es el caso de la expresión e_1 del ejemplo.
- Nexo compuesto: formado por una relación “and” u “or” más una palabra universal auxiliar, que se relaciona con el elemento coordinado mediante la relación “obj”. La palabra universal auxiliar matiza el significado de la coordinación. Es el caso de la expresión e_2 del ejemplo.

Se definirá entonces la *distancia lingüística* entre dos nodos de una expresión UNL como el número de arcos del camino más corto entre ambos, sin contar las transiciones que representen nexos coordinativos (simples y compuestos).

⁸ Posteriormente, en la sección 4.3.1, se verá con más de detalle la representación de las estructuras coordinadas.

En lo sucesivo, a la distancia lingüística se la denominará simplemente como distancia, siempre que se defina sobre nodos de una expresión UNL.

4.1.5 Patrón lingüístico

Un patrón lingüístico P_{UNL} es un grafo de encaje sobre una expresión UNL, en el que:

1. Las funciones $h_i: \mathcal{N}^*(e_{UNL}) \rightarrow \{0, 1\}$ están definidas mediante dos condiciones C_i^{uw} y C_i^{atr} sobre palabras universales y conjuntos de atributos respectivamente, de forma que:
 - a. $h_i(x) = 1$ si x no es *scope* $\wedge C_i^{uw}(f_{uw}(x)) \wedge C_i^{atr}(f_{atr}(x))$
 - b. $h_i(x) = 1$ si x es *scope* $\wedge C_i^{uw}(f_{uw}(n_p(:x))) \wedge C_i^{atr}(f_{atr}(x) \cup f_{atr}(n_p(:x)))$
 - c. $h_i(x) = 0$ en otro caso
2. Las funciones $g_i: E \rightarrow \{0, 1\}$ representan condiciones de encaje de las etiquetas ec_i con los tipos de relaciones de una expresión UNL.
3. Cada nodo nc_i tiene asociado adicionalmente dos conjuntos $r_i^{\rightarrow}$ y $r_i^{\leftarrow}$ de tipos de relaciones, que definen *restricciones de adyacencia* adicionales a los nodos nc_i una vez trasladados a la expresión UNL.

Un encaje de un patrón P_{UNL} en una expresión UNL será un homomorfismo que verifique las condiciones indicadas por los puntos 1 y 2 de la definición anterior (tal y como ocurre en el caso general de grafo de encaje), y que además cumpla las restricciones establecidas por los conjuntos $r_i^{\rightarrow}$ y $r_i^{\leftarrow}$, de forma que los nodos de la expresión UNL que correspondan con los nodos nc_i del patrón, según el homomorfismo, no podrán poseer relaciones entrantes y salientes con otros nodos si el tipo de éstas pertenece a los conjuntos $r_i^{\rightarrow}$ y $r_i^{\leftarrow}$ respectivamente. Estas restricciones no afectan a las relaciones que forman el propio patrón.

Por otro lado, se observa que las funciones $h_i(x)$ distinguen si el nodo x de la expresión UNL es un *scope* o no. En caso de serlo, se tiene en cuenta también el nodo principal del *scope*.

A continuación se ilustrará la definición anterior mediante varios ejemplos.

La siguiente figura muestra un patrón lingüístico que representa una estructura de verbo – agente:

Figura 27. Patrón lingüístico

Según la definición:

- El nodo nc_1 encajará con un nodo de una expresión UNL solo si dicho nodo contiene una palabra universal verbal, y tiene el atributo "@past"⁹.
- La etiqueta ec_1 encajará con una etiqueta de una expresión UNL si representa un tipo de relación "agt".
- El nodo nc_2 encajará con cualquier nodo (independientemente de la palabra universal y atributos que contenga) que no posea relaciones salientes de tipo "mod" ni entrantes de tipo "agt", distintas a la indicada por ec_1 .

Con el fin de simplificar la notación de los gráficos, se sustituirá la representación formal de las condiciones de encaje y las restricciones de adyacencia por una especificación abreviada.

El patrón anterior se podrá representar mediante la siguiente figura:

⁹ El atributo '@past' indica que la acción indicada por la palabra universal se ubica en tiempo pasado.

Figura 28. Notación simplificada de patrón lingüístico

En la que los nodos y arcos son etiquetados con una especificación abreviada de las condiciones de encaje:

- Se ha indicado con “V @past” la condición “palabra universal verbal con atributo ‘@past’”
- Se ha indicado la condición “cualquier nodo” como “*”
- La condición “relación de agente” se muestra simplemente con el tipo de relación UNL correspondiente (agt).

Otros ejemplos de condiciones abreviadas son las siguientes:

Tabla 13. Condiciones abreviadas en patrones lingüísticos

Condición	Notación abreviada
“palabra universal sustantivo con el atributo ‘@pl’”	N @pl
“cualquier palabra universal con el atributo ‘@pl’”	@pl
“cualquier palabra universal adjetivo”	ADJ
“palabra universal ‘he(icl>person)’ con cualquier atributo”	“he(icl>person)”
“palabra universal cuya <i>headword</i> sea ‘open’”	‘open’
“palabra universal verbal con restricción icl>be”	V (icl>be)
“cualquier palabra universal sin el atributo ‘@def’”	¬@def
“relaciones ‘and’ y ‘or’”	and, or
“cualquier tipo de relación”	*
“cualquier relación que no sea ‘cnt’”	¬cnt

Por otra parte, las restricciones de adyacencia indicadas por los conjuntos $r_i^{\rightarrow}$ y $r_i^{\leftarrow}$ se representan respectivamente con un arco entrante y otro saliente al nodo, etiquetados con los tipos de relaciones que contiene cada uno, con las siguientes excepciones:

- Si un conjunto contiene todas las relaciones UNL (es decir, es igual a TR_{UNL}), entonces el arco correspondiente se etiqueta con el símbolo “*”.
- Si ambos conjuntos son iguales a TR_{UNL} , entonces no se utilizan los dos arcos, sino que se marca el contorno del nodo con trazo grueso. Estas condiciones de adyacencia implicarán que el nodo no deberá poseer ninguna relación adicional a la indicada por el patrón, una vez trasladado a la expresión UNL.
- Si un conjunto es vacío, no se añade el arco correspondiente.

En la siguiente figura se esquematizan visualmente los casos anteriores:

Figura 29. Notación simplificada de restricciones de adyacencia

Un patrón define un conjunto de homomorfismos o encajes posibles con las expresiones UNL. El subgrafo imagen de cada homomorfismo se denomina *instanciación del patrón*.

La siguiente figura muestra la única correspondencia posible del patrón de la Figura 28 con la expresión que representa la oración “El gato cazó un ratón”¹⁰:

¹⁰ Con el fin de simplificar las figuras, no se muestran las restricciones de las palabras universales.

Figura 30. Ejemplo de encaje de un patrón lingüístico

Otro ejemplo de correspondencia posible se muestra en la siguiente figura:

Figura 31. Ejemplo de encaje de un patrón lingüístico (II)

En este caso, el nodo 1 del patrón encaja con el *scope*, ya que $C_1^{uw}(\text{"catch"}) \wedge C_1^{atr}(\{"@past"\})$, por lo que se cumple la condición $C_1^{uw}(f_{uw}(n_p(:01))) \wedge C_1^{atr}(f_{atr}(:01) \cup f_{atr}(n_p(:01)))$

Las restricciones adicionales de adyacencia que establece el patrón (conjuntos $r_i^{\rightarrow}$ y $r_i^{\leftarrow}$) se aplicarían en un caso como el siguiente, en el que no es posible el encaje si se añade el adjetivo “negro” calificando a “gato” mediante la relación “mod”. Del nodo 2 del patrón no puede salir ningún arco de este tipo, ya que pertenece al conjunto $r_2^{\leftarrow}$:

Figura 32. Encaje no válido según restricciones de adyacencia

4.2 Resolución de la anáfora pronominal

4.2.1 Anáfora en UNL

Al igual que en los lenguajes naturales, en UNL pueden aparecer también construcciones anafóricas, sin que el lenguaje ofrezca un medio para especificar cuáles son sus antecedentes. Su utilización sigue, por tanto, siendo necesaria en UNL para garantizar la coherencia del discurso, y las diferentes formas que puede adquirir se tipifican exactamente de la misma forma que en los idiomas como el español o el inglés, es decir, anáforas pronominales, léxicas, adverbiales, etc.

Sin embargo, la frecuencia de aparición de la anáfora en UNL es, en general, inferior que en dichos idiomas, ya que existen determinadas construcciones anafóricas cuya codificación en UNL puede resultar innecesaria. Esto ocurre con ciertas anáforas pronominales, cuando la referencia a su antecedente puede codificarse implícitamente mediante una relación UNL, o

bien si tanto la anáfora como su antecedente pueden ocupar el mismo lugar dentro de la expresión, esto es, se pueden representar utilizando el mismo nodo.

Como se verá a continuación, estas condiciones sólo pueden darse si ambos elementos (anáfora y antecedente) se encuentran en un mismo contexto sintáctico, de forma que pueden codificarse en UNL dentro del mismo *scope*.

Un caso de “disolución” de la anáfora en UNL lo constituyen, por ejemplo, las anáforas formadas por pronombres en genitivo (adjetivos posesivos en español). Por ejemplo, la oración:

“Juan leyó su libro”

Contiene la anáfora “su” que hace referencia al poseedor del libro. Según el contexto, este poseedor será “Juan”, o bien otro participante mencionado con anterioridad. Dependiendo del caso, la oración tendrá una codificación en UNL diferente.

Si el poseedor es Juan, la expresión en UNL sería:

Figura 33. Expresión UNL con relación de posesivo

Donde se observa que “su” ha sido codificado mediante la relación “pos” (poseedor), eliminando la ambigüedad referencial y la necesidad, por tanto, de conservar la anáfora.

Esto ha sido posible porque el antecedente “Juan” está codificado en el mismo *scope* que el elemento poseído “libro”, y ha podido establecerse la relación entre ambos dentro de la misma expresión. Si, por el contrario, dicho poseedor hubiera sido otro participante, no habría podido establecerse tal relación, por lo que habría sido necesario conservar la anáfora mediante un pronombre personal (“he(icl>person)” o “she(icl>person)”):

Figura 34. Expresión UNL con anáfora

El resto de casos en los que la anáfora pronominal desaparece en UNL son las introducidas por los pronombres relativos, los pronombres reflexivos y los pronombres personales, estos últimos cuando su antecedente está en el mismo contexto sintáctico. Por ejemplo:

“Juan leyó el libro que le prestaron”

Posee dos pronombres anafóricos: “que” y “le”. El primero es un pronombre relativo que introduce la cláusula subordinada, y tiene como antecedente a “libro”, mientras que el segundo es un pronombre personal átono que indica la persona a quien prestaron el libro, que podrá ser “Juan” u otro participante que no se menciona en la expresión.

Si estamos en el primer caso, es decir, prestaron el libro a “Juan”, la expresión UNL correspondiente sería:

Figura 35. Eliminación de pronombre anafórico

Donde se observa que los pronombres mencionados han desaparecido.

Esencialmente, lo que ocurre es que la presencia de los pronombres “le” y “que” en la oración original fue necesaria porque sus respectivos antecedentes no pudieron desempeñar por sí solos funciones sintácticas en predicaciones diferentes: “Juan” y “libro” son el sujeto y el objeto directo de la predicación principal “leer”, y no pueden desempeñar funciones en la segunda predicación (“prestar”), siendo necesario introducir pronombres que los sustituyan para ese fin.

Esta limitación no existe en UNL, donde una misma palabra universal puede realizar varias funciones determinadas por las relaciones en las que participa. Así, la palabra universal “book” de la expresión es objeto tanto de “read” como de “borrow”, y “Juan” es agente de “read” y co-agente de “borrow” simultáneamente.

Debe destacarse que la función múltiple de una misma palabra universal es posible sólo si las palabras universales con las que se relaciona se encuentran en el mismo *scope*.

Es perfectamente válido, según las reglas de UNL, codificar la expresión anterior incluyendo en un *scope* la oración subordinada:

Figura 36. Aparición de pronombre anafórico por inserción de *scope*

En esta nueva codificación, no ha sido posible eliminar la anáfora introducida por el pronombre personal átono “le”, ya que ha sido necesario codificar el beneficiario de “borrow” mediante el pronombre personal “he”, debido a que no es posible establecer una relación entre palabras universales de diferentes *scopes*.

La conclusión que podemos obtener es que la forma o el estilo en la codificación en UNL pueden influir en la aparición o no de determinados tipos de anáfora, sobre todo cuando hay que decidir si utilizar o no un *scope* en la expresión. Por este motivo, es importante seguir ciertas recomendaciones con respecto a los *scopes*, que tienden a restringir su uso a menos que sea estrictamente necesario. En este caso, aunque ambas expresiones son correctas, se prefiere la primera expresión, ya que evita la ambigüedad referencial que se introduce en la segunda con la inclusión del pronombre “he”.

La limitación gramatical que fuerza el uso de pronombres personales también se presenta con los pronombres reflexivos. Su presencia en una oración es debida a que una palabra no puede ser sujeto y complemento del verbo al mismo tiempo. Por ejemplo, en:

“María trabaja para sí misma”

Es necesario introducir el pronombre reflexivo “sí (misma)”, ya que “María” ya está desempeñando la función de sujeto.

En la codificación en UNL de la oración se observa que el nodo “María” realiza ambas funciones simultáneamente, por lo que no hace falta la introducción del pronombre:

Figura 37. Expresión UNL

De todo lo comentado anteriormente, se puede concluir que una anáfora desaparece en UNL si:

- a) La anáfora es un pronombre relativo que introduce una cláusula en una oración compuesta.
- b) La anáfora es un pronombre personal, y tanto ella como su antecedente se sitúan en un mismo contexto sintáctico, de forma que se codifican en UNL dentro de un mismo *scope*.

4.2.2 Algoritmo de resolución

En general, entender los mecanismos de cohesión del discurso, como la anáfora, proporciona información valiosa para identificar las entidades individuales o instancias que se mencionan en el texto y qué es lo que se dice de ellas.

Como se ha visto en la sección anterior, en los documentos UNL la frecuencia de aparición de la anáfora es levemente inferior que en otros lenguajes, ya que ciertas anáforas intraoracionales no tienen correspondencia en las expresiones UNL. Sin embargo, su presencia sigue siendo importante, por lo que se ha decidido incorporar al método propuesto un procedimiento automático que la reduzca en la medida de lo posible.

El procedimiento que se expondrá en esta sección, está centrado en la resolución de la anáfora pronominal en UNL, concretamente en la resolución de los pronombres anafóricos “he” y “she” cuando sus antecedentes son personas mencionadas utilizando nombres propios. Esto quiere decir que se podrán resolver anáforas del tipo siguiente:

“**María** se marchó de la clase. El resto de alumnos salió tras **ella**”

pero no se podrá resolver:

“El **profesor** se marchó de la clase. El resto de alumnos salió tras **él**”

ya que en el segundo caso, el antecedente del pronombre personal no es un nombre propio, sino el sustantivo “profesor”.

Las características del algoritmo son las siguientes:

- El objetivo del algoritmo está muy delimitado (anáfora pronominal de pronombres “he”/“she”, cuando el antecedente es un nombre propio de persona), y está orientado exclusivamente a mejorar los resultados de la construcción de la ontología.
- Resuelve tanto anáforas intraoracionales (antecedente y pronombre en la misma expresión) como interoracionales (antecedente en una expresión anterior a la anáfora).
- Es conservador: busca precisión frente a cobertura.
- Es sencillo: solo inspecciona la estructura de las expresiones UNL, sin realizar sobre ellas ningún procesamiento previo y sin utilizar información externa.

La ausencia de ambigüedad y la unicidad en la forma de representar tanto pronombres como nombres propios en UNL hacen posible que su localización sea inmediata. Los pronombres “he” y “she” poseen formas diferenciadas según se refieran o no a personas. Encontrar las palabras universales “he(icl>person)” o “she(icl>person)” garantiza que sus antecedentes

sean personas, diferenciándose de las formas “he(icl>thing)” o “she(icl>thing)”, referidos a animales o cosas.

Por otro lado, en UNL es también fácil identificar los nombres propios de personas. Éstos aparecerán representados mediante una única palabra universal con la restricción “(iof>person)”¹¹.

El algoritmo recorre las expresiones de un documento secuencialmente, aplicando los siguientes pasos para cada una de ellas:

1. Localiza los nodos que contienen nombres propios de personas (conjunto N).
2. Localiza los nodos que contienen pronombres personales (conjunto P).
3. Para cada nodo de P (pronombre):
 - 3.1. Obtiene el conjunto C de nodos candidatos a antecedente ($C \subset N$), descartando de N los nodos que no cumplan ciertas *restricciones*.
 - 3.2. Utiliza *preferencias* para seleccionar de C el antecedente de la anáfora.
 - 3.3. Si no se ha encontrado un antecedente, comprueba si el *foco actual del discurso* puede serlo.
 - 3.4. Si finalmente se ha encontrado un antecedente al pronombre:
 - 3.4.1. Sustituye el pronombre por el nombre propio correspondiente.
 - 3.4.2. Añade al conjunto N el nodo que contenía al pronombre, ya que ahora contiene un nombre propio.
4. Una vez explorados los nodos de P , determina el nuevo *foco del discurso*.

El concepto de *foco del discurso* está tomado de la Teoría del Centramiento[82], y se emplea en este caso para determinar el antecedente más probable de un pronombre cuando aquél se encuentra en una expresión anterior.

En las secciones siguientes se describirán detalladamente las herramientas que utiliza el algoritmo para llevar a cabo su función: las restricciones, las preferencias y la determinación del foco del discurso.

Posteriormente, se especificará formalmente el procedimiento en pseudocódigo¹².

¹¹ La representación de nombres propios y entidades en UNL se retomará posteriormente con más detalle, en la sección 4.4.2.

¹² En el Anexo II se ilustra el funcionamiento del algoritmo describiendo su ejecución paso a paso sobre un pequeño documento UNL de ejemplo

A) Restricciones

En general, se entenderá como *restricción* a una condición necesaria que debe satisfacer un nodo de una expresión para poder constituir el antecedente de una anáfora. El objetivo de las restricciones es reducir el espacio de búsqueda del antecedente, descartando los nodos candidatos que no verifiquen alguna de ellas.

Tal y como se adelantó en la descripción del algoritmo, las restricciones no se verifican sobre cualquier nodo en general, sino únicamente sobre aquellos que representan nombres propios de personas, y que además están ubicados en la misma expresión que contiene la anáfora (conjunto N).

Cuando un nodo que contiene un nombre propio no satisface una restricción, se está en realidad descartando al referente de ese nombre como antecedente de la anáfora. Debido a que en UNL todas las ocurrencias del mismo nombre propio son correferentes, deberán descartarse también aquéllos nodos que contienen ese mismo nombre, incluso aunque satisfagan todas las restricciones.

Esto significa que el proceso de filtrado debe realizarse en dos pasos: primero, se verifican las restricciones para todos los nodos del conjunto N , con el fin de obtener los nombres propios rechazados (denominado conjunto R) y en un segundo paso, se recorren nuevamente los elementos de N , esta vez para filtrar aquellos nodos que contengan nombres rechazados. Los nodos que no han sido filtrados forman el conjunto final de candidatos a antecedente de la anáfora, denominado conjunto C .

En los siguientes apartados se describirán las 4 restricciones que se utilizan en el algoritmo, que pueden clasificarse en morfológicas, sintácticas y semánticas:

Restricciones morfológicas:

- Concordancia de género y número

Restricciones sintácticas:

- Unicidad referencial en *scope*
- Filtro sintáctico

Restricciones semánticas:

- Compatibilidad semántica

1. Concordancia de género y número

Esta restricción establece que el antecedente debe poseer el mismo género y número que el pronombre. En el caso de la concordancia de número, y dado que se pretende resolver únicamente la anáfora de los pronombres personales “he” y “she”, se deberán descartar aquéllos nodos candidatos que posean el atributo “@pl” (plural). Sin embargo, todos los candidatos son nombres propios de persona, que necesariamente se encuentran en singular, por lo que la concordancia de número estará garantizada.

La concordancia de género es más problemática de verificar, ya que este rasgo morfológico no está contemplado como tal en UNL. No existen atributos que determinan el género de una determinada palabra universal, y tampoco puede deducirse a partir de sus restricciones. Se considera que el género es una propiedad dependiente de cada idioma y ha sido suprimido de las especificaciones del lenguaje.

A pesar de ello, existen determinadas palabras universales que poseen un género común en todos los idiomas, como son los nombres propios de personas y los pronombres personales “he(icl>person)” y “she(icl>person)” (estos últimos han conservado en UNL formas diferenciadas para el masculino y el femenino). Esto significa que el género de las anáforas siempre será conocido, aunque no así el de los nombres propios, a menos que se disponga de un almacén externo que contenga el género de cada nombre.

El procedimiento de resolución para UNL contempla la utilización de un recurso de este tipo, denominado tabla *G*, que asigna el género a cada nombre propio. No obstante, no se requiere que contenga toda la información necesaria inicialmente, es decir, la resolución de la anáfora podrá comenzar incluso si la tabla se encuentra totalmente vacía. Será durante la ejecución del algoritmo cuando se irán incorporando a la tabla nuevas entradas, a medida que se vaya infiriendo el género de los nombres.

Esta inferencia se realiza cuando se resuelve cada anáfora, utilizando la regla de la concordancia pero a posteriori, es decir, tras haber determinado el antecedente. Por ejemplo, si el algoritmo determina que el nombre “Rosa” es el antecedente de un pronombre “she”, se podrá deducir que “Rosa” es femenino, y esta información será añadida a la tabla *G*.

Como es obvio, este proceso de aprendizaje tiene el inconveniente de que, con una tabla *G* inicialmente vacía, no se podrá verificar la concordancia de género durante las primeras resoluciones del algoritmo, lo cual puede provocar errores. Además, la selección de un antecedente incorrecto puede ocasionalmente conducir a la deducción errónea del género de

un nombre, insertando en la tabla *G* una información inválida, propagando el fallo a todos los procesos de selección de candidatos en subsiguientes iteraciones del algoritmo.

En resumen, aunque el algoritmo está preparado para deducir el género de los nombres propios, es recomendable que la tabla *G* contenga información, si no de todos, al menos de los nombres propios más frecuentes del documento.

2. Unicidad referencial en *scope*

Su enunciado es el siguiente:

“Una anáfora pronominal y su antecedente no pueden ubicarse en el mismo scope dentro de una expresión UNL”

Esta condición es consecuencia directa de las reglas de codificación, que indican explícitamente que se podrán añadir pronombres personales a las expresiones únicamente si sus correspondientes referentes no están presentes en el mismo *scope*. En caso de que esto último fuera a suceder, sólo el referente deberá ser añadido, asumiendo las funciones que tuvieran ambos (pronombre y referente) en la oración original.

En la sección 4.2.1 ya pudo verse el efecto que tiene esta norma sobre la codificación de las anáforas intraoracionales, haciendo que tengan o no que ser representadas explícitamente en el interior de la expresión dependiendo de cómo se codifique ésta. En cuanto al procedimiento de resolución, la consecuencia importante de esta norma es que se podrán descartar como posibles antecedentes a los nombres propios que se sitúen en el mismo *scope* que el pronombre anafórico. Por ejemplo, en la expresión A de la Figura 38 se puede descartar a “Jesús” como posible antecedente del pronombre “he”, al situarse ambos en el mismo *scope*.

Figura 38. Diferentes codificaciones UNL de la misma oración

El poseedor de la carta deberá ser necesariamente una entidad externa. Sin embargo, en la expresión B no puede realizarse este descarte.

3. Filtro sintáctico

La idea de filtro sintáctico que se utilizará aquí está basada en los resultados que ofrece la Teoría de Ligamiento [32] para los lenguajes naturales. A grandes rasgos, esta teoría permite establecer restricciones a los antecedentes de las anáforas en función del posicionamiento de ambos en el árbol sintáctico de la oración.

Aunque por definición UNL no tiene como objetivo reflejar relaciones de tipo sintáctico, se verá posteriormente que existe cierto paralelismo entre la estructura de una expresión UNL y la sintaxis de la oración correspondiente en lenguaje natural. Este paralelismo puede ser aprovechado para descartar candidatos a antecedentes de anáforas pronominales.

La Teoría de Ligamiento establece sus principios utilizando la denominada relación *c-comando* entre los constituyentes de un árbol sintáctico. Su definición se basa parcialmente en la relación de *dominancia* (un nodo A domina a otro B si existe un camino descendente desde A hasta B en el árbol).

A partir de esta noción, se dice que un nodo A c-comanda a otro nodo B si y solo si:

- A no domina a B
- B no domina a A
- El primer nodo que domina a A también domina a B

La Figura 39 (siguiente página) muestra las diferentes relaciones c-comando entre los constituyentes de un árbol de ejemplo.

Los principios de la teoría se pueden enunciar de la siguiente forma:

- A. *Un pronombre reflexivo debe tener su antecedente en un contexto sintáctico cercano*¹³. Por ejemplo, en “John washed himself”, el pronombre “himself” tiene como antecedente al sustantivo “John”, ubicado en un contexto sintáctico cercano. Sin embargo, la frase “John asked Mary to wash himself” es errónea, ya que “himself” no puede tener como antecedente a “John”, por ubicarse en otra cláusula.

¹³ El *contexto sintáctico cercano* también se denomina *dominio de ligamiento*, y está delimitado por la cláusula que contiene al pronombre.

- B. *Un pronombre no puede ser correferente con un sintagma nominal que lo c-comanda, en un contexto sintáctico cercano.* Por ejemplo, en “John asked Mary to wash him” es posible que “John” sea en antecedente de “him”, ya que ambos se encuentran lo suficientemente lejos en la estructura sintáctica de la oración. Sin embargo, en “John washed him” sería incorrecto afirmar que “him” tiene a “John” como antecedente.
- C. *Un pronombre no puede ser correferente con un sintagma nominal no pronominal al que c-comanda.* Por ejemplo, en “He asked Mary to wash John” se puede afirmar que “John” no puede ser el antecedente de “he”.

Figura 39. Relaciones de c-comando en un árbol de ejemplo

Se puede observar un claro paralelismo entre la restricción sobre la unicidad referencial en *scope* definida en la sección anterior y el principio B de la Teoría de Ligamiento. La correspondencia entre ambos es evidente si se hace equivaler el concepto de “contexto sintáctico cercano” al concepto de *scope*, y pasando por alto la restricción sobre el posicionamiento sintáctico (relación c-comando), ya que las reglas de codificación de UNL aseguran la unicidad referencial del pronombre en todo el *scope* que lo contiene.

A pesar de ello, se ha optado por no incluir esa restricción como un tipo de filtro sintáctico por razones metodológicas, en cuanto a que la unicidad referencial en un *scope* puede verse más como una consecuencia de la interpretación de las reglas de codificación, y no como el resultado de adaptar el principio B de la Teoría de Ligamiento a UNL.

Además del principio B, contemplado por la restricción anterior, se tendrá que excluir del filtro sintáctico también el A, ya que se refiere a pronombres reflexivos, y éstos no son tratados por el algoritmo de resolución. En definitiva, el filtro sintáctico se deberá entonces elaborar exclusivamente a partir del principio C.

Como se ha comentado, a priori se podría pensar que no se puede trasladar directamente esta restricción a UNL, ya que la relación de c-comando está definida sobre árboles sintácticos y no sobre grafos dirigidos en general. Sin embargo, éste no es el problema más importante. La cuestión fundamental es que, a priori, no se puede asegurar que una condición como ésta, basada en propiedades sintácticas del lenguaje natural, tenga validez en UNL, entre otras cosas porque UNL no es un lenguaje natural sino un formalismo de representación de significados.

No obstante, a pesar de existir diferencias evidentes entre ellos, las expresiones de ambos lenguajes presentan cierto parecido estructural. La razón de ello es que UNL no fue concebido únicamente para representar significados, sino también para generar automáticamente textos a partir de sus expresiones, y esto requiere incorporar, de alguna forma, cierta información de tipo sintáctico.

Esta necesidad fue cubierta en el diseño de UNL haciendo que los tipos de relaciones representaran roles temáticos, que esencialmente son etiquetas abstractas capaces de determinar simultáneamente funciones sintácticas y semánticas del lenguaje. De esta forma, ambos niveles quedarían representados en las expresiones, lo que ha tenido como consecuencia un acercamiento significativo de la interlingua hacia el lenguaje natural, en particular a su estructura sintáctica, hasta el punto de producirse una correlación entre una expresión UNL y el árbol sintáctico de la oración que puede generar.

Esta correlación no significa que exista una biyección entre expresiones y árboles. A partir de una misma expresión se pueden construir varias oraciones diferentes, con árboles sintácticos diferentes, y también es posible codificar una misma oración de varias formas distintas. Sin embargo, es cierto que para cada par expresión-árbol puede establecerse una correspondencia a nivel de constituyente, de forma que a cada nodo de la expresión UNL se le puede asignar al menos un nodo del árbol sintáctico. Esta correspondencia, que se denominará función *I*, se puede definir formalmente de la siguiente manera:

$$I: \text{Nodos}(e_{UNL}) \rightarrow P(\text{Nodos}(AS(o)))$$

Dónde:

e_{UNL} es una expresión UNL

o es una oración que se genera a partir de e_{UNL}

$AS(x)$ es el árbol sintáctico de la oración x

$Nodos(x)$ es el conjunto de nodos de la estructura x (árbol o expresión)

$\mathcal{P}(x)$ es el conjunto de partes del conjunto x

Los criterios mediante los cuales la función I realiza las asignaciones están basados en reglas lingüísticas, por lo que esta función se encontrará íntimamente relacionada con el proceso de generación de UNL. Pero lo más importante de que exista una función de este tipo, en cuanto a la adaptación del principio C a UNL, es que permite anticipar la estructura de un árbol a partir de la estructura de la expresión, y esta capacidad de predicción es la que va permitir determinar las condiciones del filtro.

Dado que el principio C es una restricción semántica cuya premisa es una condición sintáctica, la aproximación que se va a seguir consistirá, en primer lugar, en determinar las estructuras UNL a partir de las cuales se generan árboles sintácticos que cumplen la premisa y, en segundo lugar, verificar que la conclusión semántica de la restricción es trasladable a UNL. A continuación se explicarán formalmente estos dos pasos.

Dada una oración en lenguaje natural o , puede reescribirse el principio C de la siguiente forma:

$$c\text{-comando}(p, sn) \Rightarrow \neg \text{correferentes}(p, sn) \quad (I)$$

Dónde:

p es un pronombre ($p \in AS(o)$)

sn es un sintagma nominal ($sn \in AS(o)$)

$c\text{-comando}(x, y)$: x c-comanda a y ($x, y \in Nodos(AS(o))$)

$\text{correferentes}(x, y)$: x e y son correferentes ($x, y \in Nodos(AS(o))$)

El primer paso consiste en determinar qué estructuras UNL producen árboles sintácticos con relaciones de c-comando. Es decir, siendo e_{UNL} una expresión UNL cualquiera, $\{o_i\}$ el conjunto de oraciones que se pueden generar a partir de ella e I_i las funciones $I_i: Nodos(e_{UNL}) \rightarrow P(Nodos(AS(o_i)))$, se debe encontrar una nueva relación en el dominio UNL, denominada $\alpha\text{-comando}$, tal que:

$$n_a, n_b \in \text{Nodos}(e_{UNL}) \wedge \alpha\text{-comando}(n_a, n_b) \Rightarrow \\ \forall i \forall a \in I_i(n_a) \forall b \in I_i(n_b), c\text{-comando}(a, b) \quad (II)$$

Según la implicación anterior, una relación α -comando entre dos nodos de una expresión UNL, deberá ser una condición suficiente para que exista una relación c-comando entre los nodos correspondientes del árbol sintáctico de cualquier oración generada a partir de esa expresión.

Como es obvio, deducir formalmente una relación con este requisito es extremadamente complejo y está más allá de los objetivos de esta tesis. En lugar de ello, se va a proporcionar una definición de α -comando confeccionada a partir de reglas intuitivas deducibles de la comparación entre árboles sintácticos y expresiones UNL. No se demostrará formalmente que cumple la propiedad (II), por lo que será una definición de carácter heurístico.

Con el fin de ilustrar la comparación entre ambas estructuras, se utilizará el ejemplo de la Figura 40 (página siguiente), que muestra el árbol sintáctico y la expresión UNL correspondientes a una misma oración.

La función I estaría definida de la siguiente forma:

Tabla 14. Definición de función $I(x)$

nodo UNL	$I(x)$
n₁	{b}
n₂	{a}
n₃	{c}
n₄	{d}
n₅	{f}
n₆	{e}
n₇	{g}

Y las relaciones de c-comando que se producen en el árbol sintáctico son las siguientes (solo se muestran aquéllas entre sintagmas nominales):

- a c-comanda a los nodos c, d, e y g
- c c-comanda al nodo d, e y g
- e c-comanda al nodo g

o: "He read the book of her because Mike bought it"

eUNL:

AS(o):

Figura 40. Correspondencias entre el árbol sintáctico y la expresión UNL

Según el requisito (II), la relación α -comando podrá relacionar como máximo los siguientes nodos de la expresión UNL:

- n_2 α -comanda a n_3, n_5, n_6 y n_7
- n_3 α -comanda a n_4, n_5, n_6 y n_7
- n_6 α -comanda al nodo n_7

Observando la disposición de los nodos en una y otra estructura, se pueden deducir algunas condiciones suficientes, en la expresión UNL, que garantizan una relación c-comando en el árbol sintáctico. Por ejemplo:

1. Una relación UNL entre dos nodos sustantivos da lugar a una relación de c-comando entre los correspondientes nodos del árbol. Es decir:

$$r(n_i, n_j) \Rightarrow \text{c-comando}(I(n_i), I(n_j))$$

Es el caso de la relación “pos” entre los nodos n_3 (“book”) y n_4 (“her”).

2. Si uno de los argumentos de un verbo es un *scope*, en el árbol sintáctico se producirá una relación c-comando entre cada uno de los argumentos del verbo restantes y cada uno de los nodos contenidos en el *scope*. Es decir:

$$r(v, s) \wedge \text{CAT}(f_{uw}(v)) = \text{VERBO} \wedge s \text{ es } \text{scope} \Rightarrow \\ \forall n \text{ no } \text{scope} \mid r(v, n) \quad \forall m \mid m \subset s, \text{c-comando}(I(n), I(m))$$

Donde:

$f_{uw}(n)$ indica la palabra universal que contiene el nodo n .

$\text{CAT}(w)$ indica la categoría gramatical de la palabra universal w

$m \subset s$ indica que el nodo m está incluido en el *scope* s o en alguno de sus descendientes

Es el caso del nodo n_2 (“he”) con los nodos n_5 (“buy”), n_6 (“Mike”) y n_7 (“it”).

En general, pueden deducirse más reglas si se conoce la forma en que se implementa la generación de oraciones en UNL. Sin embargo, esta tarea es compleja y en algunos casos

innecesaria, puesto que el objetivo del filtro es la eliminación de nodos candidatos a antecedente, y la restricción formulada en el apartado anterior (unicidad referencial en *scope*) ya filtra aquéllos que se ubican en el mismo *scope* que el pronombre.

Según esto, considerando el proceso de eliminación de candidatos en su totalidad, se concluye que el filtro sintáctico es igual de útil si se centra exclusivamente en los nodos ubicados en un *scope* diferente al de la anáfora, ya que del resto se encargará la restricción de unicidad referencial. Como la condición del filtro estará determinada por la relación α -comando, bastará como requisito que ésta cumpla la propiedad (II) sólo cuando n_i y n_j pertenezcan a diferentes *scopes*.

La definición de la relación que se va a proponer estará basada entonces en la segunda de las dos reglas mostradas anteriormente, relajando sus condiciones para adaptarse a un mayor número de casos. Para ello, se suprimirá el requerimiento de que el *scope* sea argumento directo del verbo, a cambio de que el nodo que contenga al verbo sea principal:

$$\alpha\text{-comando}(n_1, n_2) \Leftrightarrow \exists r, v \mid r(v, n_1) \wedge \text{CAT}(f_{uv}(v)) = \text{VERBO} \wedge \text{Scope}(n_2) \subset \text{Scope}(n_1) \wedge \\ v = n_p(\text{Scope}(v)) \Rightarrow \text{c-comando}(I(n_1), I(n_2))$$

Donde:

$\text{Scope}(n)$ es el *scope* que contiene al nodo n

$n_p(s)$ es el nodo principal del *scope* s

Es decir, n_1 α -comanda a n_2 si y solo si n_1 es un argumento del verbo principal de un *scope* s , y n_2 pertenece a cualquier *scope* contenido en s . Esta condición asegura que $I(n_1)$ se ubicará en el árbol sintáctico en un nivel superior a $I(n_2)$, dado que el argumento del nodo principal tendrá siempre una jerarquía sintáctica más alta que cualquier *scope* descendiente, puesto que estos se utilizan típicamente para representar cláusulas subordinadas.

Según esta definición, en el ejemplo de la Figura 40, se tendrán las siguientes relaciones α -comando:

- n_2 α -comanda a n_5 , n_6 y n_7
- n_3 α -comanda a n_5 , n_6 y n_7

Una vez determinada la condición suficiente para que se cumpla la premisa del principio C, el segundo objetivo es comprobar si puede asumirse su conclusión semántica, es decir, la no-correferencialidad entre los sintagmas.

Dado que UNL está diseñado para representar sin ambigüedades el significado de las oraciones, los conceptos denotados por los sintagmas de una oración deben coincidir con los denotados por las palabras universales correspondientes de la expresión UNL. Esto implica que la correferencia debe conservarse entre los constituyentes de ambas estructuras. Es decir, dada una oración o y una expresión e_{UNL} que codifica su significado, puede afirmarse lo siguiente:

$$a, b \in \text{Nodos}(AS(o)), a \in I(n_a), b \in I(n_b)$$

$$\text{correferentes}(a, b) \Leftrightarrow \text{correferentes}(n_a, n_b) \quad (\text{III})$$

Combinando las expresiones I, II y III, se obtiene la definición del filtro sintáctico para una expresión e_{UNL} :

$$\alpha\text{-comando}(n_p, n_{sn}) \Rightarrow \neg \text{correferentes}(n_p, n_{sn})$$

Donde:

n_p es un nodo de e_{UNL} que representa un pronombre “he(icl>person)” o “she(icl>person)”

n_{sn} es un nodo de e_{UNL} que representa un sustantivo

4. Consistencia semántica

Esta condición establece que las restricciones semánticas que satisfaga la anáfora deberán ser satisfechas también por su antecedente.

La única restricción de este tipo que contempla el algoritmo es la de asegurar que tanto la anáfora como el antecedente pertenecen a la misma clase semántica. Esta condición queda asegurada implícitamente durante la búsqueda de candidatos ya que, según la propia especificación del algoritmo, éstos deberán ser nombres propios de persona, y las anáforas que se resuelven son únicamente pronombres “he” o “she” referidos a personas.

B) Preferencias

Una vez descartados los nombres propios de la expresión que no cumplen alguna de las restricciones anteriores, se deberá elegir el antecedente de la anáfora entre el conjunto de los nombres restantes, denominado conjunto de candidatos C . Esta elección se llevará a cabo utilizando una serie de criterios denominados *preferencias*.

El procedimiento de resolución utiliza las preferencias para asignar una puntuación numérica entre 0 y 1 a cada uno de los candidatos, tras lo cual selecciona aquél que posea la puntuación más alta. En caso de que existan varios candidatos empatados con la puntuación máxima, se adopta una postura conservadora y se deja la anáfora sin resolver, con el fin de no mermar la precisión del algoritmo.

La puntuación de cada candidato se establece mediante la combinación de tres criterios o preferencias:

1. Prominencia funcional
2. Distancia al pronombre
3. Coincidencia con el foco

Cada una de ellas se evalúa también con un valor entre 0 y 1, de forma que la puntuación final (PCA) de un candidato c viene determinada por la suma ponderada de los tres valores:

$$PCA(c) = W_{\text{FUNCIÓN}} \cdot PCA_{\text{FUNCIÓN}}(c) + W_{\text{DISTANCIA}} \cdot PCA_{\text{DISTANCIA}}(c) + W_{\text{FOCO}} \cdot PCA_{\text{FOCO}}(c)$$

Donde los pares $\langle W_i, PCA_i \rangle$ representan el peso y la puntuación de cada preferencia respectivamente. Los pesos son valores fijos, y han sido determinados de forma experimental:

$$W_{\text{FUNCIÓN}} = 0.4 \quad W_{\text{DISTANCIA}} = 0.4 \quad W_{\text{FOCO}} = 0.2$$

A continuación se describe cada una de las preferencias:

1. Prominencia funcional

Este criterio establece un orden de preferencia entre nodos candidatos basado en la función sintáctica que desempeñan dentro de la expresión UNL. La función sintáctica de un candidato se calcula estudiando las relaciones que mantiene con sus nodos adyacentes.

El procedimiento de resolución considera un total de 4 *categorías funcionales* diferentes definidas exclusivamente para nodos sustantivos:

- Argumento verbal principal
- Argumento verbal secundario
- Origen de aposición
- Argumento nominal

A continuación se presentan por orden de peso:

- Argumento verbal principal. Un nodo sustantivo desempeña esta función cuando es destino de una relación proveniente de un verbo, y además se produce alguna de las siguientes condiciones:
 - a. El verbo es de tipo agentivo, es decir, tiene la restricción “(icl>do)”, y la relación es de tipo agt (agente).
 - b. El verbo es de tipo estativo o atributivo, es decir, tiene la restricción “(icl>be)”, y la relación es de tipo aoj (atributo).
 - c. El verbo designa un evento, es decir tiene la restricción “(icl>occur)”, y la relación es de tipo obj (objeto).

Según esto (y dependiendo del tipo de verbo) el argumento principal será el iniciador de una acción, la entidad a la que se atribuye un estado o atributo, o bien la entidad a la que afecta un evento, y generalmente representa el sujeto sintáctico en la oración correspondiente.

En los ejemplos de la Tabla 15 se han marcado en negrita los argumentos principales del verbo:

Tabla 15. Ejemplos de argumentos principales de verbos

Lenguaje natural	UNL
El sargento dio una orden a los soldados	agt(give(icl>do), sergeant)
	obj(give(icl>do), order)
	gol(give(icl>do), soldier)
El niño prefiere el balón	aoj(prefer(icl>be), child)
	obj(prefer(icl>be), ball)
El actor apareció tras el decorado	obj(appear(icl>occur), actor)
	man(appear(icl>occur), behind)
	obj(behind, set)

- Argumento verbal secundario. Un nodo sustantivo desempeña esta función cuando constituye cualquier argumento no principal de un verbo, ya sea de tipo directo o bien a través de una palabra universal adverbial.

En las siguientes oraciones se han marcado en negrita los argumentos secundarios del verbo:

Tabla 16. Ejemplos de argumentos secundarios de verbos

Lenguaje natural	UNL
El sargento dio una orden a los soldados	agt(give(icl>do),sergeant) obj(give(icl>do), order) gol(give(icl>do), soldier)
El niño prefiere el balón	aoj(prefer(icl>be), child) obj(prefer(icl>be), ball)
El actor apareció tras el decorado	obj(appear(icl>occur), actor) man(appear(icl>occur), behind) obj(behind, set)

- Origen de aposición. Un nodo sustantivo posee esta categoría funcional cuando es origen de una relación de tipo cnt (contenido). Esta relación se utiliza en UNL cuando se quiere explicar un concepto mediante la introducción de un nombre o añadiendo alguna otra característica.

El destino de dicha relación es justamente aquello que se explica o se introduce sobre el sustantivo, y típicamente se presenta en forma de aposición en la correspondiente oración en lenguaje natural.

En los siguientes ejemplos se marcan en negrita los nodos que pertenecen a esta categoría:

Tabla 17. Ejemplos de relaciones de aposición

Lenguaje natural	UNL
James, ingeniero industrial	cnt(James , engineer) mod(engineer, industrial)
NATO, North Atlantic Treaty Organization	cnt(NATO , organization) obj(organization, treaty) mod(treaty, atlantic) mod(atlantic, north)

- Argumento nominal. Un nodo sustantivo desempeña esta función cuando constituye un argumento de otro nodo sustantivo, bien sea directo o a través de una palabra universal adverbial, siempre que la relación no sea de tipo coordinativo (and, or, etc.).

Esta clase de construcciones generalmente acaban convirtiéndose en grupos preposicionales, como por ejemplo los siguientes:

Tabla 18. Ejemplos de argumentos nominales

Lenguaje natural	UNL
la tesis de Mike	pos(thesis, Mike)
el avión de Londres	plf(plane, London)
el coche junto al árbol	mod(car, close) obj(close, tree)

Si un mismo nodo satisface simultáneamente las condiciones de varias funciones, sólo se le asignará la primera de ellas, según el orden en que se han expuesto anteriormente.

Por ejemplo, en las siguientes expresiones los nodos marcados pueden desempeñar dos funciones a la vez. Sólo se considerará la que se ha marcado en la tercera columna, teniendo en cuenta el orden de preferencia establecido:

Tabla 19. Ejemplo de nodos con dos funciones

Lenguaje natural	UNL	Funciones que puede desempeñar el nodo marcado
Los relatos de Poe, escritor americano	pos(tale, Poe) cnt(Poe , writer) mod(writer, american)	Origen de aposición Argumento nominal
Le devolvió su chaqueta	agt(return(icl>do),he:01) obj(return(icl>do), jacket) gol(return(icl>do), he:02) pos(jacket, he:02)	Argumento verbal secundario Argumento nominal

La puntuación que recibe un candidato para esta preferencia (valor $PCA_{\text{FUNCIÓN}}$) dependerá de la categoría funcional a la que pertenezca. La siguiente tabla muestra los valores que son asignados en cada caso:

Tabla 20. Puntuación del candidato según su categoría funcional

Categoría funcional	$PCA_{\text{FUNCIÓN}}(c)$
Argumento verbal principal	1
Argumento verbal secundario	0,7
Origen de aposición	0,5
Argumento nominal	0,3
Ninguna (no cumple ninguna de las condiciones)	0

2. Distancia al pronombre

Este criterio favorece a los candidatos que se encuentran más cerca de la anáfora dentro de la expresión UNL. La cercanía se mide mediante de la distancia *lingüística* entre ambos nodos, definida en el apartado 4.1.3.

La puntuación que asigna esta preferencia está determinada por la siguiente fórmula:

$$PCA_{\text{DISTANCIA}}(c) = (5 - \text{Distancia}(c, \text{pronombre})) / 4 \quad \text{Si Distancia}(c, \text{pronombre}) < 5$$

$$PCA_{\text{DISTANCIA}}(c) = 0 \quad \text{Si Distancia}(c, \text{pronombre}) \geq 5$$

3. Coincidencia con el foco

Esta preferencia tendrá una puntuación PCA_{FOCO} de 1 si el candidato coincide con el foco del discurso que exista en ese momento. De lo contrario, la puntuación será 0.

C) Foco del discurso

La elección del foco del discurso se basará en el criterio de selección basado en la Teoría de Centramiento (*Centering Theory*) [82]. Esta técnica permite resolver la ambigüedad en aquellos casos en los que la elección del antecedente entre un conjunto de candidatos no queda clara, decantándose por aquél que coincida con la entidad que reciba la atención en ese momento del discurso. Esta entidad, denominada foco o centro (*center*), es calculada en cada una de las expresiones, postulándose como el antecedente más probable de una anáfora determinada, siguiendo los principios de la cohesión del discurso.

Dado que en UNL no es posible codificar el nivel pragmático del lenguaje, los mecanismos que rigen la cohesión del discurso permanecen invariables en la interlingua, lo que permitiría adaptar cualquiera de estas técnicas e incorporarla al procedimiento de resolución para UNL. Sin embargo, no se ha realizado ninguna adaptación concreta, sino que se ha utilizado una técnica personalizada y muy simplificada con respecto a los modelos convencionales.

Lo que se mantiene en común es el hecho de que, a lo largo del procesamiento del documento, se mantiene una variable con el denominado centro o foco del discurso. La diferencia está en que este foco está circunscrito exclusivamente a nombres propios de persona, sin considerar

otro tipo de entidades. Como es obvio, esto está motivado por el hecho de que el objetivo del algoritmo es la resolución de pronombres referidos a personas.

Por tanto, en lo que respecta al procedimiento para UNL, el foco se podría definir como el nombre de la persona que recibe la atención o que protagoniza lo que es relatado o descrito en una expresión concreta de un documento. Evidentemente, y atendiendo a su sentido tradicional, el centro del discurso no tiene por qué coincidir solo con personas a lo largo del documento, por lo que en caso de que ningún nombre pueda ser considerado como foco, se le asignará el valor nulo.

El algoritmo utiliza el foco principalmente para resolver anáforas interoracionales, situación que se produce cuando el conjunto de candidatos a antecedente C es vacío. De hecho, el foco constituye la única herramienta para resolver esta clase de anáforas. El procedimiento es el siguiente: si no existe ningún candidato válido a antecedente ubicado en la misma expresión que la anáfora, bien porque la expresión no contenga nombres propios de persona, bien porque todos los nombres existentes han sido descartados por no cumplir las restricciones, se utilizará el foco (si existe) como único candidato. Si satisface todas las restricciones (es decir, no pertenece al conjunto R y concuerda en género con el pronombre), será elegido automáticamente como el antecedente.

Aparte de este uso, el foco también interviene durante la selección del antecedente de las anáforas intraoracionales (cuando C no es vacío). Como se vio en la sección anterior, la coincidencia de un candidato con el foco del discurso le otorga una puntuación PC_{FOCO} adicional.

Ciclo de vida del foco

El foco es inicializado con el valor nulo al comienzo de la ejecución del algoritmo, y es revisado cada vez que concluye el procesamiento de una expresión, justo después de resolver las anáforas que pudiera contener, en su caso.

Para determinar el nuevo foco, se consideran como candidatos los nodos de la expresión que contienen nombres propios de persona, y que además poseen cualquiera de las categorías funcionales siguientes (utilizando para ello el mismo criterio seguido para la preferencia de “prominencia funcional”):

- Argumento verbal principal
- Argumento verbal secundario
- Origen de aposición

Debido a que este cálculo se realiza tras la resolución de los pronombres de la expresión, hay que tener en cuenta que el conjunto de nodos con nombres propios (conjunto N) estará formado, en ese momento, por los que contuviera inicialmente la expresión más aquéllos que hubieran resultado de sustituir cada anáfora por su correspondiente antecedente.

Una vez que se ha obtenido el conjunto de nodos candidatos a foco (denominado conjunto H), se elige uno de ellos, siguiendo un procedimiento análogo al utilizado para la selección de los antecedentes. En primer lugar, se asigna a cada candidato una puntuación numérica (PCF) entre 0 y 1, calculada a partir de la media aritmética de las puntuaciones correspondientes a dos preferencias: prominencia funcional y distancia con respecto al nodo entrada de la expresión.

La fórmula que determina la puntuación de un candidato c es la siguiente:

$$PCF(c) = (PF_{FUNCION}(c) + PF_{DISTANCIA}(c))/2$$

Donde:

$PF_{FUNCION}(c)$ es un valor que depende de la categoría funcional del nodo, y se asigna utilizando los mismos criterios especificados en la Tabla 20.

$PF_{DISTANCIA}(c)$ es un valor inversamente proporcional a la distancia del candidato con el nodo entrada de la expresión. Este criterio se basa en la idea de que cuanto más cerca se ubique el candidato con respecto al nodo entrada, más cerca de la raíz se situará el sintagma que le corresponde dentro del árbol sintáctico de la oración, siguiendo las reglas de generación de UNL. El valor de la puntuación se obtiene de la siguiente forma:

$$PF_{DISTANCIA}(c) = 1/Distancia(c, n_p(e)) + 1$$

Una vez calculados los valores PCF de todos los nodos candidatos, se elige como nuevo foco el nombre propio correspondiente a aquél que posea el valor más alto. Como en el caso de la selección de antecedentes, si existen varios candidatos (con nombres diferentes) empatados con la puntuación máxima, se descartan todos ellos, produciéndose el mismo efecto que si no existiera ninguno ($H = \text{vacío}$).

En este último caso, es decir, cuando no hay candidatos, se conservará el foco que existiera previamente, con la restricción de que este mantenimiento no podrá llevarse a cabo durante más de tres expresiones consecutivas. Dicho de otro modo, si durante tres revisiones de foco seguidas el conjunto H es vacío, aquél se pierde, recibiendo el valor nulo. Este mecanismo trata de evitar pérdidas abruptas de foco, protegiendo su continuidad cuando aparezcan expresiones aisladas que no contengan referencias a nombres propios de persona.

En resumen, dependiendo del foco que hubiera en la expresión anterior (f_{i-1}), el mejor candidato a foco calculado en la expresión actual (c_i) y el valor asignado al nuevo foco (f_i), cada revisión de foco podrá resultar en una de las cinco transiciones mostradas en la Tabla 21.

Tabla 21. Transiciones posibles del foco

Tipo de transición de foco	Descripción	Condición	Valor del nuevo foco (f_i)
Confirmación	El mejor candidato coincide con el foco de la expresión anterior, o bien no existe ninguno de ellos (ambos tienen el valor nulo).	$c_i = f_{i-1}$	c_i
Sin foco	No hay candidatos, y tampoco existe foco anterior.	$c_i = \text{nulo} \wedge f_{i-1} = \text{nulo}$	nulo
Cambio	El mejor candidato existe (no es nulo), y es diferente al foco anterior.	$c_i \neq \text{nulo} \wedge c_i \neq f_{i-1}$	c_i
Mantenimiento	No hay candidatos a foco, el foco anterior es distinto de nulo y las dos transiciones anteriores no fueron mantenimientos de foco.	$c_i = \text{nulo} \wedge f_i \neq \text{nulo} \wedge \text{manten.consec} < 3$	f_{i-1}
Pérdida	No hay candidatos a foco, el foco anterior es distinto de nulo y las dos transiciones anteriores fueron mantenimientos de foco.	$c_i = \text{nulo} \wedge f_i \neq \text{nulo} \wedge \text{manten.consec} = 3$	nulo

Especificación del algoritmo

PROCEDIMIENTO: RESOLVER_ANAFORA

ENTRADA:

$D = e_1, e_2, \dots$ documento UNL, formado por las expresiones e_i

SALIDA:

D' = documento D tras haber sustituido los pronombres personales he y she en las expresiones por sus antecedentes

COMIENZO

```

1. foco := null
2. vigencia_foco := 0
3. G("he(icl>person)") := M
4. G("she(icl>person)") := F
5. PARA CADA  $e_i$  ( $i = 1, 2, \dots$ ) de  $D$ :
    {
    5.1. Obtener el conjunto  $N = \{n_j\}$  de nodos de  $e_i$ , tal que  $\forall j, f_{uw}(n_j)$  es
        nombre propio de persona.
    5.2. Construir la lista  $P = \langle p_j \rangle$  de nodos de  $e_i$ , tal que  $\forall j$ :
        a.  $f_{uw}(p_j) = \text{"he(icl>person)"} \vee f_{uw}(p_j) = \text{"she(icl>person)"}
        b.  $\text{Distancia}(p_j, n_p(e_i)) \leq \text{Distancia}(p_{j+1}, n_p(e_i))$ 
    5.3. PARA CADA  $p_j$  ( $j=1, 2, \dots$ ) de  $P$ :
        {
        i. Obtener el conjunto  $R = \{f_{uw}(n_k)\}$  de nombres, tal que  $\forall k, n_k \in N$  y
            además cumple alguna de las condiciones siguientes:
            a.  $\neg \text{GenerosCompatibles}(f_{uw}(n_k), f_{uw}(p_j))$ 
            b.  $\text{Scope}(n_k) = \text{Scope}(p_j)$ 
            c.  $p_j \alpha\text{-comanda a } n_k$ 
        ii. Construir el conjunto de nodos  $C = [31]$  tal que  $\forall k, c_k \in N \wedge f_{uw}(c_k) \notin R$ 
        iii. Construir el conjunto de nombres  $A = \{f_{uw}(c_i)\}$  tal que  $\forall i, c_i \in C$  y
            además:
                 $\text{PCA}(c_i) = \max\{\text{PCA}(c), c \in C\}$ 
        iv. antecedente := null
        v. SI  $|A| = 1$  ENTONCES
                antecedente :=  $a \in A$ 
                SI NO SI  $(A = \emptyset) \wedge (\text{foco} \neq \text{null}) \wedge (\text{foco} \notin R) \wedge$ 
                 $\text{GenerosCompatibles}(\text{foco}, f_{uw}(p_j))$  ENTONCES
                        antecedente := foco
        vi. SI antecedente  $\neq$  null ENTONCES
                G(antecedente) := G( $f_{uw}(p_j)$ )
                 $f_{uw}(p_j)$  := antecedente
                Añadir  $p_j$  a  $N$ 
        }
    5.4. Construir el conjunto de nodos  $H = \{h_i\}$ , tal que  $\forall i, h_i \in N$  y
        además  $\text{FunciónNodo}(h_i) \in \{\text{arg\_principal\_verbo}, \text{arg\_secundario\_verbo},$ 
         $\text{origen\_aposición}\}$ 
    5.5. Construir el conjunto de nombres  $F = \{f_{uw}(h_i)\}$ , tal que  $\forall i, h_i \in H$ 
        y además:$ 
```

```

PCF( $h_i$ ) =  $\max\{PCF(h), h \in H\}$ 

5.6.      SI ( $F = \emptyset$ )  $\vee$  ( $|F| > 1$ )      ENTONCES
          permanencia_foco := permanencia_foco - 1
          SI permanencia_foco  $\leq 0$  ENTONCES
              foco := null
          SI NO
              foco :=  $f \in F$ 
              permanencia_foco := 3
      }
FIN

```

Funciones auxiliares:

- *GénerosCompatibles*(uw_1, uw_2): retorna FALSO si las palabras universales uw_1 y uw_2 tienen distinto género según la tabla G , es decir, $G(uw_1) \neq G(uw_2)$. Retorna CIERTO en caso de que tengan el mismo género, o bien no exista en G información sobre alguna de las dos palabras universales.
- *Distancia*(n_1, n_2): retorna la distancia entre los nodos n_1 y n_2 , según se definió en la sección 4.1.4.
- *Scope*(n): retorna el *scope* que contiene al nodo n .
- *FunciónNodo*(n): retorna la categoría funcional a la que pertenece el nodo n . Las categorías posibles son: *arg_principal_verbo*, *arg_secundario_verbo*, *org_aposición*, *arg_nominal* u otra.
- *PCA*(n): retorna la puntuación del nodo candidato a antecedente n .
- *PCF*(n): retorna la puntuación del nodo candidato a foco n .

4.3 Preparación de las expresiones

Una vez aplicada la resolución de la anáfora pronominal, y antes de comenzar la identificación de los elementos de la ontología, es necesario realizar una preparación previa de las expresiones UNL. El objetivo es facilitar el análisis de las estructuras lingüísticas, simplificando por un lado las expresiones, y por otro incorporando relaciones deducibles entre palabras universales, según la estructura de los grafos.

Las operaciones que se aplicarán en esta etapa a cada expresión UNL son las siguientes:

1. Incorporación de relaciones deducibles
2. Reducción de anidamiento
3. Eliminación de *stopwords*

4.3.1 Incorporación relaciones deducibles

Existen determinados fenómenos lingüísticos en lenguaje natural cuya representación en UNL puede conducir a la deducción de relaciones no incluidas explícitamente en el grafo. Es el caso de las oraciones coordinadas (frases unidas por conjunciones “y”, “o”, “pero”, etc.).

En la sección 4.1.3 ya se pudo ver cómo se representa este tipo de fenómeno en UNL. En esencia, cuando todos los elementos coordinados de una oración se relacionan de la misma forma con un tercer elemento, la estructura conjunta se puede representar de forma simplificada, especificando solo las relaciones para el primero de los elementos coordinados. Se entiende que el resto de ellos “heredan” estas relaciones por formar parte de la cadena de coordinación. La Figura 41 ilustra este hecho.

Se observa que los elementos que forma la cadena de coordinación están unidos por el nexo “and”, cuyo sentido se establece, de acuerdo a las especificaciones de UNL, de forma inversa al orden de lectura de la oración original.

De igual forma que en lenguaje natural resulta redundante decir “compró el monitor, compró el teclado y compró el ratón óptico”, en la representación correspondiente en UNL solo es necesario especificar que “buy” tiene como objeto al nodo “mouse”. El vínculo “and” entre los elementos de la cadena permite deducir que “keyboard” y “monitor” son también objetos del verbo, sin tener que añadir explícitamente las correspondientes relaciones “obj” al grafo.

“Compró el monitor, el teclado y el ratón óptico”

Figura 41. Ejemplo de coordinación lingüística (caso 1)

Es importante observar que esta herencia de relaciones no afecta a la relación “mod” que une “mouse” con “optical”. La razón de ello es puramente arbitraria. Según las normas de UNL, el esquema de representación anterior solo supone la propagación de las relaciones entrantes al primer nodo de la coordinación.

Existe otro esquema de representación que supone la propagación tanto de las relaciones entrantes como salientes. La siguiente figura muestra un ejemplo:

“El robot abrirá o bien destruirá por error la puerta”

Figura 42. Ejemplo de coordinación lingüística (caso 2)

En este caso, los elementos coordinados aparecen encerrados en un *scope* (uno de ellos es su nodo principal), y están unidos por un nexo compuesto, formado por la relación “or” y la palabra universal auxiliar “either_or”. Esta es la manera de representar que “robot” actúa como agente y “door” como objeto de ambos verbos (“open” y “destroy”). En este esquema de representación, se indica que tanto las relaciones entrantes como salientes al *scope* se aplican a todos los nodos coordinados. Mientras, las relaciones que mantengan estos nodos con otros dentro del *scope* no se transmiten.

Con el fin de facilitar el reconocimiento de las estructuras ontológicas que se realizará en etapas posteriores, es necesario incorporar estas relaciones deducibles mediante la modificación apropiada de las expresiones UNL. Esta transformación dependerá del esquema de representación de la coordinación que siga la expresión. La Figura 43 y la Figura 44 esquematizan la transformación que se realizará según el caso.

Figura 43. Transformación de la coordinación lingüística (caso I)

Figura 44. Transformación de la coordinación lingüística (caso II)

Se observa que los nexos desaparecen, incorporándose las relaciones r y s deducidas para todos los nodos de la cadena. En el segundo esquema, además, la transformación supone la desaparición de un *scope*. Al desaparecer el nexo, la expresión transformada pierde información lingüística. Con el fin de simplificar el método, se asumirá esta pérdida de información.

Aplicando los esquemas anteriores a los ejemplos de la Figura 41 y la Figura 42, se obtendrán las siguientes expresiones modificadas:

Figura 45. Expresiones transformadas

4.3.2 Reducción de anidamiento

La aparición de *scopes* en las expresiones UNL puede dificultar el reconocimiento de determinadas estructuras lingüísticas en su interior, si se utiliza como herramienta el concepto de patrón definido en la sección 4.1.5. Un patrón lingüístico es una estructura no anidada, cuyo encaje solo es posible cuando los elementos con los que se corresponde se encuentran en el mismo nivel de anidamiento.

Los *scopes* en las expresiones UNL introducen fronteras que pueden reducir el número de instanciaciones posibles de un patrón en una expresión. Por ejemplo, si se buscan las instanciaciones del patrón básico agente – verbo – objeto en la oración de la Figura 46, solamente se obtiene un encaje posible, correspondiente a la oración principal “el jugador levantó la copa”¹⁴.

La instancia “el jugador ganó el partido”, que forma parte de la oración subordinada, no encaja con el patrón debido a que el agente se ubica en un *scope* diferente al verbo y al objeto.

¹⁴ Con el fin de simplificar la figura, se han omitido tanto los atributos como las restricciones de las palabras universales de la expresión UNL.

Figura 46. Patrón instanciado únicamente en un scope

Este problema se puede resolver si se reduce el anidamiento de la expresión a un solo nivel, reescribiéndola sin utilizar *scopes*:

Figura 47. Instanciación de patrón lingüístico tras reducir el anidamiento

donde se observa que ya es posible obtener ambas instanciaciones.

Dado que la identificación de los componentes ontológicos se realizará mediante patrones UNL, será necesario reducir el anidamiento de las expresiones lo máximo posible. Según las especificaciones de UNL, el uso de *scopes* en las expresiones no está restringido, por lo que su presencia es relativamente alta. En general, su uso obedece a uno (o varios) de los siguientes motivos:

1. Simplificar la codificación de la coordinación, tal y como se ha visto en el apartado anterior.
2. Indicar que un determinado rasgo lingüístico afecta a una parte de la expresión UNL. Por ejemplo, para indicar que parte de una oración está entrecomillada, se introduce la subexpresión correspondiente en un *scope* y se le asigna el atributo “@doble_quote”.
3. Hacer más legibles las expresiones UNL complejas, por ejemplo delimitando en *scopes* las cláusulas subordinadas (como el caso de la Figura 46). De esta forma, la estructura de expresión UNL se acerca a la estructura sintáctica de la oración.
4. Representar nombres de entidades mediante una subexpresión. Ciertos tipos de nombres propios y títulos asignados a objetos individuales pueden aparecer representados mediante una subexpresión UNL encerrada en un *scope*¹⁵. En el caso de los nombres propios, se asigna al *scope* el atributo “@concept”, mientras que los *scopes* correspondientes a títulos se relacionan mediante la relación “nam” con la palabra universal que designa el objeto al que se asigna dicho título.

Se aplicará la reducción de anidamiento en todos los casos menos en el caso 4, debido a que la presencia del *scope* será necesaria para identificar las entidades, como se verá en la sección correspondiente. Mientras que el caso nº 1 ya ha sido tratado al transformar la representación de la coordinación lingüística, para los casos 2 y 3 se eliminarán los *scopes* tal y como se ha hecho en la expresión de ejemplo de la Figura 46, es decir: añadir al nodo entrada del *scope* todas las relaciones entrantes y salientes a dicho *scope* en el nivel superior de anidamiento. El esquema general de transformación se muestra en la Figura 48:

¹⁵ El la sección 4.3.2 se ve con más detalle la representación de entidades en UNL.

Figura 48. Esquema general de la operación de reducción de anidamiento

La desventaja de esta forma sencilla de eliminar los *scopes* (tanto con esta operación como en el caso de la coordinación lingüística), es que la expresión UNL resultante puede quedar invalidada para la generación de la oración en lenguaje natural, debido a lo siguiente:

1. La expresión resultante puede violar la restricción de la unicidad referencial en *scope*¹⁶: si en una expresión existen varias palabras universales con el mismo referente en distintos *scopes*, tras aplicar la operación quedarán todas en un único *scope*, violando la restricción.
2. La expresión puede perder información lingüística, si el *scope* eliminado tenía asignado un atributo.

La unidad referencial es un requisito cuya violación solo podría perjudicar, en principio, al proceso de generación de oraciones a partir de los grafos UNL, por lo que no influirá de cara a obtener la información ontológica de las expresiones.

La pérdida de atributos sí que podría tener influencia. Algunos codifican información semántica relevante, como puede ser la negación (“@not”) o el tiempo de las acciones y eventos. A pesar de ello, en este trabajo se simplificará el análisis asumiendo esta pérdida de información.

¹⁶ Ver epígrafe sobre la unidad referencial en *scope* en la resolución de la anáfora pronominal (sección 4.2.2)

4.3.3 Eliminación de *stopwords*

La última operación que se realizará en las expresiones consistirá en eliminar ciertas palabras universales que no resultarán útiles en el análisis posterior, como las que representan determinantes, pronombres (que no han sido no resueltos en la etapa anterior) y palabras con significado superfluo (*stopwords*), cuyo contenido semántico se considerará prescindible para obtener los conceptos y relaciones de la ontología:

- this(icl>adj)
- that(icl>adj)
- each(icl>thing)
- what(icl>thing)
- there(icl>location>thing, ant>here)
- much(icl>thing)
- other(icl>thing)
- anything(icl>thing)
- it(icl>thing)
- one(icl>unit>thing)

4.4 Construcción de la ontología

Una vez preparadas las expresiones UNL, se procederá a identificar las estructuras lingüísticas que permitirán deducir los componentes de la ontología. El método está focalizado en la determinación de **conceptos, instancias y relaciones de tipo taxonómico**.

La estrategia general consistirá en determinar primero los conceptos a partir de los términos y entidades reconocibles en las expresiones UNL. Se verá que la representación en interlingua de ambos elementos sigue ciertos patrones de formación específicos, que permiten identificarlos con facilidad.

Tras esta primera etapa de reconocimiento, se realizará un filtrado posterior de términos, seleccionando los más relevantes. Debido a la ausencia de corpus UNL de tamaño suficiente, en este método no se utilizarán las técnicas estadísticas típicas que se suelen emplear para medir la relevancia. Como se verá en la sección correspondiente, se determinará que un término es relevante si aparece en alguna expresión “cerca” de alguna entidad.

Después de realizar el filtrado de términos, se habrán obtenido tanto los conceptos como las instancias de la ontología. El siguiente paso consistirá en deducir las relaciones de tipo taxonómico (“es un”) que se producen entre ellos, siguiendo tres métodos:

- **Inspección de entidades:** se verá que la representación de las entidades en UNL permite obtener el concepto al que instancian mediante simple inspección.
- **Descomposición de términos:** un término compuesto por varias palabras puede descomponerse en otros que representan conceptos más generales. Este procedimiento dará lugar a la incorporación de nuevos conceptos a la ontología.
- Observando **estructuras de aposición** entre dos términos en las expresiones UNL (por ejemplo: “El árbol de levas, componente del motor, controla el funcionamiento de las válvulas”).

Las siguientes secciones están dedicadas a la explicación detallada de estas etapas.

4.4.1 Reconocimiento de términos

Los términos son palabras o secuencias de palabras que denotan conceptos de forma unívoca. Pueden aparecer en forma de sustantivos, adjetivos o verbos, aunque generalmente se suele considerar al sustantivo como la unidad terminológica más frecuente.

Cuando un término está formado por varias palabras, aparece en las oraciones como un sintagma nominal, cuya estructura se rige según ciertos esquemas o patrones variables de un idioma a otro. Los sistemas de extracción de información se valen de estos patrones para identificar los términos dentro de las oraciones.

La siguiente tabla muestra varios ejemplos de términos en español e inglés junto a sus patrones de formación. La palabra que actúa como núcleo del término aparece subrayada:

Tabla 22. Ejemplos de términos en español e inglés

Español		Inglés	
Término	Patrón de formación	Término	Patrón de formación
“ <u>comida</u> adulterada”	<u>Sust.</u> + Adj.	“Adulterd <u>food</u> ”	Adj. + <u>Sust.</u>
“ <u>administrador</u> de datos del sistema”	<u>Sust.</u> + Prep. + Sust. + Prep. + Det. + Sust.	“data system <u>administrator</u> ”	Sust. + Sust. + <u>Sust.</u>
“ <u>relato</u> corto de terror”	<u>Sust.</u> + Adj. + Prep. + Sust.	“short horror <u>story</u> ”	Adj. + Sust. + <u>Sust.</u>

El proceso de reconocimiento de los términos en un texto utiliza un conjunto de patrones, del estilo de los que aparecen en la tabla, con el fin de detectar las secuencias de palabras que pueden constituir un término. Debido a que estos patrones están especificados según las categorías de las palabras, el proceso requiere necesariamente de una etapa de análisis morfológico.

La noción de término en lenguaje natural tiene también su correspondencia en UNL. En lugar de secuencias de palabras, aparecerán en forma de palabras universales conectadas mediante

relaciones. No se provee ninguna regla especial para codificarlos, ni mecanismos para indicar que una determinada subexpresión es un término. El codificador enlazará las palabras universales que lo forman siguiendo las mismas reglas que para cualquier otro sintagma nominal.

A continuación se muestra la codificación en UNL de los términos de la tabla anterior:

Figura 49. Ejemplos de términos en UNL

A simple vista, se pueden apreciar ciertas características comunes, que se derivan de las normas de codificación en UNL, y que serán extrapolables a la codificación de cualquier otro término:

- El grafo correspondiente tiene forma de árbol (sin considerar el sentido de las flechas), en el que el nodo raíz es el núcleo del término, que es siempre un sustantivo.

- Las palabras que forman el término son sustantivos, adjetivos y participios, estos últimos codificados como palabras universales verbales¹⁷.
- Las preposiciones y determinantes que conectan las palabras en lenguaje natural son sustituidos mediante relaciones y atributos UNL.
- El tipo de las relaciones que se establecen entre las palabras universales dependerá de su categoría y significado, junto al tipo de dependencia que debe codificar la relación:
 - La relación entre un sustantivo y un adjetivo siempre es de tipo “mod”.
 - La relación entre un sustantivo y un participio dependerá de la categoría del verbo:
 - Si es un verbo *estativo* (icl>be), entonces la relación será “aoj”
 - En otro caso, es decir, si es un verbo de tipo *agentivo* (icl>do) o bien indica un *evento* (icl>occur), entonces la relación será “obj”
 - La relación entre dos sustantivos refleja una dependencia argumental o circunstancial del sustantivo destino con respecto al origen. Generalmente, esta dependencia puede expresarse utilizando alguno de los tipos de relaciones UNL existentes, aunque hay casos en los que utiliza una palabra universal adicional que actúa como preposición.

Tras inspeccionar diversos corpus UNL, se ha determinado que las relaciones que se producen entre los sustantivos de un término son expresables mediante un conjunto reducido de relaciones, que se denominará en adelante R_T , y que está formado por los siguientes tipos: “agt”, “aoj”, “ins”, “met”, “mod”, “obj” y “src”.

Con la información anterior, y del mismo modo que se hizo en la Tabla 22 para los ejemplos en español e inglés, se pueden especificar los patrones UNL de formación para cada uno de los términos anteriores, expresados mediante categorías de palabras universales y relaciones. La tabla siguiente muestra estos patrones, junto ejemplos de términos cuya codificación sigue estos mismos esquemas.

¹⁷ En UNL no se ha determinado aún con claridad si los participios deben ser codificados mediante una palabra universal adjetivo o bien con un verbo, como en el ejemplo anterior. En este trabajo se asumirá que pueden existir ambos tipos de codificación.

Tabla 23. Ejemplos de correspondencias entre patrones de formación de términos

Términos en inglés	Patrón lingüístico	Términos de español
<p>Adj. + Sust.</p> <p>“integrated <u>platform</u>”</p> <p>“adulterated <u>food</u>”</p>		<p>Sust. + Adj.</p> <p>“<u>plataforma</u> integrada”</p> <p>“<u>alimento</u> adulterado”</p>
<p>Sust. + Sust. + Sust.</p> <p>“earth sensor <u>structure</u>”</p> <p>“data system <u>administrator</u>”</p> <p>Sust. + Prep + Sust. + Sust.</p> <p>“<u>structure</u> of earth sensor”</p> <p>“<u>administrator</u> of data system”</p> <p>Sust. + Prep + Det + Sust. + Sust.</p> <p>“<u>structure</u> of the earth sensor”</p>		<p>Sust + Prep + Det + Sust + Prep + Sust</p> <p>“<u>estructura</u> del sensor de tierra”</p> <p>“<u>administrador</u> del sistema de datos”</p> <p>Sust + Prep + Sust + Prep + Sust</p> <p>“modulación por modelo de pulsos”</p>
<p>Adj. + Sust. + Sust.</p> <p>“short horror story”</p> <p>Adj. + Sust. + “of” + Sust.</p> <p>“short story of horror”</p>		<p>Sust. + Adj. + Prep + Sust.</p> <p>“Relato corto de terror”</p> <p>Sust. + Adj. Prep + Det + Sust.</p> <p>“Llave inglesa para las ruedas”</p>

Como se observa, en UNL se produce una reducción de la paráfrasis, provocando que a cada patrón lingüístico le puedan corresponder varios patrones en lenguaje natural. Esta circunstancia, junto a otras características específicas de UNL, aportan ventajas a la hora de identificar los términos en UNL. Estas ventajas se pueden resumir en los siguientes puntos:

- La **identificación del patrón** en la expresión es más sencilla que en el caso del lenguaje natural, debido a que las categorías de las palabras universales están especificadas explícitamente en sus restricciones, por lo que no es necesario realizar ningún análisis sintáctico ni morfológico.
- El **núcleo del término** es identificable de forma inmediata, como el nodo raíz del sintagma.
- Las palabras universales no presentan **variantes morfológicas**. Por ejemplo, “short horror stories” se codifica con las mismas palabras universales que “short horror story”, salvo que en el primer caso se añadiría el atributo “@pl” a “story” para indicar el plural.
- En UNL se reducen las **variantes ortográficas** de los términos, debido a que no existe la sinonimia en el diccionario de palabras universales, y a que la construcción de cada una de ellas obedece a unas reglas estrictas de formación. Por ejemplo, las palabras “database”, “data-base” y “data base” se corresponden con una única palabra universal “database(icl>information>thing)”.
- Como se ha visto, en UNL también se reducen las **variantes sintácticas**. Por ejemplo, los términos “*energy distribution*” y “*distribution of energy*” son codificados de la misma forma en UNL.

Como se vio en la sección dedicada al modelo formal, cada patrón lingüístico define intrínsecamente una operación de encaje que hace corresponder los nodos y etiquetas del patrón con los nodos y etiquetas en las expresiones UNL, conservando las adyacencias. El método propuesto utilizará un conjunto finito de patrones UNL como herramientas para localizar los términos.

Los patrones de la Tabla 23 reflejan tres estructuras típicas que pueden presentar los términos en UNL, sin embargo, todavía no se pueden utilizar como patrones de forma efectiva, debido a que no reflejan las condiciones de contorno dentro de la expresión. Esta circunstancia queda ilustrada en la siguiente figura, en la que se ha rodeado con trazo discontinuo la instanciación del patrón Sustantivo - Participio en tres expresiones de ejemplo:

Figura 50. Instanciaciones posibles del patrón N + V

De las tres instanciaciones posibles, solamente la que se produce en la oración C se puede considerar como válida. El problema en el caso A es que la instanciación contiene un verbo ("drive") que no está actuando como participio, mientras que en el caso B se ha instanciado con parte del término ("componentes distribuidos") en lugar del término completo ("sistema de componentes distribuidos"). Estos problemas se deben a que el patrón lingüístico no especifica cuál debe ser la frontera del término una vez trasladado el patrón a la expresión.

Para resolver este problema, los patrones deberán redefinirse siguiendo los siguiente criterios:

- Los nodos del patrón instanciado (exceptuando al núcleo) no podrán poseer más relaciones que las indicadas en el patrón.

Si se recuerda lo explicado en la sección 4.1.5, este tipo de condiciones pueden especificarse mediante las restricciones de adyacencia $r_i^{\rightarrow} = r_i^{\leftarrow} = TR_{UNL}$.

En el caso del ejemplo, estas restricciones se aplicarán al nodo correspondiente al participio:

Figura 51. Patrón con restricciones de adyacencia

El patrón redefinido de esta forma no podrá encajar en una expresión UNL si el verbo se relaciona con cualquier otro nodo que no sea el núcleo. El encaje del caso A, por tanto, ya no sería posible.

- El núcleo del patrón solo deberá encajar con los núcleos de los sintagmas nominales máximos (no incluidos en otros) de la expresión UNL.

Los sintagmas nominales máximos actúan como participantes en las predicciones, por lo que sus núcleos podrán aparecer:

- Actuando como argumentos de palabras universales verbales que no actúan como participio, es decir, que tienen más de un argumento.
- Como origen y destino de la relación UNL “cnt”, que codifica la aposición.

La siguiente figura muestra un ejemplo de expresión UNL en el que se han marcado los núcleos de los sintagmas de tamaño máximo:

“El compresor de gases, otro componente del sistema, funcionó mal”

Figura 52. Núcleos de sintagmas nominales máximos

“*compressor*” es núcleo del sintagma máximo “compresor de gases”, por el doble motivo de ser argumento del verbo “*work*” y origen de la relación de aposición “*cnt*”. Por otro lado, “*component*” es el núcleo del sintagma máximo “otro componente del sistema”, por ser destino de la relación de aposición.

Previamente a la aplicación de los patrones en las expresiones UNL, estos núcleos deberán ser localizados y marcados con el atributo especial “@ncl”. Este atributo es auxiliar, y no forma parte de las especificaciones de UNL. Su utilización permitirá restringir los sintagmas nominales sobre los que podrán encajar los patrones UNL, especificando que su nodo raíz requiere dicho atributo:

Figura 53. Patrón lingüístico con restricción de atributo auxiliar de núcleo

Con esta nueva condición ya se evitaría, por ejemplo, el encaje en el caso B de la Figura 50, debido a que “*component*” no es núcleo de un sintagma nominal máximo (sí lo sería “*system*”, por ser argumento del verbo “*work*”).

- El núcleo del patrón instanciado podrá tener, en la expresión UNL, otras relaciones diferentes a las indicadas en el patrón, pero éstas no deberán formar parte de un término más grande.

El patrón definido de la forma anterior puede presentar aun el problema de encajar sobre una parte de un término, en lugar del término completo. Esto es debido a que no tiene especificada ninguna restricción de adyacencia para el núcleo. La siguiente figura ilustra esta situación:

Figura 54. Instanciación de patrón en término mayor

Se observa que el núcleo del patrón ha encajado con el núcleo del sintagma máximo “material desechable reciclado”, pero este sintagma tiene un tamaño superior al definido por el patrón. La instanciación no debería haberse producido por este motivo.

Es importante observar que el nodo correspondiente al núcleo de un término actúa como un constituyente funcional, representando al término completo dentro de la expresión UNL. A través de este nodo, el término se relacionará con el resto de elementos de la expresión (en la figura, se ve que actúa como objeto de la acción “use”), por lo que no se podrán limitar sus condiciones de adyacencia de la misma forma a como se ha hecho con el resto de nodos del patrón.

Las condiciones de contorno del término en el núcleo deberán permitir la posibilidad de que se relacione otros términos y predicaciones de la expresión UNL, pero deberán evitar relaciones que supongan formar un término mayor.

Para conseguir este doble objetivo, se especificarán para el núcleo las restricciones de adyacencia siguientes:

1. Se permitirá cualquier relación entrante: $r_i^{\rightarrow} = \emptyset$.
2. Solo se permitirán las relaciones salientes que no puedan formar parte de un término. Se considerará que las relaciones entre las palabras universales de un término están restringidas a los tipos mencionados con anterioridad, es decir:
 - De tipo “mod” entre un sustantivo y un adjetivo
 - De tipo “obj” o “aoj” entre un participio y un sustantivo
 - De un tipo perteneciente a R_T entre dos sustantivos

Como el conjunto R_T contiene también a los tipos “mod”, “obj” y “aoj”, la restricción de adyacencia se formulará como: $r_i^{\leftarrow} = R_T$.

El patrón lingüístico quedará finalmente especificado de la siguiente forma:

Figura 55. Patrón lingüístico completamente especificado

A continuación se enumeran los patrones UNL para el reconocimiento de términos que se utilizará en el método, junto a un ejemplo:

Tabla 24. Patrones lingüísticos para reconocer términos

Patrones	Ejemplo
<p>Sust</p>	<p><i>“oficina”</i></p> <p>office(icl>place_of_business>thing)</p>
<p>Sust + Adj</p>	<p><i>“bosquejo biográfico”</i></p> <p>sketch(icl>summary>thing)</p> <p>mod</p> <p>biographical(icl>adj, equ>biographic, com>biography)</p>
<p>Sust + Sust</p>	<p><i>“profesor de historia”</i></p> <p>professor(icl>academician>thing)</p> <p>mod</p> <p>history(icl>humanistic_discipline>thing)</p>
<p>Sust + Sust + Sust</p>	<p><i>“distribución en vértice de la velocidad”</i></p> <p>distribution(icl>spacing>thing, ant>concentration)</p> <p>obj</p> <p>velocity(icl>rate>thing, equ>speed)</p> <p>plc</p> <p>vortex(icl>current>thing, equ>whirlpool)</p>
<p>Sust + Adj + Adj</p>	<p><i>“telemetría analógica condicional”</i></p> <p>telemetry(icl>measurement>thing)</p> <p>mod</p> <p>analogic(icl>adj, ant>digital)</p> <p>mod</p> <p>conditional(icl>adj)</p>

<p style="text-align: center;">Sust + Adj + Sust</p>	<p style="text-align: center;"><i>“escuela secundaria del condado”</i></p>
<p style="text-align: center;">Sust + Sust + Sust</p>	<p style="text-align: center;"><i>“proceso de filtración de agua”</i></p>
<p style="text-align: center;">Sust + Sust + Adj + Adj</p>	<p style="text-align: center;"><i>“profesor de ingeniería mecánica civil”</i></p>
<p style="text-align: center;">Sust + Adj + Sust + Adj</p>	<p style="text-align: center;"><i>“teoría lineal del sistema hidrológico”</i></p>

4.4.2 Reconocimiento de entidades

Las entidades (*named entities*) representan instancias de conceptos. Generalmente, se consideran como tal a los nombres propios de persona, de organizaciones, geográficos, números, fechas, acrónimos, figuras, etc. Su codificación en UNL es uno de los aspectos que no fueron bien definidos en las especificaciones iniciales de la interlingua. Las últimas normas de codificación, que se encuentran en fase de desarrollo en el momento de escribir esta tesis, establecen varias formas posibles de representarlas. En el método propuesto solo se considerarán aquellas codificaciones que permiten deducir la clase a la que pertenecen las entidades:

1. Entidades representadas mediante una única palabra universal de tipo instancia
2. Subexpresiones que representan fechas
3. Nombres o títulos asignados a objetos
4. Subexpresiones que representan nombres propios

La clase deducida formará también parte de los conceptos de la ontología.

A) Palabra universal instancia

Existe un tipo especial de palabra universal que representa unívocamente a una entidad o instancia concreta. Esta circunstancia se indica en la propia palabra mediante una restricción semántica de tipo “iof” (*instance of*) hacia una palabra que representa la clase a la que pertenece.

En UNL se emplean este tipo de palabras para hacer referencia a entidades de los siguientes tipos:

- **Nombres propios** de personas:

1. John(icl>person>thing)

- **Topónimos:**

2. Australia(iof>country>thing)
3. Mississipi(iof>river>thing)
4. Bonn(iof>city>thing)
5. Urals(iof>mountain_range>thing)

Acrónimos:

6. NATO(iof>organization>thing)

El patrón para reconocer este tipo de entidades es sencillo y se muestra en la tabla siguiente. El concepto que se instancia se deduce inmediatamente a partir de la restricción.

Tabla 25. Patrones para obtener instancias

Patrón	Ejemplo
<div> <div>N (iof>*)</div> </div>	<div> <div>"Australia"</div> <div>Australia(iof>country>thing)</div> </div>

B) Fechas

Las fechas se codifican en UNL siguiendo esquemas fijos. A continuación se muestran los patrones que se utilizarán para localizarlas en las expresiones:

Tabla 26. Patrones para obtener fechas

Patrón	Ejemplo
<div> <div> <div> <div> <div>"day" @ncl</div> <div>nam</div> <div>N</div> </div> <div> <div>tim</div> </div> <div> <div>"month"</div> <div>nam</div> <div>N</div> </div> <div> <div>tim</div> </div> <div> <div>"year"</div> <div>nam</div> <div>N</div> </div> </div> </div> </div>	<div> <div>"3 de Abril de 1990"</div> <div> <div> <div>day(icl>time_period>thing)</div> <div>mod</div> <div>3</div> </div> <div> <div>tim</div> </div> <div> <div> <div>month(icl>time_period>thing)</div> <div>mod</div> <div>April(icl>Gregorian_calendar_month>thing)</div> </div> <div> <div>tim</div> </div> <div> <div> <div>year(icl>time_period>thing)</div> <div>mod</div> <div>1990</div> </div> </div> </div> </div> </div>
<div> <div> <div> <div> <div>"month" @ncl</div> <div>nam</div> <div>N</div> </div> <div> <div>tim</div> </div> <div> <div>"year"</div> <div>nam</div> <div>N</div> </div> </div> </div> </div>	<div> <div>"Mayo de 2013"</div> <div> <div> <div>month(icl>time_period>thing)</div> <div>mod</div> <div>May(icl>Gregorian_calendar_month>thing)</div> </div> <div> <div>tim</div> </div> <div> <div> <div>year(icl>time_period>thing)</div> <div>mod</div> <div>2013</div> </div> </div> </div> </div>

C) Nombres o títulos asignados a objetos

En una expresión UNL se pueden asignar nombres o títulos con la relación "nam". El título puede estar representado por una sola palabra universal temporal entrecomillada, o bien por una subexpresión UNL encerrada en un *scope* con el atributo "@double_quote". El concepto que se instancia está referenciado por el término origen de la relación "nam".

El patrón para la obtención de este tipo de entidades es el siguiente:

Tabla 27. Patrón para obtener nombres o títulos

Patrón	Ejemplo
	<p><i>"el partido político 'Podemos'"</i></p>

D) Subexpresiones que representan nombres propios

En UNL es posible representar nombres propios mediante una subexpresión UNL encerrada en un scope con el atributo “@concept”. El concepto que instancia estará representado por el nodo principal de dicho scope.

El patrón para obtener este tipo de entidades es el siguiente:

Tabla 28. Patrón para obtener nombres propios en scopes

Patrón	Ejemplo
<div>N @concept</div>	<p><i>“Organización Mundial de la Salud”</i></p>

4.4.3 Selección de conceptos

Una vez reconocidas las entidades y los términos de las expresiones, el siguiente paso es determinar cuáles de ellos formarán parte de la ontología, siguiendo el criterio de la relevancia con respecto al dominio de las fuentes.

Como se ha mencionado anteriormente, el método no empleará ningún tipo de procedimiento estadístico para determinar la relevancia de un concepto con respecto al dominio. En lugar de ello se aplicarán los siguientes criterios:

- Las entidades identificadas, junto a los conceptos que instancian, se considerarán todas relevantes del dominio.
- Un término se considerará relevante si aparece en alguna expresión UNL participando en alguna predicación lingüística junto a una entidad.

En UNL, las predicaciones están representadas por palabras universales verbales, o bien por algunos tipos de relaciones UNL. Un término entonces será seleccionado si en alguna expresión aparece como un argumento de un verbo junto a una entidad, o bien aparece relacionado con una entidad directamente. Estas condiciones se pueden representar mediante los siguientes patrones lingüísticos:

Figura 56. Patrones lingüísticos para selección de términos

Solamente los términos que instancien alguno de los patrones anteriores será seleccionado.

4.4.4 Relaciones taxonómicas

Las relaciones taxonómicas entre los conceptos se deducirán de las siguientes formas:

1. **Entre una entidad y el concepto** al que instancia se produce una relación de tipo taxonómico “instancia de”.
2. A partir de los términos formados por 2 o más palabras universales se pueden obtener conceptos más generales, **descomponiendo el término en otros más pequeños**.

La descomposición se realiza teniendo en cuenta que la representación del término tiene forma de árbol. Los términos más generales se obtendrán eliminando los niveles inferiores del árbol sucesivamente, desde las hojas hasta la raíz, obteniendo en cada paso términos más generales.

La siguiente figura muestra el proceso de descomposición:

Figura 57. Ejemplo de descomposición de un término

Los nuevos conceptos obtenidos mediante este procedimiento estarán relacionados mediante la relación taxonómica “es un”, y serán incorporados a la ontología.

3. **La relación de aposición “cnt” entre dos conceptos** codifica una relación de tipo taxonómico, especificando que el origen es un concepto más específico que el destino. Un ejemplo puede verse en la Figura 52, en el que se deduce que “compresor de gas” es un “componente del sistema” (la palabra universal “other” se habrá filtrado como *stopword* en la etapa de preparación de expresiones).

4.5 Ejemplo paso a paso

Con el fin de ilustrar el método, en esta sección se mostrará su funcionamiento paso a paso sobre una expresión UNL de ejemplo, con una cantidad importante de información de tipo factual. La expresión es la siguiente:

“Christian Gordon, historiador y oficial del ejército, nació en 1962 en Pittsburg, Pensilvania, y publicó la novela histórica ‘Sangre guerrera’”

Figura 58. Ejemplo de expresión UNL

A continuación se detalla cada etapa del método:

Resolución de anáfora:

La expresión no contiene pronombres personales, por lo que no se aplica esta etapa.

Preparación de la expresión (incorporación relaciones deducibles):

En la expresión se representan dos coordinaciones: una entre los verbos “publish” y “be_born”, y otra entre “officer” e “historian”. En la primera no se utiliza ninguno de los dos esquemas que pueden dar lugar a la deducción de relaciones, por lo que solo se eliminará el nexa (la relación “and”).

La segunda coordinación sigue el segundo esquema de representación, así que deberá aplicarse la transformación correspondiente, eliminando el *scope* “:01” y añadiendo las relaciones de tipo “cnt” desde “Christian_Gordon” a “officer” e “historian”.

Preparación de la expresión (reducción de anidamiento):

Existe otro *scope* aparte del que representaba la coordinación (“:02”), pero se utiliza para representar un título, ya que posee la relación entrante “nam”, por lo que no se aplicará ninguna transformación.

Preparación de la expresión (eliminación de stopwords):

La expresión no contiene *stopwords*.

En este punto, la expresión ha quedado como sigue:

Figura 59. Expresión tras aplicar el preprocesamiento

Identificación de términos y entidades:

Aplicando los patrones definidos en las secciones 4.4.1 y 4.4.2, se obtendrán los siguientes elementos¹⁸:

1. Términos:

- “novela histórica”
- “oficial militar”
- “historiador”

2. Entidades:

- “Christian Gordon”, como instancia de “persona”
- “Pittsburg”, como instancia de “ciudad”
- “Pennsilvania”, como instancia de “Estado americano”
- “Sangre guerrera”, como título de “novela histórica”
- “Año 1954”, como fecha.

En la figura siguiente se muestran los diferentes elementos identificados:

Figura 60. Identificación de términos y entidades

¹⁸ Por motivos de claridad, se mencionarán los términos y entidades en lenguaje natural, en lugar de utilizar su correspondiente representación en UNL.

Selección de conceptos:

Serán seleccionadas todas las entidades que han sido identificadas, además de los términos que se relacionan con alguna de ellas, bien directamente, bien a través de una palabra universal verbal.

El término “Novela histórica” está relacionado con “Christian Gordon” (entidad) mediante un verbo (“publish”), por lo que será seleccionado. De igual forma, “oficial militar” e “historiador” están relacionados directamente con “Christian Gordon”, por lo que también serán seleccionados.

Relaciones taxonómicas:

Las relaciones tipo “es un” que pueden deducirse entre las entidades y las clases a las que instancian son:

- Entre “sangre guerrera” y “novela histórica”
- Entre “Christian Gordon” y “persona”
- Entre “Pennsilvania” y “Estado americano”
- Entre “Pittsburg” y “ciudad”
- Entre “Año 1962” y “fecha”

Según la relación de aposición, se establecen las siguientes relaciones:

- Entre “oficial militar” y “Christian Gordon”
- Entre “historiador” y “Christian Gordon”

Y por composición de términos:

- Entre “oficial militar” y “oficial”
- Entre “novela histórica” y “novela”

La combinación de todos los elementos anteriores (conceptos, instancias y relaciones) dan como resultado la siguiente ontología, expresada en interlingua:

Figura 61. Ontología obtenida de la expresión

EXPERIMENTACIÓN

Se ha llevado a cabo la experimentación del método sobre un corpus formado por un artículo perteneciente a la colección EOLSS [100]. Específicamente, se ha seleccionado el artículo “*Biographies of eminent water resources personalities*”, consistente en una recopilación de biografías de personalidades históricas relacionadas con recursos hídricos.

El artículo está formado por 395 expresiones UNL, con un total de 2534 palabras universales¹⁹, y contiene una cantidad importante de información de tipo factual (nombres propios de personas, organizaciones, fechas, etc.), lo que lo hacen apropiado para la aplicación del método.

Para realizar la experimentación, ha sido necesario desarrollar un sistema software que implementa el método propuesto. A pesar de que existen diversas herramientas públicas de programación para el manejo de estructuras de grafos, ninguna de ellas se adaptaba a los requerimientos del modelo formal expuesto en la sección 4.1, por lo que se ha desarrollado una librería específica en C++ a tal efecto. Dicha librería está formada por 28 clases, e implementa todas las operaciones definidas en el modelo: definición y encaje de patrones, distancia lingüística entre nodos, extracción de subgrafos, creación y edición de expresiones UNL, etc.

¹⁹ Las palabras universales utilizadas pertenecen al diccionario accesible en <http://pacifico.dia.fi.upm.es:8080/dicweb/>

Las funcionalidades implementadas sirven de base al resto de módulos del sistema, que se organizan según la siguiente arquitectura:

Figura 62. Arquitectura del sistema implementado

- **Módulo de carga:** se encarga de analizar la colección de documentos UNL en formato texto que recibe como entrada y cargarlos en memoria con la estructura adecuada.
- **Módulo de resolución de anáfora:** implementa el algoritmo de resolución de anáfora pronominal explicado en la sección 5.3, sustituyendo las ocurrencias de los pronombres anafóricos “he(icl>person)” y “she(icl>person)” en las expresiones por sus respectivos referentes.

- **Módulo de preparación de expresiones:** prepara las expresiones UNL para la extracción de conceptos que se realizará en la etapa posterior. Este módulo está formado por dos submódulos:
 - **Módulo de procesamiento de la coordinación:** es el encargado de reconocer los esquemas de representación de la coordinación descritos en el apartado 4.3.1, incorporando a las expresiones las relaciones que se puedan deducir de acuerdo a dichos esquemas.
 - **Módulo de simplificación de expresiones:** implementa el algoritmo de reducción de anidamiento de las expresiones UNL y la eliminación de *stopwords*, descritos en las secciones 4.3.2 y 4.3.3.
- **Módulo de construcción de la ontología:** se encarga de construir la ontología a partir de las expresiones UNL preparadas por el módulo anterior. Está formado por 4 submódulos:
 - **Módulo de reconocimiento de conceptos:** utilizando una biblioteca de patrones lingüísticos, este módulo identifica los conceptos y entidades en las expresiones (ver secciones 4.4.1 y 4.4.2).
 - **Módulo de filtrado de conceptos:** se encarga del filtrado de los términos reconocidos por el módulo anterior, seleccionando aquellos que participan en alguna predicación junto a alguna entidad (ver sección 0).
 - **Módulo de obtención de relaciones taxonómicas:** deduce las relaciones taxonómicas entre los conceptos, siguiendo las técnicas descritas en la sección 4.4.4)
 - **Módulo de síntesis:** compone la ontología a partir de los conceptos, entidades y relaciones obtenidas por los módulos anteriores.

Los resultados de la aplicación del sistema al corpus se describen a continuación, etapa a etapa:

Resolución de anáfora

En el corpus de experimentación aparecen un total de 275 pronombres personales “he(icl>person)”, repartidos en 206 expresiones, y un total de 168 nombres propios de persona. La siguiente tabla muestra las estadísticas iniciales del corpus (antes de ser procesado por el sistema):

Tabla 29. Estadísticas del corpus relacionadas con la anáfora pronominal

Expresiones con al menos un pronombre	206 (52%)
Expresiones con al menos un nombre propio de persona	143 (36%)
Expresiones con al menos un pronombre y un nombre propio de persona	36 (9,11%)
Expresiones sin pronombres ni nombres propios de persona	82 (20,7%)
Nº de nombres propios de persona	168
Nº personas únicas referenciadas	65
Número total de pronombres	275

Con el objetivo de validar el algoritmo, antes de aplicarlo se ha procedido a determinar manualmente los antecedentes de cada uno de los pronombres del corpus, anotando tanto sus coordenadas como el nombre propio al que se refieren en un fichero.

El algoritmo de resolución se ha implementado de tal forma que compara sus resultados con los anotados en ese fichero, obteniendo los resultados de la validación inmediatamente. Son los siguientes:

Tabla 30. Resultados de la resolución de la anáfora

Pronombres resueltos correctamente	261 (94,90 %)
Pronombres resueltos incorrectamente (se les asigna un nombre equivocado)	4 (1,45 %)
Pronombres no resueltos (no se les asigna ningún nombre)	10 (3,63 %)
Número total de pronombres	275

Se observa que la precisión es muy elevada ($\approx 95\%$), y el número de errores, pequeño ($\approx 1,5\%$).

Tanto los pronombres resueltos erróneamente como los no resueltos han sido anáforas interoracionales. Según la especificación del algoritmo, estas anáforas se resuelven con el foco del discurso vigente en ese momento. En los 4 casos de resolución incorrecta, el foco seleccionado por el algoritmo era erróneo, mientras que en los 10 en los que no se produjo

ninguna resolución, no existía foco, debido a que transcurrieron más de 3 expresiones consecutivas desde que fuera establecido o confirmado el último foco.

Preparación de expresiones

La etapa de preparación de expresiones supone transformarlas en función de los *scopes* y estructuras de coordinación que contengan, así como la presencia de *stopwords*. La siguiente tabla muestra la distribución de estos fenómenos en el corpus, previamente a la aplicación del procedimiento:

Tabla 31. Estadísticas del corpus relacionadas con la preparación de expresiones

Expresiones con al menos un <i>scope</i>	272
Número total de <i>scopes</i>	765
Número medio de <i>scopes</i> por expresión	≈2
Nº medio de palabras universales por expresión	≈15
Número de expresiones con al menos una estructura de coordinación lingüística	189
Número total de estructuras de coordinación lingüística en el corpus	214
Número de stopwords	291

La tabla siguiente muestra el recuento de las operaciones realizadas por el sistema durante esta fase:

Tabla 32. Resultados de la preparación de expresiones

Estructuras de coordinación transformadas	157 (73%)
Relaciones añadidas por transformación de la coordinación	316
Expresiones a las que se ha aplicado la reducción de anidamiento	248
Número de <i>scopes</i> eliminados por reducción de anidamiento	689 (90%)
Número de <i>stopwords</i> eliminadas	291

Se observa que no todas las estructuras de coordinación han sido transformadas (el 73%). Las que no lo han sido, eran estructuras que seguían alguno de los esquemas que permiten la deducción de relaciones, tal y como se definió en la sección 4.3.2. A pesar de ello, el número de relaciones incorporadas (316) es elevado.

Por otra parte, tampoco se han eliminado todos los *scopes* en las expresiones mediante la reducción de anidamiento (el 90%). Los *scopes* que no han sido eliminados con este procedimiento fueron utilizados para representar o bien nombres de entidades (por tanto no reducibles), o bien eran estructuras de coordinación que fueron tratadas antes por el proceso correspondiente (el tratamiento de la coordinación es anterior a la reducción de anidamiento).

Identificación y selección de términos y entidades

Los módulos de reconocimiento de términos y entidades utilizarán los patrones definidos en las secciones 4.4.1 y 4.4.2 para llevar a cabo la identificación. Estos patrones se encuentran almacenados en un archivo de texto utilizando un lenguaje especial diseñado a tal efecto²⁰.

Tras aplicar los patrones de identificación, se obtienen los siguientes resultados:

Tabla 33. Resultados de la identificación de términos y entidades

Número de términos	501
Número de entidades	373
- Fechas	120 (32,1%)
- Palabras universales de instancia	207 (54,5%)
- Nombres propios o títulos, representados mediante la relación “nam”	46 (12,33%)
- Nombres propios representados mediante subexpresiones UNL (<i>scopes</i> con atributo “@concept”)	0

²⁰ En el anexo Anexo III se describe este lenguaje, mientras que en el anexo Anexo IV se puede ver parte de los patrones especificados en texto.

Donde se observa la alta presencia de entidades (373), de las cuales ninguna está representada por *scopes* con el atributo “@concept”. Esto último es debido a que esta forma de representar entidades fue incorporada a las especificaciones de UNL con posterioridad a la elaboración del corpus de experimentación.

Una vez realizada la identificación, se continúa con el filtrado de los términos, seleccionando los que sitúan cerca de alguna entidad en al menos una expresión. El número de términos seleccionados finalmente es 131, distribuidos de la siguiente forma (según el número de palabras):

Tabla 34. Términos seleccionados

Términos de 1 palabra	63
Términos de 2 palabras	50
Términos de 3 palabras	16
Términos de 4 palabras	1
Total	130

Relaciones taxonómicas y composición de la ontología

Aparte de las relaciones taxonómicas que se producen entre las entidades detectadas y el concepto correspondiente al que instancian, el módulo de obtención de relaciones identifica 77 nuevas relaciones taxonómicas aplicando el procedimiento de composición de términos, implicando la incorporación de otros tantos conceptos a la ontología. Por otra parte, el mismo módulo detecta otras 38 relaciones taxonómicas, identificando las estructuras de aposición expresadas mediante la relación “cnt”.

Todos estos elementos son finalmente combinados por el módulo de síntesis, que se encarga de descartar aquellos componentes que hayan sido detectados más de una vez por diferentes procesos (por ejemplo, una misma relación taxonómica puede haber sido detectada mediante composición de términos y por identificación de la aposición). El resultado constituye la ontología final, cuya composición se muestra en la Tabla 35.

Tabla 35. Recuento de componentes de la ontología final

Número de clases	288
Número de relaciones taxonómicas entre clases	50
Número de entidades	373

En la Figura 63 (página siguiente) se muestra un pequeño fragmento de la ontología final. En el Anexo IV se recoge parte del fichero de salida generado por el sistema.

Figura 63. Fragmento de la ontología obtenida en la experimentación

6.1 Conclusiones

En esta tesis se ha presentado un método que constituye una nueva aproximación a la construcción automática de ontologías, en el que se utiliza como fuente una representación en interlingua del significado de una colección de textos. El alcance del método se ha limitado a la obtención de conceptos, relaciones taxonómicas e instancias.

Se ha utilizado como interlingua a UNL (Universal Networking Language), por ser la única de propósito general que se encuentra normalizada. Esta interlingua, basada en el formalismo de las redes semánticas, ha permitido centrar el análisis en la identificación de patrones lingüísticos en el nivel semántico.

Como se ha podido comprobar en este trabajo, el reconocimiento de patrones en las expresiones UNL ofrece ventajas con respecto a otros métodos de *Ontology Learning* basados en el estudio del lenguaje natural:

- Las expresiones UNL representan el significado de forma no ambigua, codificando los fenómenos lingüísticos necesarios para su correcta interpretación. No se requiere, por tanto, ningún tipo de análisis morfológico o sintáctico previo. Únicamente se necesita una etapa de preprocesamiento orientada a transformar convenientemente las expresiones, con el fin de facilitar la identificación de los patrones. Esta transformación es sencilla, y para realizarla solo se necesita inspeccionar la estructura de las expresiones.
- En UNL se reduce la paráfrasis con respecto a los lenguajes naturales, de forma que a un mismo grafo UNL le pueden corresponder diferentes frases con el mismo significado. Esto resulta especialmente útil para la identificación de los conceptos de

la ontología, permitiendo obtener la representación “canónica” de un concepto frente a sus diferentes variantes ortográficas (“data-base”, “database” y “data base”), morfológicas (“short story” y “short stories”) y sintácticas (“energy distribution” y “distribution of energy”).

- Los términos en UNL siguen patrones de formación del estilo de los que se producen en lenguaje natural, con la ventaja de ser independientes de la lengua. Estos patrones presentan una estructura en forma de árbol, en el que la raíz constituye el núcleo del término. Su descomposición por niveles conduce a la identificación de nuevos términos progresivamente más generales, permitiendo incorporar a la ontología las relaciones jerárquicas entre ellos.
- El estándar UNL define relaciones que por sí mismas determinan relaciones ontológicas, como es el caso de la relación de aposición, cuya identificación permite deducir relaciones taxonómicas entre conceptos. Este tipo de relación es difícil de identificar en los lenguajes naturales.
- Las representación de las entidades en UNL permite tanto su identificación inmediata como la deducción de la clase a la que instancian.

Por otra parte, y como novedad en el área de *Ontology Learning*, se ha incorporado al método un procedimiento para resolver, en el plano de la interlingua, el fenómeno de la anáfora pronominal de los pronombres personales de tercera persona. Este procedimiento se ha basado en técnicas conocidas aplicadas al lenguaje natural, obteniendo una gran precisión (95%). El objetivo es aumentar la presencia de entidades en las expresiones, sustituyendo los pronombres anafóricos por sus antecedentes.

La presencia de entidades es fundamental para el éxito del método, debido a que la selección de los conceptos relevantes se realiza en función de si el concepto aparece o no instanciado en el texto, o bien de si está ubicado cerca de alguna entidad en alguna expresión. La carencia de corpus UNL de tamaño adecuado ha impedido utilizar técnicas estadísticas clásicas para la determinación de la relevancia, siendo la proximidad con entidades el único criterio empleado, lo cual ha constituido una limitación.

A pesar que el método tiene margen para mejorar (como se verá en la sección siguiente), resulta una aproximación prometedora que permite abordar la construcción de ontologías de forma independiente de la lengua, haciendo uso de un modelo, el de las interlinguas, que ha sido obviado durante mucho tiempo y que guarda una relación directa con los modelos ontológicos, en cuanto a que ambos se definen como una representación de la semántica.

6.2 Líneas futuras

Debido a que no existen trabajos anteriores en el área de *Ontology Learning* que utilicen el modelo de la interlingua, existen muchas vías de investigación abiertas. En general, se trata de aplicar técnicas basadas en modelos conocidos aplicados sobre el lenguaje natural. A continuación se enumeran las líneas de mejora más inmediatas:

- El **algoritmo de resolución de la anáfora** que se ha presentado está limitado a al caso de los pronombres anafóricos de tercera persona “he” y “she”, cuando el antecedente es un nombre propio. El procedimiento puede ser extendido a otros tipos de anáforas de más difícil resolución, como las anáforas de pronombres “they” e “it”, o la **catáfora**, donde el antecedente se ubica con posterioridad al pronombre.
- Como generalización de la anáfora y catáfora, se puede mejorar la determinación de las relaciones entre las entidades estudiando la **correferencia** en UNL. Se dice que dos expresiones (generalmente sintagmas nominales) son correferentes si ambos aluden a la misma entidad del mundo real. Por ejemplo, en la frase “*Obama* ofreció ayer una rueda de prensa. El *presidente de EEUU* habló sobre economía y las relaciones con Irán”, tanto “Obama” como “presidente de EEUU” se refieren a una misma entidad extralingüística (Barack Obama), por lo que puede afirmarse que ambas expresiones son correferentes. El estudio de este fenómeno permitirá una mayor comprensión del texto, facilitando la deducción de relaciones de tipo taxonómico.
- Con un corpus de experimentación lo suficientemente amplio, sería posible estudiar las **colocaciones en UNL**, permitiendo medir el grado de cohesión con la que se presentan los patrones lingüísticos en las expresiones. Esto permitiría descartar términos utilizando las nociones equivalentes de *termhood* y *unithood* utilizadas en lenguaje natural.
- Una extensión clara del método lo constituye la deducción de **relaciones no taxonómicas**, a partir de los patrones lingüísticos que forman las estructuras argumentales verbales en la interlingua. Esta aproximación permitirá también inferir relaciones de tipo taxonómico entre conceptos a partir de las **preferencias de selección** de los argumentos, tal y como realizan algunos métodos de construcción de ontologías en lenguaje natural.
- Otra mejora sería la **identificación de atributos** de conceptos, a partir de las predicaciones de tipo *atributivo* o *estativo*, representadas en UNL mediante palabras universales verbales con la restricción “(icl>be)”.

BIBLIOGRAFÍA

- [1] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *20th International Conference on Very Large Data Bases*, 1994, pp. 487-499.
- [2] E. Aguirre, O. Ausa, E. Havy, and D. Martinez, "Enriching very large ontologies using the WWW," in *ECAI-2000 Workshop on Ontology Learning*, Berlin, 2000.
- [3] M. G. Ahsaei, M. Naghibzadeh, and S. E. Y. Naeini, "Semantic similarity assessment of words using weighted WordNet," *International Journal of Machine Learning and Cybernetics*, vol. 5, pp. 479-490, 2014.
- [4] E. Alfonseca and S. Manandhar, "Extending a lexical ontology by a combination of distributional semantics signatures," in *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, ed: Springer Berlin Heidelberg, 2002, pp. 1-7.
- [5] S. Beale, S. Nirenburg, and G. Mahesh, "Semantic Analysis in the Mikrokosmos Machine Translation Project," in *Second Symposium on Natural Language Processing (SNLP-95)*, Bangkok, Thailand, 1995.
- [6] S. Beale, S. Nirenburg, and K. Mahesh, "Semantic analysis in the Mikrokosmos Machine Translation Project," in *2nd Symposium on Natural Language Processing*, Bangkok, 1995, pp. 297-307.
- [7] M. Berland and E. Charniak, "Finding Parts in Very Large Corpora," in *37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, College Park, Maryland, 1999, pp. 57-64.
- [8] T. Berners-Lee and M. Fischetti, *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*. New York: Harper Collins Publishers, 1999.
- [9] C. Blake and W. Pratt, "Automatically Identifying Candidate Treatments from Existing Medical Literature," in *AAAI Spring Symposium on Mining Answers from Texts and Knowledge Bases*, Palo Alto, California, 2002.
- [10] I. Boguslavsky, J. Cardenosa, C. Gallardo, and L. Iraola, "The UNL Initiative: An Overview," in *Lecture Notes in Computer Science*. vol. 3406, ed: Springer, 2005, pp. 377-387.

- [11] C. Boitet, I. M. Boguslavskij, and J. Cardeñosa, "An Evaluation of UNL Usability for High Quality Multilingualization and Projections for a Future UNL++ Language," in *Lecture Notes in Computer Science*, ed: Springer Berlin / Heidelberg, 2007, pp. 361-373.
- [12] S. Bordag, "Word sense induction: Triplet-based clustering and automatic evaluation," in *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy, 2006, pp. 137-144.
- [13] S. Brody and M. Lapata, "Bayesian word sense induction," in *12th Conference of the European Chapter of the Association for Computational Linguistics*, Athens, Greece, 2009, pp. 102-111.
- [14] T. C. D. Bueno, H. C. Hoeschl, and C. K. Stradiotto, "Ontojuris Project: A Multilingual Legal Document Search System Based on a Graphical Ontology Editor," in *Business Information Systems Workshops*. vol. 57, W. Abramowicz, R. Tolksdorf, and K. Węcel, Eds., ed: Springer Berlin Heidelberg, 2010, pp. 310-321.
- [15] P. Buitelaar, P. Cimiano, and B. Magnini, "Ontology learning from text: An overview," in *Ontology Learning from Text: Methods, Evaluation and Applications*, P. Buitelaar, P. Cimiano, and B. Magnini, Eds., ed Amsterdam: IOS Press, 2005, pp. 3-12.
- [16] P. Buitelaar, P. Cimiano, S. Racioppa, and M. Siegel, "Ontology-based information extraction with SOBA," in *International Conference on Language Resources and Evaluation (LREC)*, Genoa, Italy, 2006.
- [17] P. Buitelaar, M. Olejnik, and M. Sintek, "A Protégé Plug-In for Ontology Extraction from Text Based on Linguistic Analysis," in *1st European Semantic Web Symposium (ESWS)*, Heraklion, Greece, 2004.
- [18] J. Cardeñosa, G. C., and L. Iraola, "UNL as a Text Content Representation language for Information Extraction," in *Lecture Notes on Artificial Intelligence*. vol. 4027, ed Berlin Heidelberg: Springer-Verlag, 2006, pp. 507-518.
- [19] J. Cardeñosa, C. Carolina, and L. Iraola, "Interlinguas: A Classical Approach for the Semantic Web. A Practical Case," in *MICAI 2006: Advances in Artificial Intelligence. Lecture Notes in Computer Science*, ed: Springer Berlin / Heidelberg, 2006, pp. 932-942.
- [20] J. Cardeñosa, M. A. de la Villa, and C. Gallardo, "Linguistic Patterns for Encyclopaedic Information Extraction," in *10th International Conference, FQAS 2013*, Granada, Spain, 2013, pp. 661-670.
- [21] J. Cardeñosa, C. G., and A. Toni, "Methodology for Language Analysis and Generation in Closed Domains: Pharmaceutical Leaflet," in *Advanced Research in Artificial Intelligence*. vol. 2, K. Markov, K. Ivanova, and I. Mitov, Eds., ed Sofia, Bulgaria: Institute of Information Theories and Applications FOI ITHEA, 2008, pp. 132-137.
- [22] J. Cardeñosa, C. Gallardo, and M. A. de la Villa, "Interlingual information extraction as a solution for multilingual QA systems," in *Flexible Query Answering Systems, 8th International Conference, FQAS 2009*, Roskilde, Denmark, 2009, pp. 500-511.
- [23] J. Cardeñosa, C. Gallardo, L. Iraola, and M. A. de la Villa, "A New Knowledge Representation Model to Support Multilingual Ontologies - A Case Study," in *International Conference on Semantic Web and Web Services (SWW'08)*, San Diego, USA, 2008, pp. 313-319.

- [24] J. Cardeñosa, C. Gallardo, and A. Toni, "Multilingual Cross Language Information Retrieval - A new approach," in *Computer Science and Information Technologies (CSIT2009)*, Yerevan, Armenia, 2009, pp. 284-287.
- [25] S. Cederberg and D. Widdows, "Using LSA and noun coordination information to improve the precision and recall of automatic hyponymy extraction," in *7th Conference on Natural Language Learning at HLT-NAACL*, Edmonton, Canada, 2003, pp. 111-118.
- [26] F. Cimiano, S. Staab, and J. Tane, "Deriving concept hierarchies from text by smooth formal concept analysis," in *GI Workshop Lernen-Wissen-Adaptivitat (LLWA)*, Karlsruhe, Germany, 2003, pp. 72-79.
- [27] P. Cimiano and S. Staab, "Learning concept hierarchies from text with a guided hierarchical clustering algorithm," in *ICML 2005 workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods*, Bonn, Germany, 2005, pp. 6-16.
- [28] P. Cimiano and J. Völker, "Text2Onto: a framework for ontology learning and data-driven change discovery," in *10th International Conference on Natural Language Processing and Information Systems*, Alicante, Spain, 2005, pp. 227-238.
- [29] E. Commision, "HEREIN Project (IST-2000-29355). Final Report," 2003.
- [30] O. Corcho and A. Gómez Pérez, "Evaluating knowledge representation and reasoning capabilities of ontology specification languages," in *ECAI 2000 Workshop on applications of ontologies and problem solving methods*, Berlin, 2000.
- [31] H. Cherfi and Y. Toussaint, "How far association rules and statistical indices help structure terminology," in *ECAI 2002 Workshop on Machine Learning and Natual Language Processing for Ontology Engineering (OLT'2002)*, Lyon, France, 2002.
- [32] A. N. Chomsky, *Lectures on Government and Binding (Studies in generative grammar 9)*. Dordrecht: Forish, 1981.
- [33] K. W. Church and P. Hanks, "Word association norms, mutual information, and lexicography," *Computational linguistics*, vol. 16, pp. 22-29, 1990.
- [34] M. Dillon and A. S. Gray, "FASIT: A fully automatic syntactically based indexing system," *Journal of the American Society for Information Science*, vol. 35, pp. 99-108, 1983.
- [35] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York: John Wiley & Sons, 2001.
- [36] D. Faure and C. Nédellec, "A corpus-based conceptual clustering method for verb frames and ontology acquisition," in *LREC Workshop on Adapting Lexical and Corpus Resources to Sublanguages and Applications*, Granada, Spain, 1998, pp. 707-728.
- [37] D. Faure and C. Nédellec, "Asium: Learning Subcategorization Frames and Restrictions of Selection," in *10th Conference on Machine Learning (ECML98) - Workshop on Text Mining*, Chemnitz, Germany, 1998.
- [38] C. Fellbaum, *WordNet: an Electronic Lexical Database*: MIT, Press, 1998.
- [39] M. Fernández, I. Cantador, V. López, D. Vallet, P. Castells, and E. Motta, "Semantically enhanced Information Retrieval: an ontology-based approach," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 9, pp. 434-452, 2011.

- [40] B. Fortuna, D. Mladenic, and M. Grobelnik, "Semi-automatic construction of topic ontology," in *8th International Multiconference Information Society IS 2005*, Ljubljana, Slovenia, 2005, pp. 170-173.
- [41] K. T. Frantzi and S. Ananiadou, "The C-value/NC-value domain-independent method for multi-word term extraction," *Journal of Natural Language Processing*, vol. 6, pp. 145-179, 1999.
- [42] K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi, "Information extraction: Identifying protein names from biological papers," in *Pacific Symposium on Biocomputing '98 (PSB'98)*, Hawaii, 1998, pp. 707-718.
- [43] F. Furst and F. Trichet, "Heavyweight ontology engineering," in *International Conference on Ontologies, Databases and Application of Semantics (ODBASE)*, Montpellier, France, 2006.
- [44] R. Gaizauskas, G. Demetriou, P. J. Artymiuk, and P. Willett, "Protein structures and information extraction from biological texts: the PASTA system," *Bioinformatics*, vol. 19, pp. 135-143, 2003.
- [45] C. Gallardo and J. Cardeñosa, "Standardization of the generation process in a multilingual environment," in *Computer Science and Information Technologies (CSIT2009)*, Yerevan, Armenia, 2009, pp. 288-292.
- [46] B. Ganter and R. Wille, *Formal Concept Analysis - Mathematical Foundations*: Springer Verlag, 1999.
- [47] J. Gantz and D. Reinsel, "The Digital Universe Decade - Are You Ready?," *International Data Corp. IVIEW*, pp. 1-15, 2010.
- [48] J. B. Gao, B. W. Zhang, and X. H. Chen, "A WordNet-based semantic similarity measurement combining edge-counting and information content theory," *Engineering Applications of Artificial Intelligence*, vol. 39, pp. 80-88, 2015.
- [49] F. Giunchiglia and I. Zaihrayeu, "Lightweight ontologies," in *Encyclopedia of Database Systems*, M. T. Ozsü and L. Liu, Eds., ed Heidelberg: Springer, 2008.
- [50] G. Grefenstette, *Cross-language information retrieval*. Norwell, USA: Kluwer Academic Publishers, 1998.
- [51] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge acquisition*, vol. 5, pp. 199-220, 1993.
- [52] N. Guarino, C. Masolo, G. Vetere, and Ontoseek, "Content-based access to the Web," *IEEE Intelligent Systems*, vol. 14, pp. 70-80, 1999.
- [53] Z. Harris, *Mathematical Structures of Language*. New York: Wiley-Interscience, 1968.
- [54] M. A. Hearst, "Automatic acquisition of hyponyms from large text corpora," in *14th conference on Computational linguistics*, Nantes, France, 1992, pp. 539-545.
- [55] W. Hengliang and Z. Weiwei, "A Web Information Extraction Method Based on Ontology," *Advances in Information Sciences & Service Sciences*, vol. 4, pp. 199-206, 2012.
- [56] D. Hindle, "Noun classification from predicate-argument structures," in *28th annual meeting on Association for Computational Linguistics*, Pittsburgh, USA, 1990, pp. 268-275.

- [57] A. Hliaoutakis, G. Varelas, E. Voutsakis, E. G. Petrakis, and E. Milios, "Information retrieval by semantic similarity," *International Journal on Semantic Web and Information Systems*, vol. 2, pp. 55-73, 2006.
- [58] J. R. Hobbs, "Information extraction from biomedical text," *Journal of Biomedical Informatics*, vol. 35, pp. 260-264, 2002.
- [59] K. Kageura and B. Umino, "Methods of automatic term recognition: A review," *Terminology*, vol. 3, pp. 259-289, 1996.
- [60] M. Kavalec and V. Svátek, "A Study on Automated Relation Labelling in Ontology Learning," in *Ontology Learning from Text: Methods, Evaluation and Applications*, P. Buitelaar, P. Cimiano, and B. Magnini, Eds., ed: IOS Press, 2005, pp. 44-58.
- [61] T. K. Landauer and S. T. Dumais, "A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge," *Psychological review*, vol. 104, pp. 211-240, 1997.
- [62] Y. Li, Z. A. Bandar, and D. McLean, "An approach for measuring semantic similarity between words using multiple information sources," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, pp. 871-882, 2003.
- [63] C.-Y. Lin and E. Hovy, "The automated acquisition of topic signatures for text summarization," in *18th conference on Computational Linguistics*, Saarbrücken, Germany, 2000, pp. 495-501.
- [64] D. Lin, "Concept discovery from text," in *International Conference on Computational Linguistics (COLING)*, Taipei, Taiwan, 2002, pp. 577-583.
- [65] D. Lin and P. Pantel, "Induction of semantic classes from natural language text," in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, San Francisco, USA, 2001, pp. 317-322.
- [66] A. Lipani, F. Piroi, L. Andersson, and A. Hanbury, "An information retrieval ontology for information retrieval nanopublications," in *Information Access Evaluation. Multilinguality, Multimodality, and Interaction*, ed: Springer, 2014, pp. 44-49.
- [67] V. Lopez and M. Enrico, "Ontology-driven question answering in Aqualog," in *9th International Conference on Applications of Natural Language to Information Systems (NLDB 2004)*, Manchester, England, 2004, pp. 89-102.
- [68] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of Research and Development*, vol. 2, pp. 159-165, 1958.
- [69] A. Maedche and S. Staab, "Discovering conceptual relations from text," in *14th European Conference on Artificial Intelligence (ECAI'2000)*, Amsterdam, 2000, pp. 321-325.
- [70] A. Maedche and S. Staab, "Ontology Learning for the Semantic Web," *IEEE Intelligent Systems*, vol. 16, pp. 72-79, 2001.
- [71] C. D. Manning and H. Schütze, *Foundations of statistical natural language processing*. Cambridge, Massachusetts: MIT Press, 1999.
- [72] P. Martin and P. W. Eklund, "Knowledge Retrieval in the World Wide Web," *Intelligent Systems and their Applications, IEEE*, vol. 15, pp. 18-25, 2000.
- [73] D. McGuinness and F. Van Harmelen, "OWL Web Ontology Language Overview," ed: World Wide Web Consortium, 2004.

- [74] A. Moreno, D. Isern, and A. C. L. Fuentes, "Ontology-based information extraction of regulatory networks from scientific articles with case studies for escherichia coli," *Expert Systems with Applications*, vol. 40, pp. 3266-3281, 2013.
- [75] K. Muraki, "PIVOT: Two-phase machine translation system," in *Second Machine Translation Summit*, Tokyo, Japan, 1989.
- [76] R. Navigli and P. Velardi, "Learning domain ontologies from document warehouses and dedicated web sites," *Computational Linguistics*, vol. 30, pp. 151-179, 2004.
- [77] S. Nirenburg and V. Raskin, *Ontological semantics*: MIT Press, 2004.
- [78] N. F. Noy, M. Crubezy, R. W. Fergerson, Knublauch, H., S. W. V. Tu, J., and M. A. Musen, "Protege-2000: An open-source ontology-development and knowledge-acquisition environment," in *Annual Symposium of the American Medical Informatics Association (AMIA-00)*, Los Angeles, 2003.
- [79] E. H. Nyberg and T. Mitamura, "The KANT system: fast, accurate, high-quality translation," in *14 th International Conference on Computational Linguistics (COLING'92)*, Nantes, France, 1992, pp. 1254-1258.
- [80] C. K. Ogden and I. A. Richards, *The meaning of meaning: a study of the influence of language upon thought and of the science of symbolism*. New York: Harcourt Brace Jovanovich, 1923.
- [81] G. Petasis, V. Karkaletsis, G. Paliouras, A. Krithara, and E. Zavitsanos, "Ontology population and enrichment: State of the art," in *Knowledge-driven multimedia information extraction and ontology evolution*, ed: Springer-Verlag, 2011, pp. 134-166.
- [82] M. Poesio and R. Stevenson, "Centering: A parametric theory and its instantiations," *Computational Linguistics*, vol. 30, pp. 309-363, 2004.
- [83] M. Rani, M. K. Mueyba, and O. P. Vyas, "A hybrid approach using ontology similarity and fuzzy logic for semantic question answering," in *Advanced Computing, Networking and Informatics*. vol. 1, ed: Springer International Publishing, 2014, pp. 601-609.
- [84] M.-L. Reinberger, P. Spyns, W. Daelemans, and R. Meersman, "Mining for lexons: Applying unsupervised learning methods to create ontology bases," in *International Conference on Ontologies, Databases and Applications of Semantics (ODEBASE)*, 2003, pp. 600-617.
- [85] M. Sabou, "Learning Web Service Ontologies: an Automatic Extraction Method and its Evaluation," *Ontology Learning from Text: Methods, Evaluation and Applications*, vol. 123, p. 125, 2005.
- [86] M. Sanderson. (1999). *The Information Retrieval Group*. Available: http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words.
- [87] S. Schlobach, M. Olstboorn, and M. de Rijke, "Type checking in open domain question answering," in *16th European conference on Artificial Intelligence (ECAI)*, Valencia, Spain, 2004.
- [88] A. Schutz and P. Buitelaar, "Relext: A tool for relation extraction from text in ontology extension," in *4th international conference on The Semantic Web ISWC'05*, Galway, Ireland, 2005, pp. 593-606.
- [89] H. Schütze, "Word Space," in *Advances in Neural Information Processing Systems 5*, Denver, USA, 1993, pp. 895-902.

- [90] S. Shamsfard and A. A. Barforoush, "The state of the art in ontology learning: a framework for comparison," *The Knowledge Engineering Review*, vol. 18, pp. 293 - 316, 2003.
- [91] G. Shikha, J. Vinod, and B. Pawan, "Combined Approach for Page Ranking In Information Retrieval System Using Context and TF-IDF Weight," *International Journal of Computer Science and Engineering*, vol. 2, pp. 39-42, 2014.
- [92] R. Snow, D. Jurafsky, and A. Y. Ng, "Learning syntactic patterns for automatic hypernym discovery," in *Advances in Neural Information Processing Systems 17*, 2004.
- [93] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," in *KDD workshop on text mining*, Boston, 2000, pp. 525-526.
- [94] M. Stevenson and M. A. Greenwood, "A semantic approach to IE pattern induction," in *43rd Annual Meeting on Association for Computational Linguistics (ACL'05)*, Ann Arbor, Michigan, 2005, pp. 379-386.
- [95] N. Sugiura, Y. Shigeta, N. Fukuta, N. Izumi, and T. Yamaguchi, "Towards On-the-fly Ontology Construction – Focusing on Ontology Quality Improvement," in *1st European Semantic Web Symposium (ESWS 2004)*, Heraklion, Greece, 2004, pp. 1-15.
- [96] H. Sundblad, "Automatic acquisition of hyponyms and meronyms from question corpora," in *15th European Conference on Artificial Intelligence*, Lyon, France, 2002.
- [97] M. Surve, S. Kagathara, P. Bhattacharyya, and A. E. Group, "AgroExplorer: a Meaning Based Multilingual Search Engine," in *International Conference on Digital Libraries (ICDL)*, New Delhi, India, 2004, pp. 167-174.
- [98] H. Uchida, "ATLAS-II: A machine translation system using conceptual structure as an Interlingua," in *Second Machine Translation Summit*, Tokyo, Japan, 1989, pp. 85-92.
- [99] H. Uchida. (2004). *The Universal Networking Language Specifications*, v3.3. Available: <http://www.undl.org>
- [100] EOLSS. (2002). *Encyclopedia of Life Support Systems*. Available: <http://www.eolss.net>
- [101] M. Uschold and M. Gruninger, "Ontologies and Semantics for Seamless Connectivity," *SIGMOD Rec.*, vol. 33, pp. 58-64, 2004.
- [102] P. Velardi, R. Navigli, A. Cuchiarelli, and F. Neri, "Evaluation of ontolearn, a methodology for automatic population of domain ontologies," in *Ontology Learning from Text: Methods, Applications and Evaluation*, P. Buitelaar, P. M. Cimiano, and B. Magnini, Eds., ed: IOS Press, 2005.
- [103] P. Vossen, *EuroWordNet: A multilingual database with lexical semantic networks*: Springer Netherlands, 1998.
- [104] D. Widdows, "Unsupervised methods for developing taxonomies by combining syntactic and statistical information," in *2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Morristown, NJ, 2003, pp. 197-204.
- [105] H. Wimmer and L. Zhou, "Word Sense Disambiguation for Ontology Learning," in *Nineteenth Americas Conference on Information Systems*, Chicago, USA, 2013.
- [106] H. F. Witchel, "Terminologie-Extraktion: Möglichkeiten der Kombination statistischer und musterbasierter Verfahren," in *Content and Communication: Terminology, Language Resources and Semantic Interoperability*, ed Würzburg: Ergon Verlag, 2004.

- [107] H. F. Witschel, "Using decision trees and text mining techniques for extending taxonomies," in *Workshop on Learning and Extending Lexical Ontologies by Using Machine Learning Methods (OntoML 05)*, Bonn, Germany, 2005.
- [108] X. Yan and X. Zhao, "Research on Sino-Tibetan Machine Translation Based on the Reusing of Domain Ontology," *International Journal of Online Engineering (ijOE)*, vol. 9, pp. 105-108, 2013.
- [109] D. Yarowsky, "Word-sense disambiguation using statistical models of Roget's categories trained on large corpora," in *14th Conference on Computational Linguistics*, Nantes, France, 1992, pp. 454-460.

Anexo I. UNIVERSAL NETWORKING LANGUAGE

Con el objetivo de describir las características de UNL, a continuación se muestra un extracto traducido del artículo [23], producido por Jesús Cardeñosa, Carolina Gallardo, Luis Iraola y Miguel Ángel de la Villa:

Durante los años noventa, la Universidad de las Naciones Unidas (UNU) desarrolló el denominado Universal Networking Language (UNL), un lenguaje para la representación de contenidos de forma independiente de la lengua con el propósito de eliminar las barreras lingüísticas en Internet. Las especificaciones de este lenguaje están accesibles gratuitamente en <http://www.undl.org>.

UNL está compuesto por tres elementos principales: palabras universales (Universal Words), relaciones y atributos. Formalmente, una expresión UNL puede verse como una red semántica, cuyos nodos son palabras universales vinculadas entre sí mediante relaciones UNL. Las palabras universales pueden ser modificadas utilizando los denominados atributos.

Universal Words

Las palabras universales (UW) conforman el vocabulario de UNL. El objetivo de cada UW es expresar un concepto que se pueda encontrar en cualquier lenguaje natural. UNL utiliza palabras y frases tomadas de los lenguajes naturales, especialmente del inglés. Los *átomos semánticos* (UWs) no son conceptos sino significados, extraídos principalmente del vocabulario inglés por conveniencia. Aunque las palabras universales adquieren sus significados a partir de los sentidos de las palabras inglesas, cada palabra universal expresa un significado muy claro, de forma que la ambigüedad léxica queda reducida al mínimo.

La razón de elegir el inglés es solo práctica: el vocabulario de este idioma está bien cubierto por muchos diccionarios autorizados, y hay diccionarios bilingües de inglés para casi cualquier otro lenguaje natural. Utilizar el vocabulario de un lenguaje natural para que sirva de vocabulario de la interlingua conllevaría, en principio, los problemas inherentes a los lenguajes naturales, como la ambigüedad y la falta de adecuación para representar conceptos

lexicalizados en otros lenguajes. Con el fin de paliar estas dificultades, a las *headwords* inglesas se le añaden restricciones semánticas, formando así las denominadas *UW restringidas*.

Por ejemplo, el verbo “land” en inglés tiene varios significados y marcos de predicado. En uno de sus sentidos, ejemplificado en la Figura 64, la restricción *icl>do* indica que es un verbo de acción, y *src* representa el origen (fuente). Los argumentos agente (*agt*) y lugar de aterrizaje (*plc*) añaden información más concreta.

Figura 64. Restricciones y argumentos para las UWs

Aunque este método está lejos de ser perfecto, demuestra algunas ventajas:

1. Permite tener una forma consensuada y “normalizada” de definir palabras universales y de interpretarlas. De esta forma, se puede capturar el significado de una UW solamente con inspeccionarla.
2. Se elimina la ambigüedad inherente al vocabulario del lenguaje natural
3. Permite servir de eje para conectar los vocabularios de los lenguajes naturales

El Sistema UNL se describe en[10], donde se discuten cuestiones más complejas, como los desajustes léxicos o la representación de términos específicos de cada cultura, entre otros asuntos.

Relaciones

El segundo componente de UNL es un conjunto de 41 relaciones conceptuales. Estas relaciones forman un conjunto cerrado definido exhaustivamente en las especificaciones de UNL. La filosofía de las relaciones conceptuales consiste en caracterizar un conjunto de nociones semánticas aplicables a la mayoría de los lenguajes naturales existentes, como por ejemplo la noción de un iniciador o causa de un evento (agente). El conjunto seleccionado de

relaciones conceptuales debe ser relevante y suficiente para producir una representación semántica interlingual de cualquier lenguaje natural. Las relaciones UNL cubren relaciones argumentales (agente, objeto, objetivo, beneficiario, etc.), lógicas (disyunción, conjunción o equivalencia), temporales y locativas, condicionales y causales, y relaciones circunstanciales (forma, instrumento y método).

Las relaciones conceptuales están definidas intensionalmente en las especificaciones del lenguaje, imponiendo restricciones semánticas sobre las palabras universales que pueden actuar como primer argumento (origen) o como segundo argumento (destino). Por ejemplo, en el caso de la relación “agente”, las restricciones indican que la palabra universal origen debe denotar una acción que requiere un iniciador (y no un evento que “solo ocurre”) y la palabra universal destino debe denotar una entidad.

Atributos

Las distinciones semánticas como el número (singular o plural) o la referencia (definida o indefinida) de las entidades; el tiempo y aspecto de las acciones, y las modalidades epistémicas y deónticas de una expresión se realizan, en la mayoría de lenguajes naturales, a través de mecanismos lingüísticos que operan sobre los elementos léxicos, o mediante construcciones sintácticas específicas.

UNL hace uso de un conjunto de atributos para expresar estas distinciones semánticas, que operan directamente sobre las palabras universales. En una expresión UNL, los atributos aparecen vinculados a las palabras universales como elementos que ofrecen flexibilidad para adaptar su significado al contexto concreto que se esté representando. Los *atributos* que pueden modificar UWs tienen una clara relevancia lingüística. Hay atributos para representar distinciones de referencia, aspectos temporales, intenciones del hablante, etc.

Seleccionar la relación conceptual apropiada entre palabras universales y utilizar los atributos correspondientes permite a UNL expresar el contenido proposicional de cualquier oración. Por ejemplo, las oraciones atributivas “El agua es vida” (Español), “Wasser ist Leben” (Alemán), “Water is life”(English) y “L'eau est vie”(Francés) se expresan todas mediante el mismo grafo UNL (Figura 65):

Figura 65. Grafo UNL

UNL proporciona una representación lo suficientemente abstracta como para ajustarse a cualquier lenguaje. La influencia anglosajona en el vocabulario puede imponer algunos obstáculos a la hora de vincular las UWs con las palabras de un lenguaje natural, pero para tareas de gran escala, esta aproximación maximal (utilizar significados de palabras para el vocabulario de la interlingua) parece más adecuada que basarse en primitivas semánticas (difíciles de definir e insuficientes para representar las diferencias en los significados de las palabras del lenguaje natural). Las relaciones UNL se basan en tradiciones bien establecidas, como los roles temáticos y las gramáticas de casos; cubriendo la mayoría de roles que pueden expresarse en los lenguajes naturales.

Anexo II. EJEMPLO DE APLICACIÓN DE ALGORITMO DE RESOLUCIÓN DE ANÁFORA

A continuación se ilustrará el funcionamiento del algoritmo de resolución de la anáfora, especificado en pseudocódigo en la sección 4.2.2, aplicándolo paso a paso sobre un documento de ejemplo formado por las siguientes 4 oraciones:

1. *Juan telefoneó a Daniel, director de área, para confirmarle su asistencia la reunión del proyecto.*
2. *Después le pidió a María, su ayudante, que fuera con él a la reunión.*
3. *María tuvo que estudiar la documentación del proyecto, ya que desconocía muchos detalles.*
4. *Juan no apareció por la oficina hasta el día de la reunión, así que no pudo resolver algunas dudas.*

La correspondiente codificación en UNL aparece en las figuras 66 - 69 (para aumentar la claridad, se han omitido tanto las restricciones de las palabras universales como los atributos, además de numerar los nodos para poder ser referenciados posteriormente). El documento contiene un total de 6 pronombres anafóricos y 3 nombres propios de persona.

El resultado de la resolución de los pronombres deberá ser el siguiente:

Tabla 36. Antecedentes de los pronombres del ejemplo

Expresión	Anáfora	Antecedente
e ₁	n ₈ : he(icl>person)	"Juan"
e ₂	n ₂ : he(icl>person)	
	n ₇ : she(icl>person)	"María"
	n ₈ : he(icl>person)	"Juan"
e ₃	n ₇ : she(icl>person)	"María"
e ₄	n ₂ : she(icl>person)	

El algoritmo comienza con una secuencia de 5 pasos en los que se inicializan las variables:

- Paso 1: foco = null
- Paso 2: vigencia_foco = 0
- Paso 3: G("he(icl>person)") = M
- Paso 4: G("she(icl>person)") = F

A continuación comienza el bucle que recorre en orden cada expresión del documento:

Expresión e_1

"Juan telefoneó a Daniel, director de área, para confirmarle su presencia en la reunión del proyecto"

Figura 66. Ejemplo de resolución de anáfora (expresión 1)

- Paso 5.1: se obtiene el conjunto N de nodos con nombres propios. En esta expresión hay dos (los nodos correspondientes a "Juan" y "Daniel"):

$$N = \{n_2, n_4\}$$

- Paso 5.2: se construye la lista P de nodos con pronombres personales, ordenados de menor a mayor distancia con el nodo entrada de la expresión. Solo hay uno, luego la lista estará formada por un elemento:

$$P = \langle n_8 \rangle$$

- Paso 5.3: se recorren en orden los pronombres de la lista P . Para $n_8 \rightarrow$
- Paso 5.3 (i): se obtiene el conjunto R , formado por los nombres propios de la expresión (conjunto N) que no pueden ser antecedentes de n_8 . “Juan” y “Daniel” no cumplen ninguna de las condiciones (**a**, **b**, y **c**), por tanto el conjunto es vacío:

$$R = \emptyset$$

- Paso 5.3 (ii): se construye el conjunto C de nodos candidatos a antecedente, que estará formado por todos los elementos de N (n_2 y n_4), ya que no hay nombres rechazados ($R = \emptyset$):

$$C = \{n_2, n_4\}$$

- Paso 5.3 (iii): hay nombres diferentes entre los candidatos, así que se calcula la puntuación PCA de cada uno y se eligen aquéllos con la puntuación más alta:

PCA(n_2):

- $PC_{DISTANCIA} := (5 - Distancia(n_2, n_8))/5 = (5 - 4)/5 = 0,2$
- $PC_{FUNCION} := 1$ (FunciónNodo(n_2) = arg_principal_verbo)
- $PC_{FOCO} := 0$ (“Juan” ≠ foco)
- $PCA(n_2) = (0,4 \cdot 0,2) + (0,4 \cdot 1) + (0,2 \cdot 0) = \underline{\underline{0,48}}$

PCA(n_4):

- $PC_{DISTANCIA} = (5 - Distancia(n_4, n_8))/5 = (5 - 3)/5 = 0,4$
- $PC_{FUNCION} = 0,7$ (FunciónNodo(n_4) = arg_secundario_verbo)
- $PC_{FOCO} := 0$ (“Daniel” ≠ foco)
- $PCA(n_4) = (0,4 \cdot 0,4) + (0,4 \cdot 0,7) + (0,2 \cdot 0) = \underline{\underline{0,44}}$

El nodo con la puntuación más alta es n_2 , por lo que:

$$A = \{f_{uw}(n_2)\} = \{\text{“Juan”}\}$$

- Paso 5.3 (iv):

antecedente = null

- Paso 5.3 (v): el conjunto A tiene un único nombre, luego se elige como antecedente:

antecedente = "Juan"

- Paso 5.3 (vi): se le asigna, al nombre elegido, el mismo género que el pronombre anafórico, y se almacena en la tabla de géneros G . Es decir:

$$G(\text{"Juan"}) = G(\text{"he(icl>person)"}) = \text{Masculino}$$

El contenido de G ahora es:

UW	Género
he(icl>person)	Masculino
she(icl>person)	Femenino
"Juan"	Masculino

Se sustituye el pronombre por el nombre propio elegido:

$$f_{uw}(n_8) = \text{antecedente} = \text{"Juan"}$$

Se añade el nodo que contenía el pronombre al conjunto N , ya que ahora contiene un nombre propio:

$$N = \{n_2, n_4, n_8\}$$

- Paso 5.4: en este paso y los siguientes se comprueba si hay que cambiar el foco. En primer lugar, se calcula el conjunto H de nodos candidatos, a partir del conjunto N . El nodo n_8 es el único que debe descartarse, ya que $\text{FunciónNodo}(n_8) = \text{arg_nominal}$:

$$H = \{n_2, n_4\}$$

- Paso 5.5: se seleccionan aquéllos candidatos con puntuación **PCF** máxima:

$\text{PCF}(n_2)$:

- $\text{PC}_{\text{DISTANCIA}} = (5 - \text{Distancia}(n_2, \text{NodoEntrada}(e_1)))/5 = (5 - 1)/5 = 0,8$
- $\text{PC}_{\text{FUNCION}} = 1$ ($\text{FunciónNodo}(n_2) = \text{arg_principal_verbo}$)
- $\text{PCF}(n_2) = (0,5 \cdot 0,8) + (0,5 \cdot 1) = \underline{\underline{0,9}}$

$\text{PCF}(n_4)$:

- $\text{PC}_{\text{DISTANCIA}} = (5 - \text{Distancia}(n_4, \text{NodoEntrada}(e_1)))/4 = (5 - 1)/5 = 0,8$
- $\text{PC}_{\text{FUNCION}} = 0,7$ ($\text{FunciónNodo}(n_4) = \text{arg_secundario_verbo}$)
- $\text{PCF}(n_4) = (0,5 \cdot 0,8) + (0,5 \cdot 0,7) = \underline{\underline{0,75}}$

Hay un único nodo (n_2) con la puntuación más alta (0,9), por lo que:

$$F = \{f_{uw}(n_2)\} = \{\text{"Juan"}\}$$

- Paso 5.6: el conjunto F solo tiene un elemento, así que es elegido como foco:

$$\text{foco} = \text{"Juan"}$$

$$\text{permanencia_foco} = 3$$

Expresión e_2

“Después le pidió a María, su ayudante, que fuera con él a la reunión”

Figura 67. Ejemplo de resolución de anáfora (expresión 2)

- Paso 5.1: se obtiene el conjunto N de nodos con nombres propios. En esta expresión hay solo uno:

$$N = \{n_3\}$$

- Paso 5.2: se construye la lista P de nodos con pronombres personales, ordenados de menor a mayor distancia con el nodo entrada de la expresión. En esta expresión hay tres:

$$P = \langle n_2, n_7, n_8 \rangle$$

- Paso 5.3: se recorren en orden los pronombres de la lista P . Para $n_2 \rightarrow$
 - Paso 5.3 (i): el nodo n_3 está en el mismo *scope* que el pronombre n_2 , luego cumple la condición **b**:

$$\text{Scope}(n_2) = \text{Scope}(n_3)$$

$$R = \{f_{uw}(n_3)\} = \{\text{"María"}\}$$

- Paso 5.3 (ii): el único nodo del conjunto N es descartado. No existen candidatos:

$$C = \emptyset$$

- Paso 5.3 (iii):

$$A = \emptyset$$

- Paso 5.3 (iv):

antecedente = null

- Paso 5.3 (v): al no haber ningún candidato a antecedente, se utiliza el foco actual "Juan", que es compatible en género con el pronombre ($G(\text{"Juan"}) = G(\text{"he(icl>person)"} = \text{Masculino})$):

antecedente = foco = "Juan"

- Paso 5.3 (vi): se le asigna, al nombre elegido, el mismo género que el pronombre anafórico, y se almacena en la tabla de géneros G . Es decir:

$G(\text{"Juan"}) = \text{Masculino}$ (ya estaba almacenado en G)

El contenido de G permanece invariable:

UW	Género
he(icl>person)	Masculino
she(icl>person)	Femenino
"Juan"	Masculino

Se sustituye el pronombre por el nombre propio elegido:

$$f_{uw}(n_2) = \text{antecedente} = \text{"Juan"}$$

Se añade el nodo que contenía el pronombre al conjunto N , ya que ahora contiene un nombre propio:

$$N = \{n_2, n_3\}$$

- Paso 5.3: Para $n_7 \rightarrow$

- Paso 5.3 (i): el género de “Juan” es conocido (fue establecido durante el procesamiento de la expresión e_1) y no es compatible con el género del pronombre n_7 (she). Se cumple entonces la condición **a**, lo que impide que sea el antecedente:

$$G(\text{“Juan”}) = \text{Masculino} \neq G(\text{“she(icl>person)”})$$

$$R = \{f_{uw}(n_2)\} = \{\text{“Juan”}\}$$

- Paso 5.3 (ii): el conjunto C estará formado por el nodo n_3 , ya que n_2 es descartado:

$$C = \{n_3\}$$

- Paso 5.3 (iii): solo hay un candidato, así que no es necesario calcular la puntuación:

$$A = \{f_{uw}(n_3)\} = \{\text{“María”}\}$$

- Paso 5.3 (iv):

antecedente = null

- Paso 5.3 (v):

antecedente = “María”

- Paso 5.3 (vi): se le asigna, al nombre elegido, el mismo género que el pronombre anafórico, y se almacena en la tabla de géneros G . Es decir:

$$G(\text{“María”}) = \text{Femenino}$$

El contenido de G será el siguiente:

UW	Género
he(icl>person)	Masculino
she(icl>person)	Femenino
"Juan"	Masculino
"María"	Femenino

Se sustituye el pronombre por el antecedente:

$$f_{uw}(n_7) = \text{antecedente} = \text{"María"}$$

Se añade el nodo que contenía el pronombre al conjunto **N**, ya que ahora contiene un nombre propio:

$$N = \{n_2, n_3, n_7\}$$

- Paso 5.3: Para $n_8 \rightarrow$

- Paso 5.3 (i): el nodo n_3 ("María") es femenino, y por tanto incompatible en género con el pronombre n_8 ("he"), cumpliendo la condición **a**. El nodo n_7 ("María"), aparte de incompatible en género, también se sitúa en el mismo *scope* que el pronombre, cumpliendo la condición **b**:

$$\text{Scope}(n_7) = \text{Scope}(n_3)$$

$$G(\text{"María"}) = \text{Femenino} \neq G(\text{"he(icl>person)"})$$

$$R = \{f_{uw}(n_3), f_{uw}(n_7)\} = \{\text{"María"}\}$$

- Paso 5.3 (ii): el conjunto **C** estará formado por el nodo n_2 , ya que el resto son descartados:

$$C = \{n_2\}$$

- Paso 5.3 (iii): solo hay un candidato, así que no es necesario calcular la puntuación:

$$A = \{f_{uw}(n_2)\} = \{\text{"Juan"}\}$$

- Paso 5.3 (iv):

antecedente = null

- Paso 5.3 (v):

antecedente = "Juan"

- Paso 5.3 (vi): se le asigna, al nombre elegido, el mismo género que el pronombre anafórico, y se almacena en la tabla de géneros G . Es decir:

$$G(\text{"Juan"}) = \text{Masculino}$$

El contenido de G permanece invariable:

UW	Género
he(icl>person)	Masculino
she(icl>person)	Femenino
"Juan"	Masculino
"María"	Femenino

Se sustituye el pronombre por el antecedente:

$$f_{uw}(n_8) = \text{antecedente} = \text{"Juan"}$$

Se añade el nodo que contenía el pronombre al conjunto N , ya que ahora contiene un nombre propio:

$$N = \{n_2, n_3, n_7, n_8\}$$

- Paso 5.4: se calcula el conjunto H de nodos candidatos a foco, a partir del conjunto N . Todos cumplen el requisito funcional, así que:

$$H = \{n_2, n_3, n_7, n_8\}$$

- Paso 5.5: se selecciona el candidato con puntuación PCF máxima:

PCF(n_2):

- $PC_{DISTANCIA} = (5 - Distancia(n_2, NodoEntrada(e_2)))/5 = (5 - 1)/5 = 0,8$
- $PC_{FUNCION} = 1$ (FunciónNodo(n_2) = arg_principal_verbo)
- $PCF(n_2) = (0,5 \cdot 0,8) + (0,5 \cdot 1) = \underline{\underline{0,9}}$

PCF(n_3):

- $PC_{DISTANCIA} = (5 - Distancia(n_3, NodoEntrada(e_2)))/5 = (5 - 1)/5 = 0,8$
- $PC_{FUNCION} = 0,7$ (FunciónNodo(n_3) = arg_secundario_verbo)
- $PCF(n_3) = (0,5 \cdot 0,8) + (0,5 \cdot 0,7) = \underline{\underline{0,75}}$

PCF(n_7):

- $PC_{DISTANCIA} = (5 - Distancia(n_7, NodoEntrada(e_2)))/5 = (5 - 2)/5 = 0,6$
- $PC_{FUNCION} = 1$ (FunciónNodo(n_7) = arg_principal_verbo)
- $PCF(n_7) = (0,5 \cdot 0,6) + (0,5 \cdot 1) = \underline{\underline{0,8}}$

PCF(n_8):

- $PC_{DISTANCIA} = (5 - Distancia(n_8, NodoEntrada(e_2)))/5 = (5 - 2)/5 = 0,6$
- $PC_{FUNCION} = 0,7$ (FunciónNodo(n_8) = arg_secundario_verbo)
- $PCF(n_8) = (0,5 \cdot 0,6) + (0,5 \cdot 0,7) = \underline{\underline{0,65}}$

Hay un único nodo (n_2) con la puntuación más alta (0,9), por lo que:

$$F = \{f_{uw}(n_2)\} = \{\text{"Juan"}\}$$

- Paso 5.6: el conjunto F solo tiene un elemento, así que es elegido como foco:

$$\text{foco} = \text{"Juan"}$$

$$\text{permanencia_foco} = 3$$

Expresión e_3

«María tuvo que estudiar la documentación del proyecto, ya que desconocía muchos detalles»

Figura 68. Ejemplo de resolución de anáfora (expresión 3)

- Paso 5.1: se obtiene el conjunto N de nodos con nombres propios. En esta expresión hay solo uno, “María”, en el nodo n_2 :

$$N = \{n_2\}$$

- Paso 5.2: solo hay un pronombre (“she(ict>person)”) en el nodo n_7 :

$$P = \langle n_7 \rangle$$

- Paso 5.3: se recorren en orden los pronombres de la lista P . Para $n_7 \rightarrow$

- Paso 5.3 (i): el nodo n_2 no cumple ninguna de las condiciones **a**, **b** y **c**, por lo que el conjunto R estará vacío:

$$R = \emptyset$$

- Paso 5.3 (ii): al ser R vacío, el conjunto C será igual a N :

$$C = \{n_2\}$$

- Paso 5.3 (iii): solo hay un candidato, así que no es necesario calcular la puntuación PCA:

$$A = \{f_{uw}(n_2)\} = \text{"María"}$$

- Paso 5.3 (iv):

antecedente = null

- Paso 5.3 (v): se elige como antecedente al único nodo de A :

antecedente = "María"

- Paso 5.3 (vi): se le asigna, al nombre elegido, el mismo género que el pronombre anafórico, y se almacena en la tabla de géneros G . Es decir:

$$G(\text{"María"}) = \text{Femenino (ya estaba almacenado)}$$

El contenido de G permanece invariable:

UW	Género
he(icl>person)	Masculino
she(icl>person)	Femenino
"Juan"	Masculino
"María"	Femenino

Se sustituye el pronombre por el nombre propio elegido:

$$f_{uw}(n_7) = \text{antecedente} = \text{"María"}$$

Se añade el nodo que contenía el pronombre al conjunto N , ya que ahora contiene un nombre propio:

$$N = \{n_2, n_7\}$$

- Paso 5.4: se calcula el conjunto H de nodos candidatos a foco, a partir del conjunto N . No se descarta ningún elemento:

$$H = \{n_2, n_7\}$$

- Paso 5.5: ambos nodos contienen el mismo nombre ("María") por lo que no es necesario calcular las puntuaciones:

$$F = \{\text{"María"}\}$$

- Paso 5.6: hay cambio de foco:

$$\text{foco} = \text{"María"}$$

$$\text{permanencia_foco} = 3$$

Expresión e_4

"Juan no apareció por la oficina hasta la reunión, así que no pudo resolver algunas dudas"

Figura 69. Ejemplo de resolución de anáfora (expresión 4)

- Paso 5.1: se obtiene el conjunto N de nodos con nombres propios. En esta expresión hay solo uno, “Juan”, en el nodo n_6 :

$$N = \{n_6\}$$

- Paso 5.2: solo hay un pronombre en la expresión (“she(icl>person)”) en el nodo n_2 :

$$P = \langle n_2 \rangle$$

- Paso 5.3: se recorren en orden los pronombres de la lista P . Para $n_2 \rightarrow$

- Paso 5.3 (i): el género de “Juan” no es compatible con el género del pronombre she(icl>person). Además, el nodo n_2 α -comanda a n_6 , por lo que se cumplen las condiciones **a** y **c**:

$$n_2 \alpha\text{-comanda a } n_6$$

$$G(\text{“Juan”}) = \text{Masculino} \neq G(\text{“she(icl>person)”})$$

$$R = \{\text{“Juan”}\}$$

- Paso 5.3 (ii): no existe en la expresión ningún otro candidato:

$$C = \emptyset$$

- Paso 5.3 (iii):

$$A = \emptyset$$

- Paso 5.3 (iv):

$$\text{antecedente} = \text{null}$$

- Paso 5.3 (v): no hay candidatos a antecedente en la expresión. Se comprueba si el foco (“María”) es compatible en género, y se selecciona en caso afirmativo.

$$G(\text{“María”}) = G(\text{“she(icl>person)”}) = \text{Femenino}$$

$$\text{antecedente} = \text{foco} = \text{“María”}$$

- Paso 5.3 (vi): se le asigna, al nombre elegido, el mismo género que el pronombre anafórico, y se almacena en la tabla de géneros G . Es decir:

$G(\text{"María"}) = \text{Femenino}$ (ya estaba almacenado en la tabla)

El contenido de G permanece invariable:

UW	Género
he(icl>person)	Masculino
she(icl>person)	Femenino
"Juan"	Masculino
"María"	Femenino

Se sustituye el pronombre por el nombre propio elegido:

$$f_{uw}(n_2) = \text{antecedente} = \text{"María"}$$

Se añade el nodo que contenía el pronombre al conjunto N , ya que ahora contiene un nombre propio:

$$N = \{n_2, n_6\}$$

- Paso 5.4: se calcula el conjunto H de nodos candidatos a foco, a partir del conjunto N . No se descarta ningún elemento:

$$H = \{n_2, n_6\}$$

- Paso 5.5: se selecciona el candidato con puntuación **PCF** máxima:

$PCF(n_2)$:

- $PC_{\text{DISTANCIA}} = (5 - \text{Distancia}(n_2, \text{NodoEntrada}(e_4)))/5 = (5 - 1)/5 = 0,8$
- $PC_{\text{FUNCION}} = 1$ ($\text{FunciónNodo}(n_2) = \text{arg_principal_verbo}$)
- $PCF(n_2) = (0,5 \cdot 0,8) + (0,5 \cdot 1) = \underline{\underline{0,9}}$

$PCF(n_6)$:

- $PC_{\text{DISTANCIA}} = (5 - \text{Distancia}(n_6, \text{NodoEntrada}(e_4)))/5 = (5 - 2)/5 = 0,6$
- $PC_{\text{FUNCION}} = 1$ ($\text{FunciónNodo}(n_6) = \text{arg_principal_verbo}$)
- $PCF(n_2) = (0,5 \cdot 0,6) + (0,5 \cdot 1) = \underline{\underline{0,8}}$

El nodo n_2 es el que tiene la puntuación más alta:

$$F = \{f_{uw}(n_2)\} = \{\text{"María"}\}$$

- Paso 5.6: se reafirma el foco "María":

$$\text{foco} = \text{"María"}$$

$$\text{permanencia_foco} = 3$$

Anexo III. LENGUAJE DE ESPECIFICACIÓN DE PATRONES LINGÜÍSTICOS

La librería que implementa el modelo permite definir un patrón lingüístico mediante un lenguaje de especificación especial. El objetivo es la persistencia de los patrones, permitiendo almacenarlos en memoria secundaria en forma de fichero de texto.

La especificación de un patrón está formada por cuatro bloques consecutivos:

- Bloque de definición de nodos: es una lista en la que se enumeran los nodos del patrón, asociando a cada uno las condiciones de encaje con los nodos de una expresión UNL.
- Definición de etiquetas: es una lista en la que se enumeran las etiquetas de los arcos del patrón, asociando a cada una las condiciones de encaje con los tipos de relaciones de una expresión UNL.
- Definición de arcos: lista de los arcos de que consta el patrón.

Debido a que un patrón puede constar solo de nodos, sin que existan relaciones entre ellos (patrón inconexo), solo el primer bloque de los cinco es obligatorio (en este caso, no sería necesario definir ni etiquetas ni arcos).

Por ejemplo, el siguiente patrón²¹:

²¹ Los nodos y arcos se han etiquetado en la figura para facilitar la lectura de su especificación posterior.

Figura 70. Patrón lingüístico de ejemplo (I)

Tiene dos nodos:

- n_1 : solo encaja con palabras universales cuya *headword* sea “open”
- n_2 : puede encajar con cualquier palabra universal

Y una etiqueta:

- r : solo encaja con relaciones UNL de tipo “obj”

Su especificación sería la siguiente:

```

:n1<open>
:n2
.@r(<obj><>)
r(:n1,:n2)
  
```

En la que el bloque de definición de nodos estaría formado por las dos primeras líneas, el bloque de definición de etiquetas sería la tercera línea y el de definición de arcos la cuarta.

Otro ejemplo con condiciones más complejas sería el siguiente:

Figura 71. Patrón lingüístico de ejemplo (II)

Que tendría la siguiente especificación:

```
:n1 (<>.@<past><not>.@<><>.@<(icl:be, occur, do)><>)
:n2:Groningen(iof>city>thing) :
.@r (<><and, or>)
r (:n1, :n2)
```

Donde el significado de las condiciones es el siguiente:

- n_1 : puede encajar con un nodo que:
 - Contenga cualquier palabra universal verbal. Para ello, se especifica que la palabra debe tener cualquiera de las tres restricciones que caracterizan los diferentes tipo de verbos en UNL: icl>occur, icl>be o icl>do.
 - Tenga al menos el atributo “@past” pero no tenga el atributo “@not”.
- n_2 : solo encaja con la palabra universal “Groningen(iof>city>thing)”.
- r : solo encaja con relaciones UNL diferentes a “and” y “or”.

Por último, las restricciones de adyacencia de los nodos condición, se especifican mediante líneas encabezadas por los símbolos “<”, “>”, para indicar las relaciones no permitidas, salientes y entrantes respectivamente, o bien por “<” o “>”, para indicar las relaciones permitidas. Por ejemplo, si se añaden las siguientes restricciones de adyacencia a los nodos del patrón anterior:

Figura 72. Patrón lingüístico de ejemplo (III)

Su especificación sería la siguiente (se señalan en **negrita** las líneas que definen las restricciones de adyacencia):

```

:n1 (<>.@<past><not>.@<><>.@<(iof:be, occur, do)><>)
<<:n1 (obj)
:n2:Groningen(iof>city):
.@r(<><and, or>)
<n2 ()
>n2 ()
r(:n1, :n2)

```

“<<:n1(obj)” indica que, una vez instanciado, el nodo n_1 no puede poseer relaciones salientes de tipo “obj” diferentes a r . Por otro lado, “<n2()” y “>n2()” indican que el conjunto de relaciones permitidas, tanto salientes como entrantes, es vacío.

Especificación del lenguaje en BNF

<patrón> ::= <bloque_def_nodos>

[<bloque_def_etiquetas>]

[<bloque_restricciones_arcos>]

[<bloque_arcos>]

<bloque_def_nodos> ::= <def_nodo> { <def_nodo> }

<def_nodo> ::= ‘:’ <id_nodo> [<condiciones_nodo_basicas>]

[<condiciones_nodo_avanzadas>]

<restricción_uw> ::= ‘(’ <tipo_relacion_unl> ‘:’ [<conjunto_uw>] ‘)’

<jerarquía_kb> ::= ‘<’ <lista_nodos_kb> ‘>’

<lista_nodos_kb> ::= <número> { ‘,’ <número> }

<bloque_def_etiquetas> ::= ‘.@’ <id_rel> [<condiciones_etiquetas>]

<condiciones_etiquetas> ::= ‘(’ <relaciones_incluídas> <relaciones_excluídas> ‘)’

<relaciones_incluídas> ::= ‘<’ [<conjunto_tipos>] ‘>’

<relaciones_excluídas> ::= ‘<’ [<conjunto_tipos>] ‘>’

<conjunto_tipos> ::= <tipo_relación_unl> { ‘,’ <tipo_relación_unl> }

<bloque_restricciones_arcos> ::= { <restricción_arcos> }

<restricción_arcos> ::= (<sentido_arcos_permitidos> | <sentido_arcos_no_permitidos>) ‘:’

<id_nodo> ‘(’ [<conjunto_tipos>] ‘)’

<sentido_arcos_permitidos> ::= ‘<’ | ‘>’

<sentido_arcos_no_permitidos> ::= ‘<<’ | ‘>>’

<bloque_arcos> ::= { <arco> }

<arco> ::= <id_rel> ‘(’ <id_nodo> ‘,’ <id_nodo> ‘)’

Anexo IV. PATRONES LINGÜÍSTICOS

ESPECIFICADOS EN TEXTO

A continuación se muestra parte del fichero de texto que contiene la definición de los patrones lingüísticos utilizados en el método. Este fichero sirve de entrada al sistema en la etapa de reconocimiento de conceptos y entidades. El lenguaje en el que están especificados los patrones se describe en el Anexo III.

```
#####
# Patterns for identifying ENTITIES
#####

# 1. DATE (day - month - year)

:UWDAY_DATE<day>(<>.@<ncl><>.@<><>.@<><>)
<:UWDAY_DATE()
:UWMONTH<month>(<>.@<><>.@<><>.@<><>)
<:UWMONTH()
<:UWMONTH()
:UWYEAR<year>(<>.@<><>.@<><>.@<><>)
<:UWYEAR()
>:UWYEAR()
.@MOD(<mod><>)
.@NAM(<nam><>)
MOD(:UWDAY_DATE, :UWMONTH)
MOD(:UWMONTH, :UWYEAR)
NAM(:UWDAY_DATE, :DAY)
NAM(:UWMONTH, :MONTH)
NAM(:UWYEAR, :YEAR)

# 2. DATE (month - year)

:UWMONTH_DATE<month>(<>.@<ncl><>.@<><>.@<><>)
```

```

<:UWMONTH_DATE()
:UWYEAR<year>(<>.@<><>.@<><>.@<><>)
<:UWYEAR()
>:UWYEAR()
.@MOD(<mod><>)
.@NAM(<nam><>)
MOD(:UWMONTH_DATE, :UWYEAR)
NAM(:UWMONTH_DATE, :MONTH)
NAM(:UWYEAR, :YEAR)

# 3. DATE (year,month,century)

:UWYEAR_DATE<year,month,century>(<>.@<ncl><>.@<><>.@<><>)
<:UWYEAR_DATE()
.@NAM(<nam><>)
NAM(:UWYEAR_DATE, :DAY)

# 4. Explicit UW instance

:NOUN(<>.@<ncl><>.@<><>.@<(iof:)><>)

# 5. Title ("nam")

:NOUN1(<>.@<><>.@<><>.@<(icl:thing)><>)
:NOUN2(<>.@<><>.@<><>.@<(icl:thing)><>)
.@NAM(<nam><>)
NAM(:NOUN1, :NOUN2)

# 6. Scope proper noun

:NOUN(<>.@<concept><>.@<><>.@<><>)

```

```
#####
# Patterns for identifying TERMS
#####

# 1. [NN]

:NOUN(<>.@<ncl><>.@<><>.@<(icl:thing)><>)
<<:NOUN(agt,aoj,ins,met,mod,obj,src)

# 2. [NN] + [NN]

:NOUN1(<>.@<ncl><>.@<><>.@<(icl:thing)><>)
<<:NOUN1(agt,aoj,ins,met,mod,obj,src)
:NOUN2(<>.@<><>.@<><>.@<(icl:thing)><>)
<:NOUN2()
>:NOUN2()
.@R2(<agt,aoj,ins,met,mod,obj,src><>)
R2(:NOUN1, :NOUN2)

# 3. [NN] + [ADJ]

:NOUN(<>.@<ncl><>.@<><>.@<(icl:thing)><>)
<<:NOUN(agt,aoj,ins,met,mod,obj,src)
:ADJECTIVE(<>.@<><>.@<><>.@<(icl:adj)><>)
<:ADJECTIVE()
>:ADJECTIVE()
.@MOD(<mod><>)
MOD(:NOUN, :ADJECTIVE)

# 4. [NN] + [ADJ] + [NN]

:NOUN1(<>.@<ncl><>.@<><>.@<(icl:thing)><>)
<<:NOUN1(agt,aoj,ins,met,mod,obj,src)
:ADJECTIVE(<>.@<><>.@<><>.@<(icl:adj)><>)
<:ADJECTIVE()
>:ADJECTIVE()
:NOUN2(<>.@<><>.@<><>.@<(icl:thing)><>)
<:NOUN2()
>:NOUN2()
.@R(<agt,aoj,ins,met,mod,obj,src><>)
.@MOD(<mod><>)
R(:NOUN1, :NOUN2)
MOD(:NOUN1, :ADJECTIVE)
```

Anexo V. RESULTADOS

A continuación se muestra un extracto de la salida que produce el sistema implementado para el corpus de entrada utilizado en la experimentación.

La salida se presenta en forma de fichero de texto, en el que se enumeran las clases, subclases e instancias ordenadas jerárquicamente según la sangría en el fichero. El esquema es el siguiente:

```
CLASS: A
Instances (n): (n instancias de la clase A)
-   Instancia 1
-   Instancia 2
-   ...
    SUBCLASS: B (B es subclase de A)
    Instances (n): (n instancias de la clase B)
    -   Instancia 1
    -   Instancia 2
    -   ...
    SUBCLASE: C (C es subclase de A)
    Instances (n): (n instancias de la clase C)
    -   Instancia 1
    -   Instancia 2
    -   ...
        SUBCLASE D: (D es subclase de C)
        Instances (n): (n instancias de la clase D)
        -   Instancia 1
        -   ...
```

```

DOCUMENT CLASSES:

*****

CLASS: empire(icl>domain>thing)
*****

Instances (1):
-   Roman_Empire(iof>empire>thing)

*****

CLASS: academy(icl>institution>thing)
*****

Instances (1):
-   mod(academy(icl>institution>thing),Paris(iof>national_capital>thing))
    mod(academy(icl>institution>thing),science(icl>discipline>thing))

*****

CLASS: professor(icl>academician>thing)

Instances (1):
-   professor(icl>academician>thing)

    SUBCLASS: mod(professor(icl>academician>thing),
hydraulics(icl>mechanics>thing, equ>fluid_mechanics))

    SUBCLASS: mod(professor(icl>academician>thing),
mathematics(icl>science>thing))

    SUBCLASS: mod(professor(icl>academician>thing), hygiene(icl>medicine>thing))

    SUBCLASS: mod(professor(icl>academician>thing),
engineering(icl>discipline>thing))

    SUBCLASS: mod(professor(icl>academician>thing),
engineering(icl>discipline>thing))
                                mod(engineering(icl>discipline>thing), civil(icl>adj))
                                mod(engineering(icl>discipline>thing),
mechanical(icl>adj, com>machinery))
*****

*****

```

```

CLASS: prize(icl>gift>thing)
*****

Instances (3):
- Nobel(iof>prize)
- prize(icl>gift>thing)
- Camere(iof>prize)

*****

CLASS: continent(icl>landmass>thing)
*****

Instances (3):
- Africa(iof>continent>thing)
- South_America(iof>continent>thing)
- Europe(iof>continent>thing)

*****

CLASS: entrepreneur(icl>businessperson>thing)
*****

      SUBCLASS: mod(entrepreneur(icl>businessperson>thing),American(icl>adj,
com>America))
      Instances (1):
      - De_Witt_Clinton(iof>person)

*****

CLASS: statesman(icl>politician>thing)
*****

      SUBCLASS: mod(statesman(icl>politician>thing),American(icl>adj,
com>America))
      Instances (1):
      - De_Witt_Clinton(iof>person)

*****

CLASS: reformer(icl>disputant>thing)
*****

      SUBCLASS:
obj(reformer(icl>disputant>thing),social(icl>connected_with_society(icl>adj))
      mod(reformer(icl>disputant>thing),English(icl>adj, com>England))

```

```

Instances (1):
- Edwin_Chadwick(iof>person)

*****

CLASS: South_American_country(icl>thing)
*****

Instances (1):
- Brazil(iof>South_American_country>thing)

*****

CLASS: laboratory(icl>workplace>thing, equ>lab)
*****

Instances (1):
- Cornell(iof>laboratory)

      SUBCLASS: mod(laboratory(icl>workplace>thing, equ>lab),
hydraulics(icl>mechanics>thing, equ>fluid_mechanics))

      SUBCLASS: mod(laboratory(icl>workplace>thing, equ>lab),
hydraulics(icl>mechanics>thing, equ>fluid_mechanics))
                  mod(laboratory(icl>workplace>thing, equ>lab),
European(icl>adj, com>Europe))
*****

*****

CLASS: researcher(icl>scientist>thing, equ>research_worker)
*****

      SUBCLASS: obj(researcher(icl>scientist>thing,
equ>research_worker),hydraulics(icl>mechanics>thing, equ>fluid_mechanics))

Instances (1):
- Hubert_Engels(iof>person)

      SUBCLASS: mod(researcher(icl>scientist>thing,
equ>research_worker),American(icl>adj, com>America))

Instances (1):
- Hunter_Rouse(iof>person)

```

```

*****
CLASS: physician(icl>medical_practitioner>thing, equ>doctor)
*****

      SUBCLASS: mod(physician(icl>medical_practitioner>thing,
equ>doctor),English(icl>adj, com>England))

      Instances (1):
      - Alexander_Houston(iof>person)

*****
CLASS: union(icl>organization>thing)
*****
Instances (1):
-   American_Geophysical_Union(iof>union>thing)

*****
CLASS: secretary(icl>head>thing)
*****

      SUBCLASS:
mod(secretary(icl>head>thing),IAHS(equ>International_Association_of_hydrological
_Sciences, iof>association))
      mod(secretary(icl>head>thing),general(icl>adj))

      Instances (1):
      - Leon_Jean_Tison(iof>person)

*****
CLASS: hill(icl>natural_elevation>thing)
*****
Instances (1):
-   Pentland_hills(iof>hill)

*****
CLASS: place(icl>geographical_area>thing)
*****
Instances (1):
-   Threipmuir(iof>place)

```



```

*****
CLASS: hydrologist(icl>person>thing)
*****

Instances (1):
-   mod(hydrologist(icl>person>thing), Czech(icl>adj, com>Czechoslovakia)

*****

CLASS: politician(icl>leader>thing)
*****

Instances(1):
-   James_Dooge(iof>person)

*****

CLASS: village(icl>settlement>thing)
*****

Instances (2):
-   Parma(iof>village)
-   Grendon_Underwood(iof>village)

*****

CLASS: Asian_country
*****

Instances (1):
-   India(iof>Asian_country>thing)

*****

CLASS: disease(icl>illness>thing)
*****

Instances (4):
-   rinderpest(iof>disease)
-   bubonic_plague(iof>disease)
-   sleeping_sickness(iof>disease)
-   malaria(iof>disease)

```

```

*****
CLASS: biologist(icl>scientist>thing)
*****

SUBCLASS: mod(biologist(icl>scientist>thing),English(icl>adj, com>England))

Instances (1):
- Philip_Monro(iof>person)

*****
CLASS: Canadian_province
*****
Instances (1):
- Alberta(iof>Canadian_province>thing)

*****
CLASS: journal(icl>periodical>thing)
*****
Instances (2):
- Traveaux(iof>journal)
- journal(icl>periodical>thing)

*****
CLASS: African_country
*****
Instances (4):
- South_Africa(iof>African_country>thing)
- Egypt(iof>African_country>thing)
- Guinea(iof>African_country>thing)
- Nigeria(iof>African_country>thing)

*****
CLASS: channel(icl>body_of_water>thing)
*****
Instances (2):
- channel(icl>body_of_water>thing)
- Canal_du_Midi(iof>channel>thing)

```

```

*****
CLASS: lake(icl>body_of_water>thing)
*****

Instances (1):
-   Lake_Erie(iof>lake>thing)


*****
CLASS: principality(icl>domain>thing)
*****

Instances (1):
-   Wales(iof>principality>thing):EI


*****
CLASS: association(icl>organization>thing)
*****

Instances (2):
-   IAHS(equ>International_Association_of_hydrological_Sciences,
iof>association)

-   Northwest_Hydraulics_Consultants(iof>association)


*****
CLASS: clan(icl>social_group>thing, equ>kin)
*****

Instances (1):
-   Rennies(iof>clan)


*****
CLASS: North_American_country
*****

Instances (4):
-   America(iof>North_American_country>thing, equ>United_States)
-   US(iof>North_American_country>thing, equ>United_States)
-   Canada(iof>North_American_country>thing)
-   U.S.(iof>North_American_country>thing, equ>United_States)


*****
CLASS: kingdom(icl>country>thing)

```

```

*****
Instances (2):
- Britain(iof>kingdom>thing, equ>United_Kingdom)
- UK(iof>kingdom>thing, equ>United_Kingdom)

*****

CLASS: county(icl>region>thing)
*****

Instances (2):
- Durham(iof>county)
- county(icl>region>thing)

*****

CLASS: world_war(icl>thing)
*****

Instances (1):
- First_World_War(iof>world_war>thing, equ>World_War_I)

```