Universidad de La Habana
Facultad de Matemática y Computación

Generación Automática de Ontologías

Autor: **José Ariel Romero Costa**

Tutor: **MSc. Juan Pablo Consuegra Ayala**

Trabajo de Diploma
presentado en opción al título de
Licenciado en Ciencia de la Computación

CÓDIGO QR EN UN GRAFO SIMULANDO UN GRAFO DE CONOCIMIENTO PARA QUE QUEDE CON UN USO PARECIDO AL RESULTADO DE LA TESIS :)

2020

– Solo se vive una vez, y ya hice mi elección.
– ¡Error!, solo se muere una vez; vivimos todos los días. Además, no importa cuánto tiempo hayas viajado en la dirección equivocada, siempre puedes cambiar de dirección.

Dedicado a mis padres, Ramón y Amparo, quienes han batallado a mi lado incansablemente y de forma incondicional durante estos 23 *años de vida y* 18 *años de estudio.*
A Julio, mi hermano, quien siempre ha estado certero con sus consejos y apoyándome en todo lo que he necesitado fuera y dentro del ámbito educacional.
A Odalmis, mi novia, prometida y próximamente esposa, hemos cursado los buenos y malos momentos en en estos últimos 4 *años y ha estado siempre presente para dar un consejo o apoyo cuando lo necesito.*
A todo aquel que, sin importar si fue un simple "buenos días", una charla pasajera o esporádica o una larga amistad de años, intervino en mi vida para hacer de mí lo que soy hoy.

Agradecimientos

Considero fuertemente que, si un día no hubiera conocido a una persona en específico o algo “tan sencillo” como no haber saludado a alguien o no haber visto una noticia o información, hoy, quizás, no sería quien soy. Por este motivo aprovecho la investigación que aquí se presenta para agradecer a todas las personas que me han llevado a ser quien soy hoy. Hayamos compartido momentos buenos, malos o un simple gesto de saludo. Gracias a ellos siempre quedarán las marcas que me hicieron llegar a este punto, forjando mi carácter, físico y cualidades buenas y malas.

Algunos de ellos me acompañan desde que nací, por ejemplo, mis padres Ramón y Amparo, que me han dedicado gran parte de su vida y por lo que estoy eternamente agradecido. No tengo manera de pagar todo lo que han hecho por mí, más allá de que con cada paso que doy intento que se sientan más orgullosos de mí. Mi hermano Julio y su esposa Judith, mi abuela Rosa, a la cual cariñosamente desde pequeño le apodé Ia y así la conocen hoy día, a mis tíos Teté y Alberto, mis primos Mayté y Albertico, mis padrinos María Eugenia y Manero. En fin, sin ánimos de extenderme tanto agradezco a modo general a mí familia, sin importar dónde vivan, por haberme apoyado y ayudado en las decisiones que tomé y al mismo tiempo, ofrecer consejos y su amor incondicional.

Otros aparecieron en mi vida, pero lo importante no es cuándo lo hayan hecho, sino los aportes que en ella hicieron y que hayan llegado para quedarse. Agradezco a mi novia Odalmis por todo el apoyo y amor que me ha dado, a Ismael, gran amigo que me ayudó durante los cinco años de universidad en los viajes de Matanzas a La Habana y viceversa para que pudiera estudiar y alimentarme, e incluso, me dio varios consejos como si fuera mi segundo padre. A Mayte y Jorge, junto a Fany y Jorge Carlos mis vecinos, quienes en estos últimos dos años han sido como mis padres y hermanos adoptivos en La Habana. A Carlos mi padrino de bodas, del cual estoy muy agradecido de haber tenido la posibilidad de conocer y

entablar una buena amistad. Nardo, quien nos ha ayudado a mí y a mi familia a lo largo de todos estos años.

Otras personas han quedado en el pasado, pero no porque nos hayamos distanciado emocionalmente, sino físicamente, las distintas situaciones en las que nos puso la vida nos llevó a dejar de vernos, pero son personas que cada vez que hablamos o nos vemos recordamos los momentos compartidos con gran alegría y vivimos nuevos para recordarlos en la próxima ocasión. Ellos son Andy, mi amigo de la infancia, crecimos juntos y fue como un hermano para mí. Arla, su abuela, la cual también me cuidó y enseñó como si fuera su nieto. Yusmeidys, amiga de la vocacional, vivimos muy buenos momentos juntos y entablamos una bonita amistad, si de algo es culpable, es de haber alimentado mi "bichito interior" con los deseos de estudiar medicina, pero lo logró tarde, una vez comenzado a cursar esta carrera, aunque por ese motivo trabajaré duro en el futuro para dedicarme a la medicina computacional. A Yudisleydis, mi profesora de la primaria, que me inculcó los primeros pasos en el mundo del estudio, me enseño a leer, escribir y calcular, y fue de las primeras personas que incentivaron las matemáticas en mí.

Por último, pero no menos importante, dos seres que a pesar de que no pueden hablar, no hacen más que expresar sus sentimientos y amor hacia mí: mis hijos caninos Gema y Ody, los cuales me acompañan desde hace 4 y 3 años respectivamente. En más de una ocasión han sabido alegrarme el día en una situación donde me sentía triste y estoy más que agradecido y orgulloso por como son.

Quisiera darle mención a todos pero no puedo, son muchos los que han aparecido y estado presente en mi vida. Espero que sigamos relacionándonos en el futuro, junto con las nuevas personas que conoceré.

Gracias a todo aquel que aportó un granito de arena en mi formación y que creyó en mí; incluso en los momentos en que ni yo mismo creía.

José Ariel Romero Costa
Facultad de Matemática y Computación
Universidad de La Habana

Opinión del tutor

La representación de conocimiento se ha convertido en un área de investigación muy activa en los últimos años, motivada tanto por la disponibilidad masiva de nuevos recursos, como por la necesidad de hacer computacionalmente tratable el volumen de datos producidos diariamente. Su relevancia en tareas más amplias, como el descubrimiento automático de conocimiento, la vuelven un área crucial para el desarrollo de varios sectores de la sociedad. En el dominio médico, la aplicación de estas técnicas se vuelve especialmente interesante, ya que procedimientos de inferencia sobre una base de conocimiento puede potencialmente ayudar a diseñar nuevos tratamientos para combatir enfermedades aún no resueltas. En este marco se desarrolla la tesis de licenciatura de José Ariel Romero Costa, con quien pude trabajar este último año en el diseño y validación de un algoritmo para la construcción de ontologías a partir de textos anotados. Esta tesis da continuidad a una línea de investigación que se ha venido desarrollando en la facultad en los últimos años ligada al descubrimiento de conocimiento.

La propuesta de José consiste en un algoritmo para la creación automática de ontologías a partir de una colección anotada de documentos. El sistema utiliza el esquema de anotación del *eHealth-KD Challenge* que ha sido empleado en dos competencias internacionales de extracción de conocimiento, en el marco de los eventos *IberLEF 2019* e *IberLEF 2020*. El trabajo conllevó reconstruir un corpus de texto de Medline sobre el que identificar y reordenar las oraciones del corpus anotado. A partir de las entidades y relaciones señaladas en el texto, se realiza un proceso de normalización con el objetivo de unificar aquellas entidades que difieren sintácticamente pero comparten la misma semántica. La tesis presenta un procedimiento para organizar la información recogida en múltiples oraciones, formando una base de conocimiento que integra las distintas instancias de anotaciones mencionadas entre colecciones. La representación final obtenida constituye un paso de avance en la formalización del esquema de anotación, y sienta las bases

para futuros procesos de inferencia.

Durante el desarrollo de la tesis José demostró independencia y creatividad para lidiar con los problemas encontrados. Tuvo que dominar conceptos y tecnologías del estado del arte, con muchas de las cuales no tuvo contacto durante la carrera. Los problemas que hubo de resolver le servirán de aprendizaje para su desarrollo futuro. El proceso de investigación e implementación desarrollado por José queda recogido en un documento de tesis que avala la capacidad adquirida para presentar resultados de investigación de forma concisa y coherente. Todo esto lo han realizado a la par de las actividades docentes, como estudiante de pregrado y como alumno ayudante de la asignatura *Programación*, donde ha sabido asumir con éxito todas las responsabilidades y retos.

José ha sido alumno ayudante desde su tercer año en la carrera, tiempo que pude compartir con él directamente en clases y en las reuniones del colectivo. En esos años he podido comprobar su interés y dedicación por la asignatura y otros temas relacionados. Este último ejercicio demuestra que ya ha adquirido la madurez necesaria para desarrollar proyectos de alta complejidad con calidad y esmero. Como tutor, estoy complacido por los resultados obtenidos, y por el trabajo realizado con José, que aunque no estuvo exento de obstáculos, logró superar los desafíos. Por estos motivos estoy convencido de que José será un excelente profesional de la Ciencia de la Computación.

MSc. Juan Pablo Consuegra Ayala
Facultad de Matemática y Computación
Universidad de La Habana

Resumen

En los últimos años se ha evidenciado un aumento en el desarrollo de técnicas para descubrir conocimiento de forma automática en documentos escritos en lenguaje natural. El procesamiento automático va aparejado a la posibilidad de analizar colecciones de información con disímiles textos. En el área de la medicina, el auge de estas tecnologías es especialmente significativo, pues permite favorecerse de la gran cantidad de información disponible para el avance de este campo, que posee gran importancia para la sociedad. Por otra parte, estas técnicas suelen apoyarse en corpus anotados, los cuales son recursos escasos. Esto se vuelve crítico en el idioma español, donde la cantidad existente es ínfima y de dominio menos generalizado.

En este estudio se define un modelo de anotación de propósito general con el objetivo de capturar los rasgos semánticos más relevantes en los documentos de texto. Además, se presenta un esquema de ontología que se usará para la extracción de conocimiento de forma automática. También se ofrecen los pasos a seguir para la implementación de un algoritmo computacional que busca representar un corpus anotado como un grafo de conocimiento, siguiendo las reglas definidas por la propia ontología. Por último, se muestran las tareas realizadas para la validación de las propuestas dadas, así como resultados en términos matemáticos.

Los resultados alcanzados demuestran que el descubrimiento de conocimiento constituye un campo de investigación activo, donde pueden aplicarse técnicas de aprendizaje automático logrando resultados positivos. Se propone la verificación y comparación de un grafo de conocimiento específico creado a partir de las propuestas brindadas en este estudio respecto a la capacidad de aprendizaje e interpretación de un grupo de expertos en el mismo tema. Además se ofrece la continuación de esta línea de investigación con el objetivo de mejorar la efectividad de las propuestas dadas y su aplicación en otros dominios.

Abstract

In recent years there has been an increase in the development of techniques for automatic knowledge discovery from documents written on natural language. Automatic processing provides the possibility to analyze collections of information containing a large number of texts. In the medical field, the rise of these technologies is significantly special, because it allows taking advantage of the huge amount of data available for it and improve research on this area, which is really important to society. These techniques tend to rely on annotated corpus, and they are a scarce resource. This becomes a critical fact in Spanish language, where the existing amount of them is very low and from a less general domain.

In this study, a general-purpose annotation model is defined to capture the most relevant semantic features contained in text documents. Also, an ontology scheme is presented and used for automatic knowledge extraction. A theoretical step by step implementation of a computational algorithm aiming to build a knowledge graph from an annotated corpus and following the rules of the previously defined ontology is also proposed. Finally, the evaluation and validation process is exposed, as well as statistics results.

The results achieved demonstrate that knowledge discovery constitutes an active research field, where machine learning techniques can be applied achieving positive results. The verification and comparison of a specific knowledge graph built from the proposals provided in this investigation against the learning and interpretation skills of a group of experts on the same field is proposed. Also, the continuation of this research line is suggested, aiming to improve the effectiveness of the proposals given and their application in other domains.

Índice general

Índice de figuras

Índice de tablas

Introducción

El presunto desarrollo tecnológico se ha exacerbado con el advenimiento cada vez mayor del uso del Internet y otros medios avanzados y efectivos que garantizan un mejor futuro para cuestiones importantes de la vida. Debido al continuo aumento del flujo informativo, se hace cada vez más necesaria la utilización de herramientas que permitan identificar, capturar y representar el conocimiento dentro de los sistemas de información, ya sea de dominio específico o de propósito general.

Para ello, ciencias avanzadas como la Ciencia de la Computación y de la Comunicación comprenden la ontología como esa vía formal de tipos, propiedades, y relaciones entre entidades que existen y están definidas en el dominio de trabajo. Estas son creadas para limitar la complejidad de cualquier tema y para organizar la información. Es una medida eficaz en las soluciones de problemas comunes en la vida diaria, que debido a la sobreinformación no se podrían llevar a cabo de forma manual.

Otro de sus beneficios es el hecho de que permiten crear entendimiento compartido al unificar los diferentes puntos de vista. Esto sirve para facilitar la comunicación entre los actores implicados en la construcción de sistemas de información referidos al tema en cuestión. Además, permiten el reuso del conocimiento del dominio, pues sirve de base ya creada para posteriores investigaciones. También facilitan la recuperación, integración e interoperatividad entre fuentes de conocimiento heterogéneas. Se provee una base para la representación del conocimiento del dominio y ayudar a identificar las categorías semánticas del mismo. [23]

Surge entonces el creciente interés de estudiar técnicas para el descubrimiento automático de conocimiento. El procesamiento automático trae consigo la posibilidad de analizar colecciones masivas de información. Sin embargo, la mayor parte de estas colecciones almacenan la información disponible en documentos

textuales escritos en lenguaje natural. La naturaleza en que se expresa la información y su estructura semántica poco unificada se vuelven la principal fuente de retos de dicho procesamiento. Entre los retos destacan la ambigüedad del lenguaje natural y la gran cantidad de idiomas en que puede estar escrito.

En la actualidad, las ontologías se están aplicando en áreas heterogéneas. Aunque quizás se las conoce más por su papel en el desarrollo de nuevos servicios en la web, basados en la descripción del significado de los contenidos de las sedes o portales de Internet (web semántica), también se están utilizando para el desarrollo de mecanismos que faciliten la comunicación entre las personas y las máquinas por medio del lenguaje natural. [8]

En el contexto de la salud y la medicina las ontologías adquieren particular interés, debido a que se están utilizando cada vez más para la solución de disímiles tareas, como la recuperación de información y la búsqueda de respuestas en fragmentos de texto que resuelven preguntas. Diariamente se publican muchos artículos médicos y se hace imposible acceder a todos y mantener un absoluto control sobre las novedades médicas y las herramientas que se van desarrollando para solucionar las enfermedades o los problemas de la salud de manera general.

La extracción automática de conocimiento proveería de una herramienta para asistir el desarrollo en esta área a partir de la normalización e integración de los resultados encontrados. Una vez extraído y representado el conocimiento computacionalmente, procesos de inferencia permitirían el descubrimiento de nuevo conocimiento. Ejemplo de esto es el constante descubrimiento de nuevas interacciones entre medicamentos, proteínas y genes. Un sistema de descubrimiento de conocimiento posibilitaría la detección automática de nuevas relaciones entre ellos, y por ende, el descubrimiento de nuevas causas de enfermedades, síntomas y tratamientos.

Problemática

En disímiles ocaciones, es necesaria la lectura de un amplio grupo de documentos con gran cantidad de páginas para poder tener conocimiento acerca de algún tema. Incluso algunas veces la información leída no es relevante para lo que se desea, y por tanto, fue una inversión de tiempo en vano. Las ontologías, por otra parte, aceleran en gran medida este proceso, pues el análisis y representación de uno o más corpus en un grafo de conocimiento es cuestión de segundos, y esto posibilita posteriormente, buscar la información deseada a través de consultas realizadas a un sistema computacional.

Para diseñar una ontología no existe una única forma o metodología correcta a emplear y tampoco es objetivo de esta investigación definir una. En aras de cumplir con los objetivos propuestos, se realizaron varios estudios para lograr el diseño de una ontología de propósito general. Una vez definida, se hicieron múltiples acercamientos al problema de la creación del grafo de conocimiento. Un claro error fue representar a través de aristas las relaciones del corpus solo con las entidades o fragmentos de texto implicados en ellas. Esto provocó la necesidad de profundizar en el estudio del problema desde otros enfoques y obligando a crear nodos más complejos, los cuales representan las relaciones construidas entre otros más simples. Al mismo tiempo, fue necesario implementar un algoritmo en orden topológico para completar satisfactoriamente esta tarea.

La resolución de estos problemas devino en la construcción de un grafo de conocimiento partiendo de un corpus que consiste en un subconjunto de los artículos en idioma español de *Medline* [15], y anotado basándose en la propuesta mostrada en esta investigación. Esto permitió avanzar hacia métodos de descubrimiento de conocimiento que se encontraba implícito en estos artículos, y se alcanzó al correlacionarlos por medio de la ontología sugerida en la misma.

Objetivos

La investigación se plantea como objetivo general crear una ontología para representar el conocimiento descrito en un corpus anotado, a través de la implementación computacional de un grafo de conocimiento y a la misma vez que sea generalizable a múltiples dominios.

Se proponen los siguientes objetivos específicos:

- Estudiar los esquemas de anotación y corpus usados en diversas tareas de extracción del conocimiento.
- Definir un esquema conceptual de anotación para la representación de los rasgos semánticos más relevantes en textos escritos en lenguaje natural.
- Definir un formato de anotación de archivos para el esquema conceptual previamente definido.
- Diseñar una propuesta de ontología donde se pueda representar un corpus de documentos escritos en lenguaje natural.
- Implementar un algoritmo computacional para representar un corpus anotado como grafo de conocimiento a través de dicha ontología.
- Implementar un marco experimental para la evaluación de la propuesta de solución.

Organización de la tesis

El contenido de la tesis se organiza de la siguiente forma. El capítulo 1 introduce los principales conceptos relacionados con las ontologías y la extracción y representación de conocimiento. En este capítulo, además, se analizan los principales corpus y representaciones semánticas existentes en la literatura. El capítulo 2 describe un modelo de anotación de propósito general que busca capturar los rasgos semánticos más importantes en documentos de texto. En el capítulo 3 se presenta una propuesta para la creación de un grafo de conocimiento a través de un corpus anotado. En el capítulo 4 se muestran los resultados alcanzados en esta investigación, y en función de estos, se discute la efectividad de cada uno de los elementos propuestos en la misma. La investigación finaliza presentando las conclusiones pertinentes y las recomendaciones para su continuidad y mejora.

Capítulo 1

Generación Automática de Ontologías

En los últimos años, el desarrollo de especificaciones formales explícitas de los términos en el dominio y las relaciones entre ellos [7] en una ontología ha dejado de tener lugar en laboratorios o departamentos de inteligencia artificial para pasar a tener un rol protagónico en los escritorios de expertos en el tema.

Las ontologías se han vuelto común en el espacio de la *red mundial* (conocida en inglés como *world wide web*). En esta red ellas abarcan un gran espectro de campos, desde grandes taxonomías que categorizan sitios web, como sucede en Yahoo, hasta categorizaciones de productos a la venta y sus características, como sucede en Amazon.

El *World Wide Web Consortium* (W3C) desarrolló el *Resource Description Framework* [4] (RDF, traducido al español como *Marco de Descripción de Recursos*). El mismo, es un lenguaje para codificar el conocimiento en las páginas web y que sea comprensible para los usuarios que buscan esa información.

La *Agencia de Proyectos de Investigación Avanzada de Defensa* (del inglés *Defense Advanced Research Projects Agency*, DARPA), en conjunto con W3C, desarrollaron el *DARPA Agent Markup Language* (DAML, traducido al español como *Lenguaje de Marcado de DARPA para Agentes*), el cual es una extensión de RDF con construcciones más expresivas destinadas a facilitar la interacción de los agentes en la web. [10]

Muchas disciplinas desarrollan ontologías estandarizadas que los expertos en el dominio pueden usar para compartir y anotar información en sus campos. La

medicina, por ejemplo, ha producido vocabularios estructurados, estandarizados y extensos como *SNOMED* [24] y la red semántica del *Unified Medical Language System* [11] (traducido al español como *Sistema de Lenguaje Médico Unificado*).

También están surgiendo amplias ontologías de propósito general. Por ejemplo, *United Nations Development Program* (traducido al español como *Programa de las Naciones Unidas para el Desarrollo*) y *Dun & Bradstreet* juntaron sus esfuerzos para desarrollar la ontología *United Nations Standard Products and Services Code* [28] (UNSPSC, traducido al español como *Código Estándar de Productos y Servicios de las Naciones Unidas*) que proporciona terminología para productos y servicios.

Una ontología define un vocabulario común para los investigadores que necesitan compartir información en un determinado dominio, y a la misma vez, facilita la búsqueda y comprensión de esta información por personas no expertas en el tema. Esto incluye definiciones computacionalmente interpretables de conceptos básicos y relaciones entre ellos pertenecientes al dominio. Hay una interrogante que cabe preguntarse, ¿por qué alguien quisiera desarrollar una ontología? Algunas de las razones principales son:

- Compartir conocimiento de la estructura de la información entre investigadores y/o usuarios.
- Permitir la reutilización del conocimiento del dominio.
- Hacer explícitas las suposiciones o conocimientos a priori del dominio.
- Separar el conocimiento explícito del dominio del conocimiento implícito operacional.
- Analizar el conocimiento del dominio.

Compartir conocimiento de la estructura de la información entre investigadores y/o usuarios es uno de los objetivos comunes en el desarrollo de ontologías [16, 7]. Por ejemplo, si varios sitios web diferentes entre sí contienen información médica o proporcionan servicios médicos de comercio electrónico, y estos comparten y publican la misma ontología subyacente de los términos que utilizan, los agentes informáticos pueden extraer y agregar información de los mismos. Además, estos últimos pudieran utilizar dicha información para responder consultas de los usuarios o como datos de entrada para otras aplicaciones.

Permitir la reutilización del conocimiento del dominio fue una de las fuerzas impulsoras detrás del reciente aumento de la investigación ontológica. Por ejemplo, los modelos para muchas áreas diferentes deben representar la idea de tiempo. Esta representación incluye, entre otros, las nociones de intervalos, puntos y medidas relativas a este. Si un grupo de investigadores desarrolla tal ontología en detalle, otros pueden simplemente reutilizarla para sus dominios. Además, si se necesita construir una grande, se pueden integrar varias ya existentes que describan partes específicas de la rama deseada. También se puede reutilizar una de propósito general, como UNSPSC, y extenderla para describir el área de interés.

Hacer explícitas las suposiciones o conocimientos a priori del dominio hace posible cambiarlas fácilmente si cambian las ideas tenidas de antemano en este tema. Los supuestos de *codificación rígida* (del inglés *hard-coding*) sobre el mundo hechos en lenguajes de programación hacen que estas no solo sean difíciles de encontrar y comprender, sino también de cambiar, en particular para alguien sin experiencia en el ámbito computacional. Además, las especificaciones explícitas del conocimiento del dominio son útiles para los nuevos usuarios que deben aprender qué significan los términos de este.

Separar el conocimiento explícito del dominio del conocimiento implícito operacional es otro uso común de las ontologías. Se puede describir la tarea de configurar un producto a partir de sus componentes, de acuerdo con una especificación requerida e implementar un programa que realice esta configuración independientemente del producto. [14] Seguido de esto, se puede desarrollar una ontología de componentes y características de los ordenadores personales y aplicar el algoritmo para configurar uno de ellos a medida. También este último puede usarse para realizar la misma tarea en ascensores, si se le "alimenta" con una ontología de los elementos de estos. [26]

Analizar el conocimiento del dominio es posible una vez que se dispone de una especificación declarativa de los términos. El análisis formal de estos es extremadamente valioso cuando se intenta reutilizar ontologías existentes y ampliarlas. [13]

Capítulo 2

Modelo de Anotación

El primer paso para el algoritmo presentado en este trabajo, es tener un corpus[1] anotado basado en el esquema presentado a continuación. Además, en este capítulo son analizadas algunas estadísticas de un corpus de oraciones del dominio médico en idioma español, el cual es utilizado en la presente investigación y son mostradas las herramientas empleadas para construirlo y trabajar con el mismo.

2.1. Esquema de anotación

El modelo de anotación de propósito general empleado busca capturar los rasgos y relaciones semánticas más relevantes presentes en oraciones del lenguaje natural. Este debe evitar ambigüedades tanto como sea posible, de forma que anotadores humanos distintos tengan una alta probabilidad de coincidir. A la misma vez, necesita ser lo suficientemente expresivo para representar los conceptos relevantes del dominio y sus interacciones. Además, debe ser capaz de construir conceptos complejos a partir de combinar otros más simples mediante el uso de un conjunto reducido de reglas. También está diseñado para asistir en desarrollo de sistemas de descubrimiento de conocimiento. Por este motivo es necesario independizar la representación del modelo de la estructura gramatical de las oraciones, y en su lugar, tratar de representar el significado semántico.

Este modelo de anotación se basa en las tripletas *Subject-Action-Target* (en es-

[1]Un corpus es un conjunto de oraciones y/o documentos de ejemplos reales usados en el lenguaje natural.

pañol *Sujeto-Acción-Objetivo*) y en la estructura gramatical *sujeto-verbo-objeto*, normalmente expresada con su abreviatura **SVO**. Tiende a ser el orden predeterminado porque el verbo se usa para dividir el sujeto del predicado, sin necesidad de usar partículas para indicar dónde empiezan o terminan los mismos. Es, por ende, una de las secuencias más frecuente en el lenguaje natural y de hecho es usada en la mayoría de lenguas occidentales y un buen número de orientales.

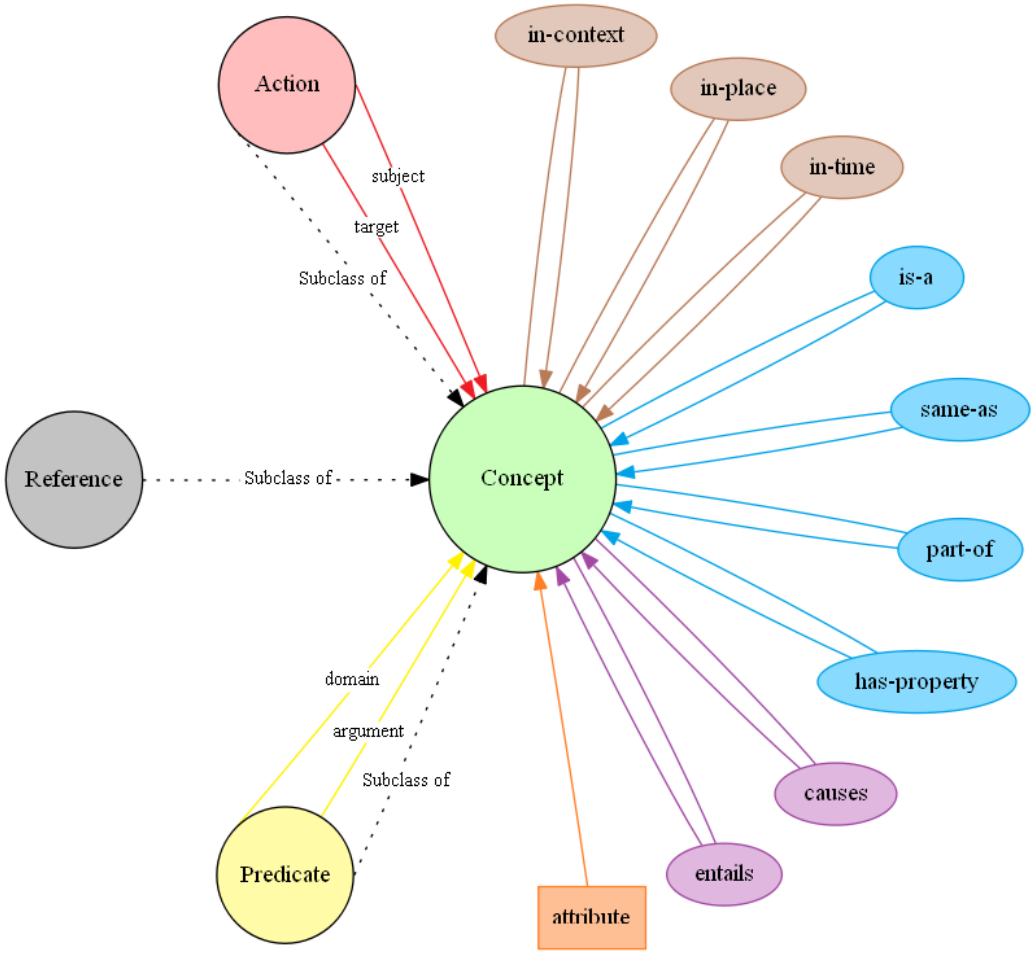


Figura 2.1: Esquema conceptual del modelo de anotación.

Es válido destacar que al estar interesados en fragmentos de conocimiento, el rol semántico de las entidades anotadas puede no coincidir con su rol gramatical. Los roles semánticos fundamentales de este modelo son *Concept* y *Action*

(en español *Concepto* y *Acción* respectivamente), siendo usados para representar información objetiva acerca de lo que se está haciendo, por quién, y a quién. Estas estructuras pueden ser contextualizadas en tiempo, lugar y otras circunstancias generales.

Existen otros 2 roles semánticos, llamados *Predicate* y *Reference* (en español *Predicado* y *Referencia* respectivamente). *Predicate* es utilizado para construir conceptos más complejos a partir de otros más simples. *Reference* define un término del que se menciona un hecho, pero en el contexto de la oración no está escrito explícitamente, por lo que la información semántica de este rol no está contenida en las anotaciones.

Por último, son usadas seis relaciones con semántica específica para representar conocimiento de propósito general. Las relaciones *is-a*, *part-of*, *same-as* y *has-property* (en español *es-un*, *parte-de*, *igual-que* y *tiene-propiedad* respectivamente) son tomadas de representaciones ontológicas y taxonómicas, mientras que *causes* y *entails* (en español *causa* e *implica* respectivamente) se toman del dominio de la comprensión del texto. Además, las relaciones *in-time*, *in-place* e *in-context* (en español *en-tiempo*, *en-lugar* y *en-contexto* respectivamente) son usadas para dar contexto y cuatro atributos booleanos son asociados a los conceptos. Las próximas secciones explican cada rol semántico y las relaciones detalladamente, incluyendo ejemplos de su uso en oraciones del lenguaje natural.

La figura 2.1 muestra una representación gráfica del modelo de anotación. En el esquema conceptual se puede apreciar que cada uno de los roles semánticos definidos en el modelo de anotación está representado por un círculo. Además, las posibles relaciones definidas entre cada pareja de roles se representan con óvalos y con un rectángulo los atributos que pueden tener los mismos. En color café están representadas las relaciones de contexto, en azul las taxonómicas y en violeta las de causalidad e implicación.

2.1.1. Conceptos

El rol *Concept* es usado para anotar fragmentos de texto que representan una unidad atómica de información en el dominio. Puede ser una entidad nombrada, un sustantivo, adjetivo o verbo, que representa un concepto relevante en el dominio del texto. Por ende, la gran mayoría de palabras o frases que expresan un significado propio es anotado de esta manera (o uno de sus derivados, como se explica más adelante). Palabras tales como artículos, preposiciones y conjuncio-

nes, las cuales solo realizan una función gramatical y sin significado semántico, no son anotados.

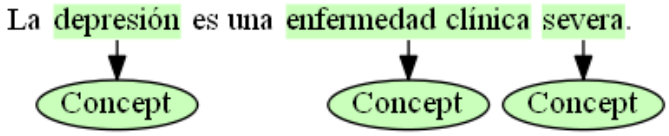


Figura 2.2: Ejemplo de anotación de conceptos.

En la figura 2.2 se distinguen claramente como conceptos en el dominio médico las palabras «`depresión`» y «`severa`», cuyo significado de cada uno de ellos es independiente del rol gramatical que tengan en la oración. Algunos conceptos, como «`enfermedad clínica`» en este caso, se componen de múltiples palabras, ya sea porque de manera independiente no tienen relevancia, o porque al unirlos cobra un significado diferente al de sus componentes individuales. En esta ocasión, a pesar de que «`enfermedad`» y «`clínica`» poseen una connotación bien definida por sí mismas, el concepto «`enfermedad clínica`» tiene gran importancia en el dominio médico, lo cual lo hace una unidad única de información, es decir, un especialista en este campo puede identificarla claramente. Las palabras que conforman un concepto no tienen que estar consecutivas en el texto, pero sí son seleccionadas de izquierda a derecha.

2.1.2. Acciones

El rol *Action* es un tipo particular de *Concept* que indica una acción o evento que otro concepto puede realizar o ser objetivo de ella. Un *Action* puede ser enlazado con otros conceptos relevantes a partir de 2 roles semánticos: *subject* y *target* (en español *sujeto* y *objetivo* respectivamente). El *subject* es el que produce la acción, mientras que el *target* es el que recibe los efectos o el objetivo de la acción.

En la figura 2.3 la acción es indicada por una palabra con el rol gramatical de verbo. Intuitivamente este es el caso más común, sin embargo, una acción puede ser indicada además por una palabra con otro rol gramatical, como los sustantivos. Por ejemplo, en la frase "... *el empeoramiento de los síntomas* ...", la palabra «`empeoramiento`» se considera también un *Action* a pesar de que no es un verbo, dado que describe un proceso o evento que ocurre sobre otros conceptos.

Por tanto, el rol semántico *Action* describe el significado de un concepto en el

dominio semántico, en lugar de su función gramatical en una oración específica. Si un concepto del dominio expresa un proceso o evento que realiza otro concepto o produce un efecto sobre otro(s), entonces es un *Action*, incluso si puede ser usado con una función gramatical distinta.

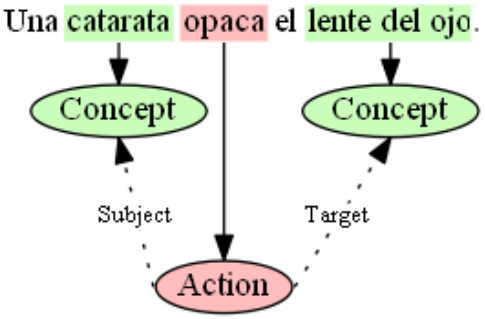


Figura 2.3: Ejemplo de anotación de acción.

2.1.3. Referencias

El rol *Reference* es un tipo de *Concept* que no tiene un significado semántico específico, pero que es necesario por razones gramaticales. Es usado para anotar pronombres (por ejemplo, *este*, *aquel*) y demás elementos que hacen referencia a otro *Concept* presente en la oración, documento y/o corpus. En la figura 2.4 puede verse un ejemplo.

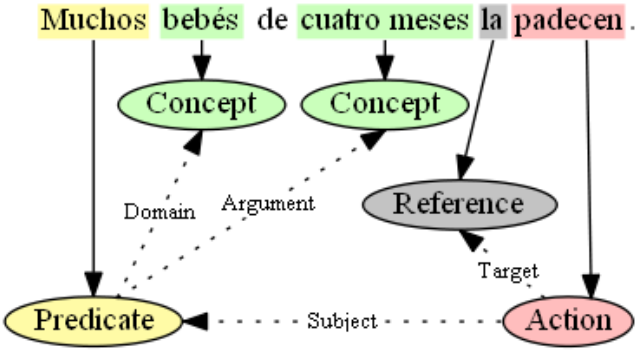


Figura 2.4: Ejemplo de anotación de referencia y predicado.

2.1.4. Predicados

El rol *Predicate* es usado para formar conceptos más complejos a partir de aplicar un determinado criterio sobre otros en una oración. Un caso de uso común es para definir el subconjunto perteneciente a un concepto y que cumple determinadas propiedades.

Por ejemplo, en la figura 2.4, la palabra *muchos* cumple la función de filtrar algunos de los bebés, por eso es anotada como *Predicate*.

De conjunto con esta relación, cualquier concepto puede jugar dos roles adicionales: *domain* y *argument* (en español *dominio* y *argumento* respectivamente), completando así su significado. El *Predicate* define el conjunto de objetos pertenecientes al dominio (el concepto enlazado con el rol *domain*) que cumplen el predicado anotado según los argumentos señalados (el o los conceptos anotados con el rol *argument*).

De forma matemática, la relación *Predicate* define al conjunto:

$$\{x \in Domain \mid Predicate(x,\ arg_1,\ arg_2, \ldots,\ arg_n)\}$$

En el ejemplo de la figura 2.4, el dominio de este *Predicate* es representado por el *Concept* «`bebés`», y el único argumento es «`cuatro meses`». Esta construcción da lugar a un nuevo concepto, el de «`muchos bebés de 4 meses`», el cual puede ser entendido como la aplicación del filtro «`muchos`» sobre el conjunto de elementos definido por el *Concept* «`bebés`», de los cuales son seleccionados aquellos con el argumento «`cuatro meses`».

$$\{x \in Bebés \mid muchos(x,\ cuatro\ meses)\}$$

El nuevo concepto complejo construido de esta forma es representado en la oración por la anotación *Predicate* en sí misma. Por tanto, para continuar con el ejemplo anterior, en caso de querer que estos «`muchos bebés`» jugaran el rol *subject* o *target*, la anotación correspondiente debe ir desde un *Action* hacia el *Predicate*, como se muestra en la figura 2.4. Es un error anotar que el *subject* de «`padecen`» es «`bebés`» porque este concepto representa «`todos los bebés`». Por ende, el *Predicate* es usado para representar el concepto filtrado en sí, no el operador de filtrado.

Como caso de uso particular de esta anotación, se encuentra el caso en que un término no representa un concepto relevante por sí mismo (por tanto no debe ser

anotado como *Concept*), sino que denota una propiedad o rasgo medible de otro concepto. Por ejemplo, «`tipo`», «`parte`», «`nivel`» y «`cada`» en "*tipo de cáncer*", "*parte del cuerpo*", "*nivel de glucosa*" y "*cada trimestre*" respectivamente. En tales casos, el *Predicate* debe carecer de alguno de los roles *domain* o *argument*. Si el tipo o clase resultante de formar el predicado coincide con el del concepto a enlazar, entonces el rol utilizado es *domain*. En otro caso, se enlaza al concepto con el rol *argument*.

2.1.5. Componiendo conceptos

Así como un *Predicate* puede utilizarse para componer conceptos, se puede lograr un resultado similar al considerar un *Action* como el *subject* o *target* de otro.

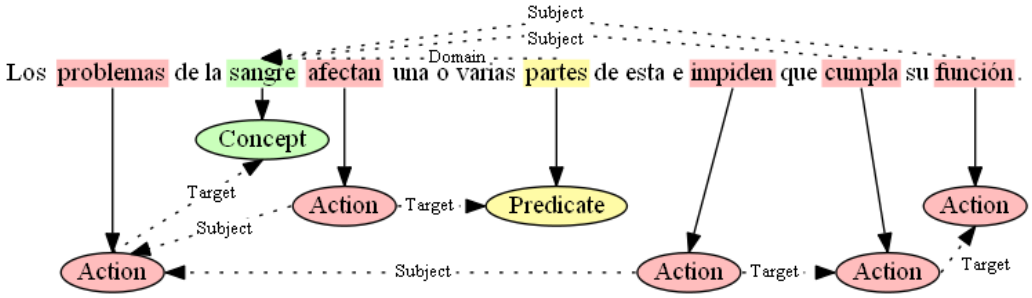


Figura 2.5: Ejemplo de anotación de conceptos compuestos.

Por ejemplo, en la figura 2.5, hay un concepto complejo formado con las palabras «`problemas`» y «`sangre`». Este a su vez, actúa como *subject* de «`afectan`», dado que no todos los «`problemas`» se «`afectan`», sino solo aquellos que son «`problemas de la sangre`». Por otro lado, la propia palabra «`sangre`» actúa como *domain* del predicado «`partes`», el cual es el *target* de «`afectan`». De manera similar sucede con los otros tres conceptos complejos «`impiden`», «cumpla» y «`función`».

De esta forma puede apreciarse que la construcción y/o anotación de conceptos complejos es una tarea compleja en sí. Además, esta estrategia puede ser usada para representar la nominalización de un verbo, pues al anotar el *Action* y los correspondientes *subject* y *target* se construye el concepto complejo.

2.1.6. Relaciones taxonómicas

Los roles *Action* y *Concept* permiten capturar gran parte del significado semántico de una oración a partir de anotar como acción todos los conceptos que indican alguna interacción entre ellos. Sin embargo, algunos tipos específicos de interacciones son tan comunes que son considerados en diferentes dominios del conocimiento como los bloques constructores para las representaciones ontológicas y taxonómicas. Tal es el caso de las parejas de hiperonimia/hiponimia, anotadas como relaciones *is-a* (en español *es-un*) y meronimia/holonimia, anotadas como relaciones *part-of* (en español *parte-de*), que forman el centro de muchas bases de conocimiento.

Estos dos tipos de relaciones son muy comunes en la mayoría de los dominios del conocimiento, y hay muchas formas distintas para expresar estas ideas en texto. Debido a ello, resulta mejor representarlas explícitamente como relaciones entre conceptos, en lugar de recurrir a anotar como *Action* las formas del verbo ser o estar. Además, una anotación explícita de estas relaciones permite que sistemas de descubrimiento de conocimiento entrenados en estas anotaciones extraigan estructuras más compactas y concisas, dado que no es necesario realizar interpretaciones adicionales.

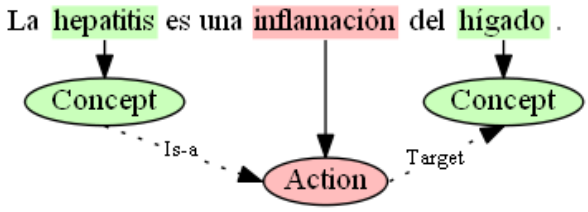
Las relaciones *is-a* y *part-of* pueden ser indicadas explícitamente en el texto por la aparición de patrones textuales comunes, como es el caso de los patrones de Hearst [9]. Sin embargo, aun cuando no ocurrieran en el texto indicaciones explícitas de estas relaciones, se considera su anotación.

En la figura 2.6a puede verse un ejemplo de anotación de la relación *is-a*. En esta oración, las palabras «`hepatitis`» e «`hígado`» son claramente conceptos, mientras que «`inflamación`» es una acción. Como se ha visto anteriormente, una relación con un rol complejo, es decir, un rol que esté relacionado hacia otros roles, implica la relación con él como un todo y no solo con su significado semántico. Por ende, «`hepatitis` *is-a* `inflamación del hígado`» es el resultado de la anotación de esta oración.

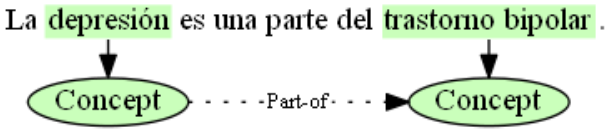
Por otra parte, en la figura 2.6b se puede apreciar un ejemplo de la relación *part-of*. En esta oración son anotadas como conceptos la palabra «`depresión`» y la frase «`trastorno bipolar`». Esta oración, a modo de anotación, resulta en «`depresión` *part-of* `trastorno bipolar`».

Las parejas de sinonimia, anotadas como relaciones *same-as* (en español *igual-que*) es usada para indicar sinónimos o conceptos que son considerados iguales en

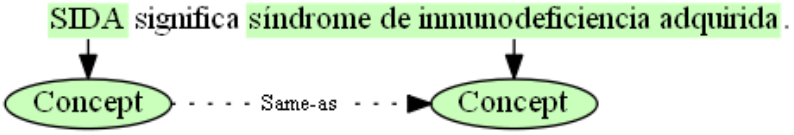
el dominio del documento. Puede ser usada cuando un concepto simple es definido a partir de describirlo como otro concepto más complejo.



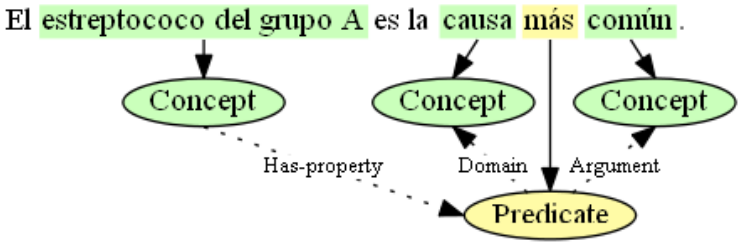
(a) Ejemplo de anotación de hiperonimia e hiponimia.



(b) Ejemplo de anotación de meronimia y holonimia.



(c) Ejemplo de anotación de sinonimia.



(d) Ejemplo de anotación de propiedad.

Figura 2.6: Anotación de las relaciones taxonómicas

La figura 2.6c muestra un ejemplo de anotación de la relación *same-as*. La palabra «SIDA» y la frase «síndrome de inmunodeficiencia adquirida» son anotadas como conceptos. Por tanto, esta oración a modo de anotación queda «SIDA *same-as* síndrome de inmunodeficiencia adquirida».

Las propiedades, anotadas como relaciones *has-property* (traducido al español como *tiene-propiedad*) es usada para especificar que un concepto tiene una propiedad, característica, o puede ser descrita por otro concepto. Sin embargo, este tipo de relación puede conllevar a ciertas dificultades, como por ejemplo la paradoja de Bertrand Russell [27] y la de Grelling-Nelson [29]. Además, una propiedad puede implicar gran cantidad de propiedades e incluso una cantidad infinita de ellas. Por ejemplo si «la persona pesa más de 60 kilogramos» entonces también se cumple que «la persona pesa más de 59 kilogramos» y por consiguiente, que «la persona pesa más de 58 kilogramos». De manera similar, dado que en este caso, el peso es un valor numérico decimal, se pueden construir infinitas propiedas de este tipo.

En la figura 2.6d se puede observar un ejemplo de la anotación *has-property*. En esta oración son anotadas la frase «estreptococo del grupo A» y las palabras «causa» y «común» como conceptos. También, la palabra «más» es anotada como un predicado, en conjunto con «causa» y «común» como dominio y argumento respectivamente. Al ser anotada la relación *has-property*, la anotación resulta como «estreptococo del grupo A *has-property* causa más común».

Para todas las relaciones taxonómicas, solo se considera su anotación cuando la oración implica la existencia de ella, aun cuando fuese implícita. En ningún caso se anota basada solamente en conocimiento externo o del dominio.

2.1.7. Causalidad e implicación

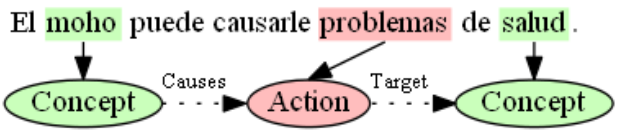
Las cuatro relaciones semánticas presentadas hasta ahora son útiles para capturar la estructura taxonómica del conocimiento expresado en textos del lenguaje natural. Dos relaciones adicionales son definidas para construir conexiones lógicas entre conceptos: *causes* y *entails* (en español *causa* e *implica* respectivamente). La relación *causes* es usada para expresar que un evento, identificado en general como un concepto, es una posible causa para otro evento. En la figura 2.7a se muestra un ejemplo anotado.

Esta relación indica causalidad, no correlación ni implicación lógica. Por tanto, debe estar declarado con claridad en la oración que hay una conexión de causa

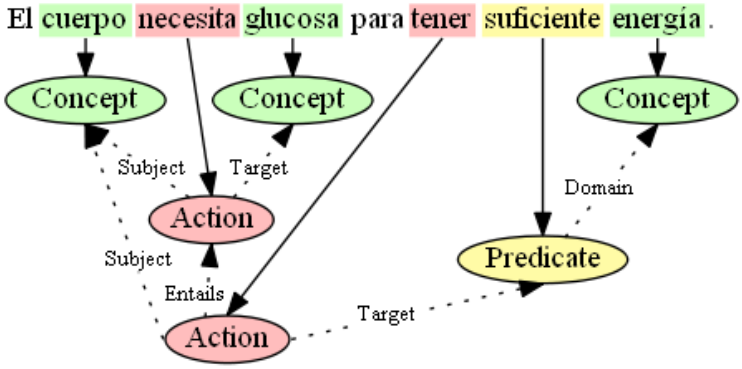
directa entre ambos eventos. Además, hay un grado de incertidumbre implicada en la causalidad, lo cual significa que si «A *causes* B», eso no necesariamente implica que cada vez que pase «A» sería seguido por «B», ni que en cualquier caso que ocurra «B» será a causa de «A».

En contraste, la relación *entails* es usada para denotar implicación lógica. En este caso, no es necesario que los eventos estén relacionados por causalidad; lo único que debe cumplirse es que cuando la proposición «A» es verdadera entonces siempre sucede el caso de que la proposición «B» es verdadera. En la figura 2.7b puede verse un ejemplo, donde un concepto complejo, en este caso «tener» implica «necesita», el cual es también un concepto complejo. Desde otro punto de vista, la anotación de esa oración resulta en:

«(tener [suficiente energía] en el cuerpo)
entails
(necesitar glucosa en el cuerpo)»



(a) Ejemplo de anotación de causalidad.



(b) Ejemplo de anotación de implicación.

Figura 2.7: Anotación de causalidad e implicación

La anotación de causalidad e implicación evita anotar varias palabras y frases que comparten el mismo significado semántico. Por ejemplo, en la figura 2.7a no

resulta necesario anotar «`puede causarle`» debido a que el significado correcto está siendo representado por la relación *causes*.

2.1.8. Contextualización

En ocasiones, los conceptos solo participan en determinada relación con precondiciones, como por ejemplo, si dura un período específico de tiempo, solo en una ubicación específica o con algunas propiedades adicionales.

En la figura 2.8b, la anotación «`injerto óseo-transplanta-tejidos`» falla en capturar la semántica completa del mensaje, dado que el «`injerto óseo`» no es necesariamente siempre «`transplantar tejidos`», sino solo en la situación específica en la que este tejido es de los «`huesos`». Para resolver estas situaciones, se incluyen tres relaciones de contexto: *in-time*, *in-place* y el más general *in-context* (en español *en-tiempo*, *en-lugar* y *en-contexto* respectivamente).

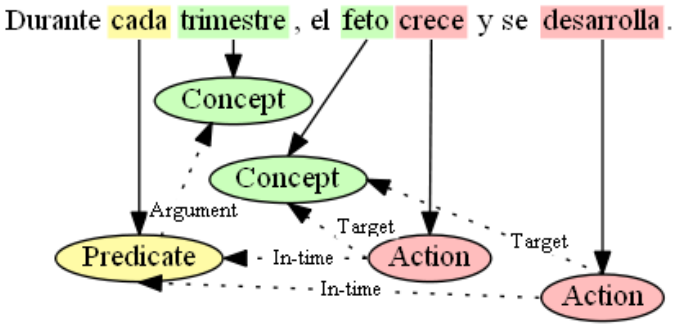
La relación *in-time* restringe un concepto a un instante de tiempo determinado. Además, permite atrapar restricciones más generales, siempre que hablen del concepto mientras cumpla determinada condición o durante el tiempo que lo hace. Puede verse un ejemplo en la figura 2.8a.

La relación *in-place* restringe un concepto a un lugar determinado. Además, puede ser visto como la contextualización de la relación *part-of*, de esta forma permite plantear un hecho sobre un concepto que es parte de otro. Puede verse un ejemplo en la figura 2.8b.

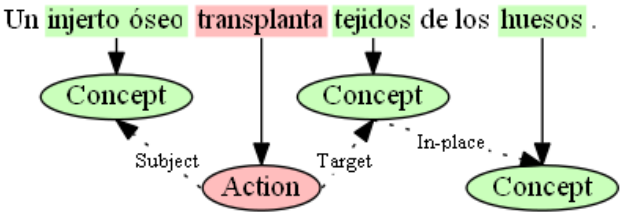
La relación *in-context* restringe un concepto a condiciones más abarcadoras que las descritas anteriormente. Es el contextualizador más general y al igual que el resto, solo debe ser aplicado cuando el contexto habla de un rasgo o valor que puede tener el concepto a contextualizar. Eso implica que el objeto a contextualizar debe tener semántica propia independiente del contexto. A grandes rasgos, puede verse como el contextualizador de la relación *has-property*. Un caso particular de su uso es en oraciones imperativas, donde fragmentos de oración escritos como «`... si X entonces haga Y ...`» se anotarían como «`Y` *in-context* `X`». Puede verse un ejemplo en la figura 2.8c.

La diferencia entre las relaciones de contexto y el resto es que ellas no definen una aserción, sino que son útiles solo para construir conceptos más complejos. Por ejemplo, la anotación «`problemas` *in-context* `únicos`» no solo significa que las mujeres tienen problemas de salud, sino que además, son únicos. Es exclusivamente cuando se enlaza con otro concepto, a través de *has-property* u otra rela-

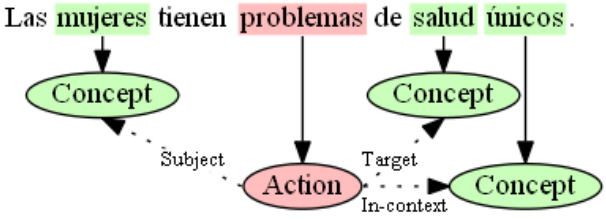
ción, que la construcción toma sentido. Por esta razón, no es correcto intercambiar arbitrariamente *in-context* con *has-property*, ya que una relación *has-property* declara una aserción concreta por sí misma. De igual forma enlazar un concepto sobre el que se ha establecido una relación que no es de contextualización, con otro, a través de alguna relación o rol, no indica que dicha relación o rol sea válida solamente para aquellas instancias del concepto que cumplan la propiedad indicada por la relación que no es de contextualización, puesto que estas relaciones, no construyen conceptos complejos que se puedan enlazar.



(a) Ejemplo de anotación de tiempo.

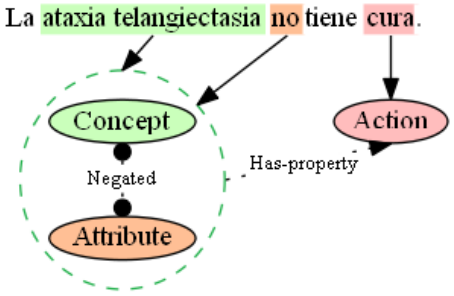


(b) Ejemplo de anotación de lugar.



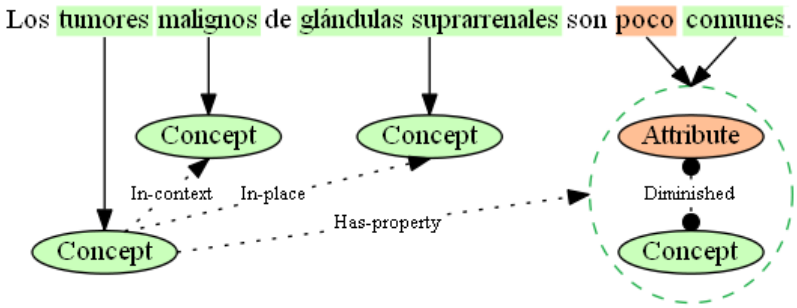
(c) Ejemplo de anotación de contexto.

Figura 2.8: Anotación de contextualización

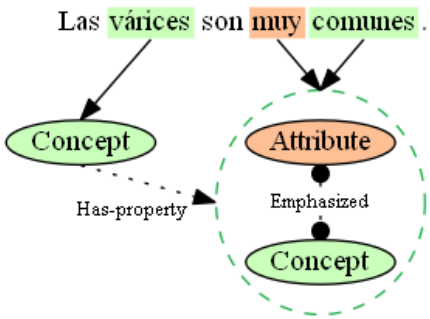


(a) Ejemplo de anotación del atributo negación.

(b) Ejemplo de anotación del atributo incertidumbre.



(c) Ejemplo de anotación del atributo disminución.



(d) Ejemplo de anotación del atributo énfasis.

Figura 2.9: Anotación de los atributos

2.1.9. Atributos

Cuatro atributos booleanos[2] adicionales pueden ser asociados a cualquier concepto para calificarlo o describirlo un poco más. Ellos son: *negated*, *uncertain*, *diminished* y *emphasized* (en español *negación*, *incertidumbre*, *disminución* y *énfasis* respectivamente). Estos atributos son usados para evitar anotar palabras del idioma que son usadas con bastante frecuencia como *no*, *puede*, *poco*, *mucho*, y en su lugar asociar directamente el calificador correspondiente al concepto en sí. Además, los atributos capturan la negación, incertidumbre, disminución o énfasis que se pretendía en la oración, aun cuando sea implícito y no indicado explícitamente por otra palabra de la misma. Estos atributos acompañan al concepto que modifican en todas las relaciones en que este participe. En la figura 2.9 puede verse un ejemplo anotado de cada uno de estos cuatro atributos.

2.2. Formato de anotación

Las anotaciones creadas con este formato son guardadas en archivos separados del documento de texto. Este último nunca es modificado. Para cada documento que vaya a ser anotado, se crea su respectivo archivo de anotación. Ambos archivos son asociados a través de su nombre, los cuales coinciden completamente exceptuando en su extensión. Las extensiones correspondientes al documento de texto y al anotado son `.txt` y `.ann` respectivamente.

En el documento anotado, las anotaciones individuales se conectan a fragmentos de texto a través de rangos de posiciones de los caracteres. Por ejemplo, si el documento comenzara así:

"*El consumo de alcohol puede causar problemas en el hogar . . .*"

La palabra «`consumo`» se identifica por el rango de posiciones 3 . . . 10. Las posiciones comienzan en 0 con el inicio del documento, además, todo carácter cuenta como posición válida, incluyendo los espacios en blanco y los cambios de línea.

[2]El tipo de dato lógico o booleano es en computación aquel que puede representar valores de lógica binaria, estos son dos valores, los cuales normalmente se representan como falso y verdadero. Se utiliza generalmente en la programación, estadísticas, electrónica y matemáticas mediante la utilización del álgebra booleana.

De manera formal, lo anterior queda expresado como: sea $\sum$ un alfabeto de símbolos, el documento $D = c_1 c_2 \ldots c_n$ y la palabra o frase w perteneciente a este, port tanto, $\exists v$ tal que vw es un prefijo de D, entonces el rango de posiciones de w en el archivo anotado comienza en $|v|$ y acaba en $|v| + |w|$.

2.2.1. Archivo de texto

El archivo de texto debe tener como extensión «`.txt`» y contener la información del documento original. Además, debe estar guardado en texto plano y codificado usando UTF-8[3]. Puede contener cambios de línea, los cuales cuentan como un símbolo.

2.2.2. Archivo de anotación

El archivo de anotación debe tener como extensión «`.ann`». Además, estar guardado en texto plano y codificado usando UTF-8[3]. Los tipos de anotación específicas que pueden estar presentes en este archivo se explican en las proximas secciones.

2.2.3. Estructura general de la anotación

Todas las anotaciones tienen la misma estructura básica: cada línea tiene una sola anotación específica y esta tiene un identificador único que se encuentra al comienzo de la misma. Luego, separado por un carácter TAB, se encuentra el resto de la información en la anotación específica.

El resto de la estructura de cada anotación específica varía en dependencia de los distintos tipos que hay. Esto se explica detalladamente en las siguientes secciones.

```
T9  Concept 658 664 cuerpo
R13 in-place Arg1:T3 Arg2:T13
A1  Negated T10
*   same-as T1 T2
```

Figura 2.10: Ejemplo de escritura del identificador de anotaciones.

[3]Siglas de ***U**nicode **T**ransformation **F**ormat - 8*, un formato de codificación de caracteres.

2.2.4. Convenio de anotación de identificadores

Todos los identificadores de anotaciones consisten en un único carácter en mayúsculas identificando el tipo de anotación y a continuación su número. Este carácter inicial es:

- `T`: texto
- `R`: relación
- `A`: atributo
- `#`: nota o comentario

Adicionalmente, un asterisco (“*”) puede ser usado como un identificador, pero solo en casos especiales.

2.2.5. Anotación de texto

La anotación de texto es una categoría importante, incluso pudiera decirse que es la base de la anotación, pues es quien delimita los fragmentos de texto específicos que serán usados y además, sobre estos es que las relaciones y atributos surten efecto. Estas se basan en la siguiente estructura:

```
T<id>[TAB]<type>[SPACE]<span>[TAB]<text>
```

Figura 2.11: Estructura de anotación de texto.

- `[TAB]` y `[SPACE]` son el carácter TAB y el carácter espacio respectivamente.
- `<id>` es el número correspondiente a esa anotación específica.
- `<type>` es el tipo de rol, el cual puede ser solo uno de los cuatro siguientes: *Concept*, *Action*, *Reference* o *Predicate*.
- `<span>` son los rangos de posiciones de las palabras pertenecientes al texto que será anotado. Si este fragmento contiene más de una palabra, entonces se unirán sus rangos de posiciones usando el carácter punto y coma (“;”) como separador y estas serán escsritas separadas por un carácter espacio.
- `<text>` es el texto específico que será anotado.

Para este ejemplo, y los restantes en esta sección pertenecientes a explicar el formato de anotación, se usará el siguiente documento de ejemplo, el cual contiene una única oración:

«*Las mujeres embarazadas también pueden desarrollar diabetes, llamada diabetes gestacional.*»

En la figura 2.12 se puede ver el resultado de anotar el texto relevante en el documento de prueba.

```
T1  Concept 4 11    mujeres
T2  Concept 12 23   embarazadas
T3  Action 39 50    desarrollar
T4  Concept 51 59   diabetes
T5  Concept 69 77;78 89 diabetes gestacional
```

Figura 2.12: Ejemplo de anotación de texto.

2.2.6. Anotación de relaciones

Todas las relaciones anotadas son binarias[4]. Estas anotaciones siguen la siguiente estructura:

```
R<id>[TAB]<type>[SPACE]Arg1:T<id1>[SPACE]Arg2:T<id2>
```

Figura 2.13: Estructura de anotación de texto.

- `[TAB]` y `[SPACE]` son el carácter TAB y el carácter espacio respectivamente.
- `<id>` es el número correspondiente a esa anotación específica.
- `<type>` es el tipo de relación anotada, el cual puede ser solo uno de los trece vistos anteriormente: *subject*, *target*, *domain*, *argument*, *is-a*, *part-of*, *same-as*, *has-property*, *causes*, *entails*, *in-time*, *in-place* o *in-context*.
- `<id1>` e `<id2>` son los identificadores de las anotaciones de texto que participan en esta relación.

[4]Una relación binaria R es el subconjunto de los elementos del producto cartesiano $A_1 \times A_2$ que cumplen una determinada condición: $R = \{(a_1, a_2) : (a_1, a_2) \in A_1 \times A_2 \wedge R(a_1, a_2) = Verdadero\}$.

Es necesario tener en cuenta que este tipo de anotaciones se interpretan de izquierda a derecha, es decir, se interpretan «T<id1> <*type*> T<id2>». Por tanto, el orden en que son anotados sus argumentos tiene vital importancia.

En la figura 2.14 se puede ver el resultado de anotar las relaciones en el documento de prueba.

```
R1  in-context Arg1:T1 Arg2:T2
R2  subject Arg1:T3 Arg2:T1
R3  target Arg1:T3 Arg2:T4
R4  same-as Arg1:T3 Arg2:T5
```

Figura 2.14: Ejemplo de anotación de relaciones.

Como se comentó anteriormente, en algunos casos especiales un asterisco (“*”) puede ser usado en vez de un identificador. Este es el caso de la relación `same-as`, la cual es simétrica y transitiva. Ellas pueden pueden tener un asterisco al comienzo o un identificador estándar como las restantes relaciones. Cuando se utiliza un asterisco, no es necesario anotar los argumentos explícitamente junto a los identificadores, ni tampoco que sean solo dos argumentos. Por este motivo puede ser compactada y los identificadores que actúan en esta relación deben ser anotados separados por un único carácter espacio, sin importar si estos son dos o más.

A continuación en la figura 2.15 se puede ver un ejemplo de anotación de este tipo en el documento de prueba.

```
*   same-as T3 T5
```

Figura 2.15: Ejemplo de anotación de la relación `same-as`.

2.2.7. Anotación de atributos

Los atributos booleanos anteriormente mencionados, son asociados a su respectivo concepto a través de una relación binaria. La misma contiene un único concepto específico unido a un tipo de atributo. Cabe aclarar que, como se comentó en la sección 2.1.9, estos sintetizan una familia de palabras y por ende, semánticamente no tienen un palabra explícitamente asociada.

Las anotaciones de atributos siguen la siguiente estructura:

```
A<id>[TAB]<type>[SPACE]T<id1>
```

Figura 2.16: Estructura de anotación de texto.

- `[TAB]` y `[SPACE]` son el carácter TAB y el carácter espacio respectivamente.
- `<id>` es el número correspondiente a esa anotación específica.
- `<type>` es el tipo de atributo anotado, el cual puede ser solo uno de los cuatro vistos anteriormente: *Negated*, *Uncertain*, *Diminished* o *Emphasized*.
- `<id1>` es el identificador de la anotación de texto que es modificada por este atributo.

En la figura 2.17 se puede ver el resultado de anotar el único atributo contenido en el documento de prueba.

```
A1  Uncertain T3
```

Figura 2.17: Ejemplo de anotación de atributo.

2.2.8. Anotación de comentarios

Un comentario es texto que no se procesa, por lo que sirve para escribir notas a modo de guía. Las anotaciones de comentarios siguen la siguiente estructura:

```
#<id>[SPACE]<comment>
```

Figura 2.18: Estructura de anotación de comentarios.

- `[SPACE]` es el carácter espacio.
- `<id>` es el número correspondiente a esa anotación específica. Este no es obligatorio para anotar comentarios.
- `<comment>` es el comentario en sí, y puede contener cualquier texto.

En la figura 2.19 se puede ver un ejemplo de un comentario anotado.

```
# Las mujeres embarazadas también pueden desarrollar
# diabetes, llamada diabetes gestacional.
```

Figura 2.19: Ejemplo de anotación de comentario.

2.2.9. Consideraciones finales

Las oraciones anotadas en el documento con extensión «.ann» deben estar en el mismo orden con el que aparecen en el documento de texto con extensión «.txt». Además, no es necesario que todas las oraciones en el documento de texto sean anotadas, incluso puede que ninguna lo esté.

Como se vio en la sección 2.2.4, la letra inicial del identificador debe ser mayúscula, no obstante, no importa si es minúscula, aunque siempre guiarse por los convenios es una buena práctica.

Por otra parte, el convenio de anotación de los tipos es hacerlo con la primera letra en mayúscula y el resto en minúsculas para los textos y atributos, mientras que para las relaciones se debe anotar completamente en minúsculas.

Además, el orden de las anotaciones en el archivo no es relevante y tampoco lo son los números específicos de los identificadores. Aunque por cuestión de estética y para una mejor comprensión del anotador, se recomienda escribir siempre las anotaciones de texto primero, luego las relaciones y por último los atributos. En caso de los identificadores, estos deberían comenzar con el número 1 y continuar la secuencia de forma incremental de uno en uno.

Por último, es recomendable ordenar las anotaciones de texto respecto a la posición inicial de la primera palabra en estos. Las relaciones deben estar ordenadas por los identificadores de sus argumentos, es decir, ordenar primero por el identificador del primer argumento y en caso de que varios coincidan, ordenar por el del segundo argumento.

Como caso especial están las anotaciones de relaciones de igualdad que comienzan con asteriscos, las cuales deben ser ordenadas entre ellas con igual criterio que las demás relaciones, pero deben ir después de estas. Estas anotaciones deberían comenzar siempre con un asterisco, aunque, como se mencionó en la sección 2.2.6, no tiene ningún inconveniente hacerlo de manera similar a las demás.

Por último, los atributos deben ordenarse de manera similar, pero en este caso, guiándose por el identificador su único argumento.

Siguiendo las recomendaciones anteriores, el documento de prueba quedaría anotado de la siguiente manera:

```
# Las mujeres embarazadas también pueden desarrollar
# diabetes, llamada diabetes gestacional.
T1  Concept 4 11    mujeres
T2  Concept 12 23   embarazadas
T3  Action 39 50    desarrollar
T4  Concept 51 59   diabetes
T5  Concept 69 77;78 89 diabetes gestacional
R1  in-context Arg1:T1 Arg2:T2
R2  subject Arg1:T3 Arg2:T1
R3  target Arg1:T3 Arg2:T4
*   same-as T3 T5
A1  Uncertain T3
```

Figura 2.20: Ejemplo de anotación de comentario.

2.3. Anotación automática de documentos

El proceso de anotación de un documento, llevado a cabo por un humano, puede ser engorroso. Dado que un corpus contiene varios de estos, anotarlo puede tomar gran cantidad de tiempo.

No es de extrañar que con el avance tecnológico y científico, principalmente de la inteligencia artificial en este último, este proceso pueda automatizarse. Los avances en este aspecto tienen ventajes y desventajas, a la vez de márgenes de errores y precisión.

En la competencia *eHealth-KD Challenge* presentada en *IberLEF 2019* [22] e *IberLEF 2020* [21] tomaron lugar sistemas automáticos para la anotación de corpus [5]. Las propuestas que compitieron están entrenadas para hacerlo en textos médicos del idioma español y usando el modelo de anotación explicado en esta investigación [19, 5]. De igual forma, otros pueden ser encontrados en la literatura para anotar automáticamente documentos pertenecientes a diferentes dominios e idiomas.

2.4. Análisis del corpus

El corpus usado [20] fue construido a partir de un fichero *XML*[5] tomado del sitio web de *Medline* el 9 de enero de 2018, específicamente a las 02 : 30 : 31. *Medline* fue producida y es mantenida por la Biblioteca Nacional de Medicina de los Estados Unidos. Recoge referencias bibliográficas de los artículos publicados en aproximadamente 5,500 revistas médicas desde 1966. Actualmente reúne más de 30,000,000 de citas. Cada registro de *Medline* es la referencia bibliográfica de un artículo científico publicado en una revista médica, con los datos bibliográficos básicos de un artículo: título, autores, nombre de la revista, año de publicación, entre otros. Esto permite la recuperación de estas referencias posteriormente en una biblioteca o a través de un *software* específico de recuperación.

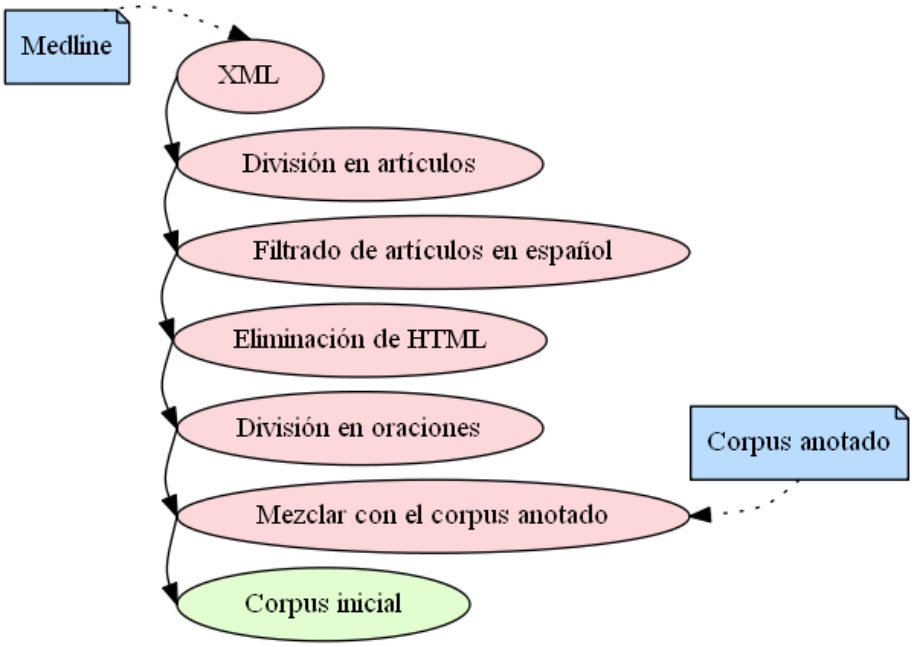


Figura 2.21: Esquema del procesamiento inicial del corpus.

En la figura 2.21 puede verse una representación esquemática del procesamiento inicial que se le hace al corpus de *Medline*, el cual de forma análoga puede aplicarse a otros.

[5]Siglas en inglés de *eXtensible Markup Language*, un lenguaje de marcado desarrollado por el *World Wide Web Consortium* (W3C).

Capítulo 3

Propuesta de Solución

Esta investigación busca poder expresar un corpus anotado a través de una ontología definida, generando un grafo de conocimiento como resultado. Otro de los objetivos claros, es poder hacer esto mediante un algoritmo computacional, el cual debe ser, además, finito y determinista.

3.1. Analizador sintáctico

Primeramente, es necesaria la creación de una herramienta capaz de fragmentar en objetos con significado computacional el contenido de los archivos de anotación escritos con el formato visto en la sección 2.2.

Dado que en el formato de archivo propuesto contiene una única relación anotada por línea y a su vez, las relaciones tienen su formato de escritura bien definido y sin ambigüedades, llevar a cabo la implementación de este analizador sintáctico es bastante sencillo. Esto puede hacerse a través de expresiones regulares, las cuales son ampliamente usadas y muchos de los lenguajes de programación modernos las incluyen como estructuras integradas.

3.2. Modelo ontológico

La ontología propuesta es de propósito general y basada en el modelo de anotación visto en la sección 2.1. Esto posibilita la continuidad del proceso, partiendo desde documentos escritos en lenguaje natural, hasta la creación de una base de conocimiento a partir de ellos.

3.2.1. Clases en la ontología

La ontología se basa en tres clases fundamentales:

- Entidad simple: la más sencilla de las clases, no tiene ningún significado especial. En el modelo de la sección 2.1 representa un concepto sin haberle aplicado relaciones.
- Entidad con atributo: está compuesta por una *entidad simple* y uno o más atributos de los mencionados en la sección 2.1. Puede verse como el resultado de haber aplicado todos los atributos pertenecientes a un mismo concepto.
- Entidad compuesta: está compuesta por dos o más entidades, estas pueden ser de cualquier tipo. Puede verse como el resultado de aplicar las relaciones en el modelo de anotación explicado en la sección 2.1.

Estas clases contienen, además, dos propiedades; una de ellas es la palabra o fragmento de texto en sí que representa la entidad y la otra es el tipo de concepto al que representan. Esta última propiedad se basa en los cuatro tipos de concepto existentes en el modelo de anotación explicado en la sección 2.1, estos son: *Concept*, *Action*, *Reference* y *Predicate*.

3.2.2. Relaciones en la ontología

Los trece tipos de relación vistos en la sección 2.2.6 son relaciones válidas para este modelo. Estas, a su vez, pueden originarse y terminar en cualquiera de las tres posibles clases de la ontología propuesta.

Estas relaciones cobran el mismo significado semántico que en la sección 2.1, estos son: *subject*, *target*, *domain*, *argument*, *is-a*, *part-of*, *same-as*, *has-property*, *causes*, *entails*, *in-time*, *in-place* o *in-context*.

En esta ocasión, los implicados en las relaciones son entidades de la ontología, y por ende, la palabra o fragmento de texto que estas representan. El proceso de creación de una base de conocimiento específica a partir de la definición de esta ontología es llevado a cabo de forma totalmente automática y no de la manera tradicional, con expertos en el dominio añadiendo relaciones entre clases una tras otra. A la interrogante de en qué orden se llevan a cabo estas relaciones y quiénes participan en ellas se le da respuesta en la próxima sección.

3.3. Grafo de conocimiento

Una vez que se haya analizado sintácticamente todo el corpus, se tendrá la información de las anotaciones en objetos computacionales y será más sencillo el trabajo con estos. En aras de evitar ambigüedades y concentrar el conocimiento para un mejor entendimiento de este y a la misma vez, poder facilitar la tarea de extraerlo de este grafo por un equipo de cómputo, el texto anotado es normalizado. Esto es llevado a cabo teniendo en cuenta las palabras que lo componen, y anotando en su lugar la palabra primitiva de esta. Por ejemplo, la palabra «`sangramiento`» será anotada como «`sangrar`» y la frase «`glóbulos rojos`» como «`glóbulo rojo`».

Como se vio en la sección 2.1, es necesario aclarar que hay que darle un orden a la creación de las instancias de las clases y las relaciones en este grafo, pues las propias anotaciones de texto y las relaciones tienen un orden implícito entre ellas. Por ejemplo, en la propia figura 2.5, se debe procesar primero el rol ejercido por «`problemas`» y por «`cumpla`» antes de poder procesar «`impiden`»; de lo contrario, el conocimiento descrito por «`impiden`» quedaría incompleto o mal representado.

3.3.1. Orden topológico

El orden establecido por el modelo de anotación visto en el capítulo 2 es un orden topológico. Para ello se tiene en cuenta el siguiente orden:

1. Texto
2. Atributos
3. Relaciones de acción, predicado y contextualización
4. Relaciones taxonómicas y de causa e implicación

Es válido aclarar que una vez divididas las relaciones en estos tres grupos, estas son asociadas nuevamente, esta vez teniendo en cuenta la parte izquierda de cada una de ellas (`Arg1` en el archivo de anotación). Se crea una instancia de clase por cada una de estas agrupaciones, una instancia creada en un nivel más avanzado, teniendo en cuenta el orden visto anteriormente, representa una mayor cantidad de información y al mismo tiempo, información más específica respecto a su instancia asociada en niveles anteriores.

En la figura 3.1 puede verse un ejemplo de documento de texto, en este caso contiene una sola oración para que sea sencillo y entendible, pero en la práctica estos archivos contienen muchas más.

En la figura 3.2 se ve el documento anotado asociado a este. Como puede apreciarse, se siguieron los convenios establecidos en la sección 2.2.9.

En la figura 3.5 se puede ver cómo queda el grafo de conocimiento resultante de este corpus. Este contiene un solo documento en esta ocasión, pero en la práctica usualmente tiene muchos más.

La flecha que puede haber en algunas líneas del ejemplo significa que esta en el documento es muy larga para ser mostrada en una única línea en el ejemplo y se continuará escribiendo debajo.

En estos ejemplos, las palabras o frases no son normalizadas para un mejor entendimiento en lenguaje natural y además, porque hay varias formas, métodos y decisiones de cómo normalizar palabras o frases.

```
El desmayo (o síncope) es una pérdida temporal de la ↪
conciencia.
```

Figura 3.1: Ejemplo 1: documento "`desmayo.txt`".

```
# Sentence 1: El desmayo (o síncope) es una pérdida ↪
temporal de la conciencia.
# Keyphrases
T1  Concept 3 10    desmayo
T2  Concept 14 21   síncope
T3  Action 30 37    pérdida
T4  Concept 38 46   temporal
T5  Concept 53 63   conciencia
# Relations
R1  is-a Arg1:T1 Arg2:T3
R2  in-context Arg1:T3 Arg2:T4
R3  target Arg1:T3 Arg2:T5
*   same-as T1 T2
```

Figura 3.2: Ejemplo 1: documento "`desmayo.ann`".

Siguiendo el orden establecido anteriormente, se puede ver en la figura 3.3 el grafo de conocimiento resultante luego de realizado el punto 1 es:

Figura 3.3: Ejemplo 1: grafo de conocimiento luego de realizado el punto 1.

En el punto 2 no hay nada que hacer en este corpus, pues no hay atributos, por tanto, el grafo de conocimiento quedará idéntico. Para el punto 3 son usadas las relaciones `R2` y `R3`, resultando:

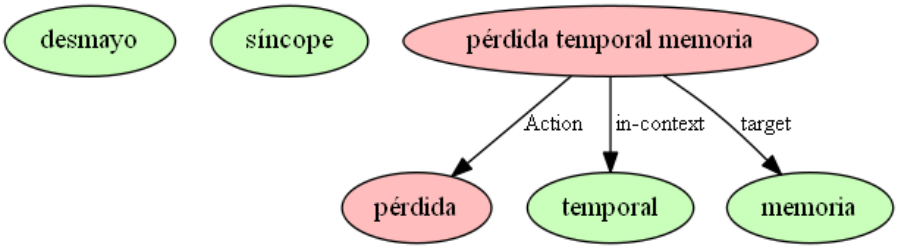


Figura 3.4: Ejemplo 1: grafo de conocimiento luego de realizado el punto 3.

Para el punto 4 son usadas las relaciones restantes: `R1` y $*$. Resultando:

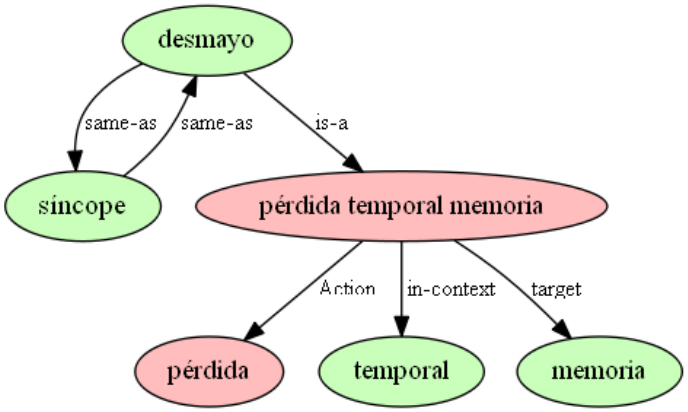


Figura 3.5: Ejemplo 1: grafo de conocimiento luego de realizado el punto 4.

El ejemplo anterior es sencillo y muestra un documento con una única oración también sencilla. Ahora se muestra un ejemplo de un corpus que igualmente contiene un solo documento compuesto por una oración, pero esta vez, una oración más compleja y que abarca todos los grupos de relaciones que hay en el modelo de anotación.

La figura 3.6 muestra el documento de texto, en la 3.7 se aprecia su respectivo archivo anotado y en la 3.11 el resultado final del grafo de conocimiento.

```
Las buenas prácticas de higiene, incluyendo lavarse las ↪
manos correctamente, pueden evitar infecciones.
```

Figura 3.6: Ejemplo 2: documento "higiene.txt".

```
# Sentence 1: Las buenas prácticas de higiene, incluyendo ↪
lavarse las manos correctamente, pueden evitar infecciones.
# Keyphrases
T1  Concept 4 10    buenas
T2  Predicate 11 20 prácticas
T3  Concept 24 31   higiene
T4  Action 44 51    lavarse
T5  Concept 56 61   manos
T6  Concept 62 75   correctamente
T7  Action 84 90    evitar
T8  Concept 91 102  infecciones
# Relations
R1  in-context Arg1:T2 Arg2:T1
R2  domain Arg1:T2 Arg2:T3
R3  causes Arg1:T2 Arg2:T7
R4  is-a Arg1:T4 Arg2:T2
R5  target Arg1:T4 Arg2:T5
R6  in-context Arg1:T4 Arg2:T6
R7  target Arg1:T7 Arg2:T8
# Attributes
A1  Uncertain T7
```

Figura 3.7: Ejemplo 2: documento "higiene.ann".

Una vez más, siguiendo el orden establecido anteriormente, se puede ver en la figura 3.8 el grafo de conocimiento resultante luego de realizado el punto 1.

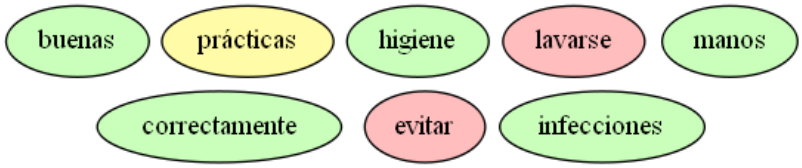


Figura 3.8: Ejemplo 2: grafo de conocimiento luego de realizado el punto 1.

Dándole solución al punto 2, el grafo quedaría así:

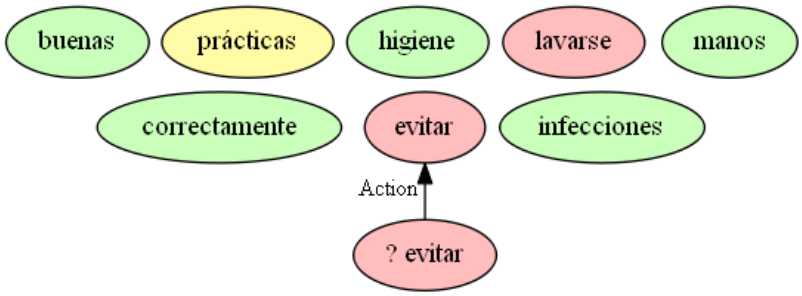


Figura 3.9: Ejemplo 2: grafo de conocimiento luego de realizado el punto 2.

Una vez completado el punto 3, este es el resultado:

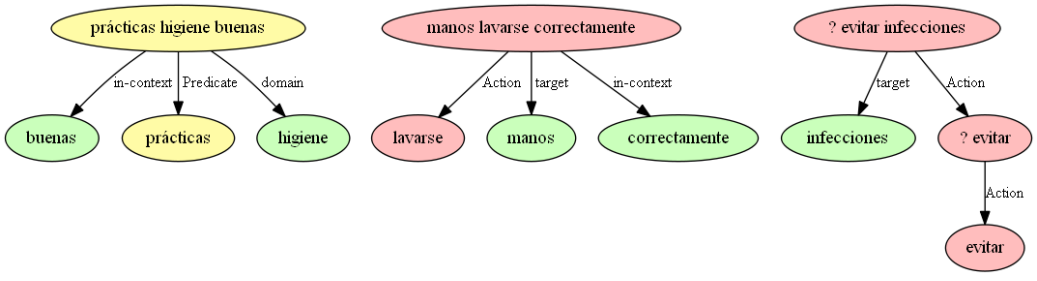


Figura 3.10: Ejemplo 2: grafo de conocimiento luego de realizado el punto 3.

Finalmente, al llevar a cabo el punto 4, este sería el resultado:

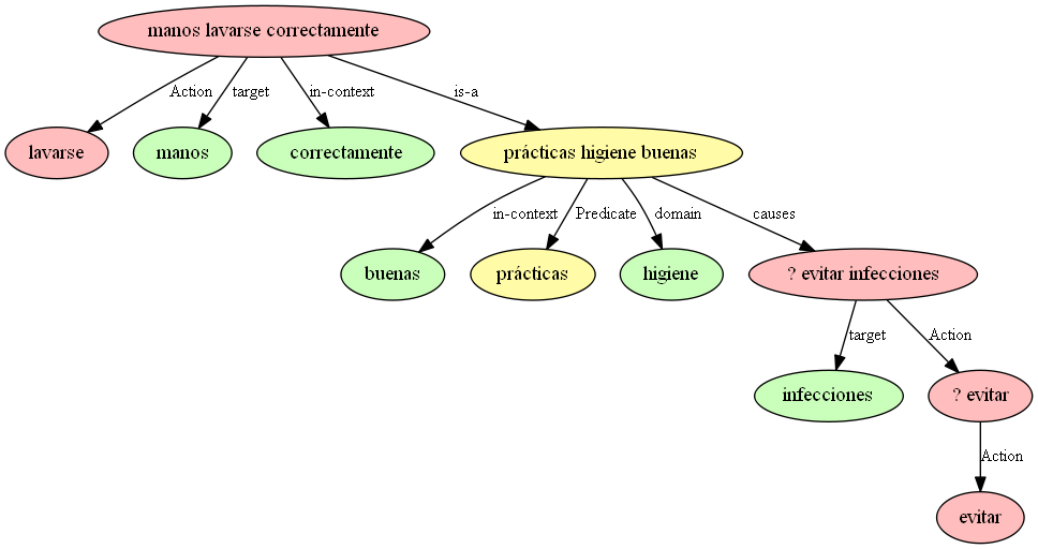


Figura 3.11: Ejemplo 2: grafo de conocimiento luego de realizado el punto 4.

Como pudo apreciarse en el ejemplo anterior, se optó por mostrar los atributos en el grafo a través de caracteres en vez de la palabra en sí. Son usados los siguientes caracteres:

¬ negación

? incertidumbre

↓ disminución

↑ énfasis

3.3.2. Creación de instancias de clases

Para llevar a cabo el punto 1 en el orden previamente expuesto, se crea una *entidad simple* por cada concepto existente en el documento de anotación. Esto sienta las bases para la posterior realización y correctitud del algoritmo expuesto en la sección anterior.

Para cumplimentar lo propuesto en el punto 2, cumplen un papel protagónico las entidades de los conceptos que tienen atributos asociados. En este punto, todas estas son del tipo *entidad simple* y cada una de ellas se une con todos sus respectivos atributos, formando una *entidad con atributo*.

En el paso 3 tienen lugar algunas de las relaciones. Cada una de ellas conforma una *entidad compuesta*. Esta nueva instancia se relaciona con los conceptos de las partes derecha de dichas relaciones, ahora devenidos en alguno de los tres tipos de clases de esta ontología, a través del tipo de relación. A la vez que se relaciona con la parte izquierda de estas por medio del tipo de entidad que sean.

El paso 4 no crea instancias nuevas, solo establece la relación entre dos instancias creadas previamente en el grafo, aportando así conocimiento al mismo.

3.4. Resolución de correferencias

La resolución de correferencias es la tarea de encontrar todas las expresiones que refieren a la misma entidad en un texto. Es un paso importante para muchos algoritmos de procesamiento de lenguaje natural y por tanto, lo es también para esta investigación.

Ejemplificando, es el problema de darse cuenta que en el fragmento de texto "*... si una ampolla es grande, dolorosa o parece que se reventará por sí sola, usted puede drenar el líquido. Esto debería reducir su dolor...*" la palabra «`Esto`» hace referencia a «`drenar el líquido`».

En el marco del ámbito social y mundial en que fue hecha esta investigación y por motivos principalmente de recursos, no se pudo dar resultados concretos en esta tarea. Convirtiéndose así en un objetivo que con el paso del tiempo pasó de estar plenamente involucrado con este estudio a ser una recomendación para el futuro. De esta manera se dejan abiertas las puertas para la continuación y mejora de lo que aquí se presenta.

Capítulo 4

Análisis de Resultados

El esquema de anotación y el modelo ontológico presentado en este trabajo, y por tanto el grafo de conocimiento generado a partir de ellos, tienen como objetivo fundamental asistir en el desarrollo de sistemas de descubrimiento de conocimiento en documentos escritos en lenguaje natural.

En este capítulo, se presenta el marco experimental diseñado para comprobar la efectividad del esquema de anotación descrito en el capítulo 2 y del modelo ontológico y la propuesta de solución presentados en el capítulo 3.

4.1. Marco experimental

En esta investigación solo se trabajará con los artículos del corpus de *Medline* en español, estos son procesados para eliminar las marcas específicas de *HTML*[6] y ser divididos en oraciones. Luego de ser anotadas son mezcladas con sus respectivos artículos. Potencialmente, un artículo podrá no tener ninguna de sus oraciones anotadas o estarlo completamente.

Las tablas 4.1 y 4.2 muestran algunas estadísticas acerca de este corpus y de las oraciones anotadas pertenecientes al mismo.

Estos resultados son extraídos usando *python* [25] como lenguaje de programación y los paquetes *nltk* [17] y *spacy* [6] para el procesamiento del lenguaje natural.

[6]Siglas en inglés de ***H**yper**T**ext **M**arkup **L**anguage*, un lenguaje de marcado usado en la elaboración de páginas web.

Métrica	**Total**	
Oraciones	999	
Conceptos	6,324	% conceptos
`Concept`	3,914	≈ 61.89
`Action`	1,661	≈ 26.27
`Reference`	213	≈ 3.37
`Predicate`	536	≈ 8.47
Relaciones	5,925	% relaciones
`Subject`	859	≈ 14.5
`Target`	1,688	≈ 28.49
`Domain`	346	≈ 5.84
`Argument`	333	≈ 5.62
`Is-a`	570	≈ 9.62
`Part-of`	95	≈ 1.6
`Same-as`	124	≈ 2.09
`Has-property`	168	≈ 2.84
`Causes`	381	≈ 6.43
`Entails`	170	≈ 2.87
`In-time`	154	≈ 2.6
`In-place`	384	≈ 6.48
`In-context`	653	≈ 11.02
Atributos	559	% atributos
`Negated`	160	≈ 28.62
`Uncertain`	262	≈ 46.87
`Diminished`	17	≈ 3.04
`Emphasized`	120	≈ 21.47

Tabla 4.1: Estadísticas del corpus anotado.

Estas cifras son las estadísticas del corpus sin haber aplicado el algoritmo de generar la base de conocimiento. Es decir, solo tomando los conceptos, atributos y relaciones que se encuentran en los documentos anotados.

Cabe destacar que estas cifras pueden variar una vez ejecutado el algoritmo, ya sea por la creación de nuevas entidades, como se vio en la sección 3.3.2 o por el hecho de normalizar las palabras o frases, y de esta forma, muchos conceptos pueden llegar a ser la misma entidad.

Métrica	***Medline***	**Anotado**	**% anotado**
Artículos	1,013	25*	≈ 2.47
Oraciones	12,830	999	≈ 7.79
Promedio de oraciones por artículo	≈ 13	≈ 40	≈ 307.69
Menor cantidad de oraciones en un artículo	2	39	1,950
Artículos con la menor cantidad de oraciones	9	1	≈ 11.11
Mayor cantidad de oraciones en un artículo	65	40	≈ 61.54
Artículos con la mayor cantidad de oraciones	1	24	2,400
Palabras	191,256	14,529	≈ 7.6
Promedio de palabras por artículo	≈ 189	≈ 581	≈ 307.41
Promedio de palabras por oración	≈ 15	≈ 15	100
Menor cantidad de palabras en un artículo	33	489	≈ 1,481.82
Artículos con la menor cantidad de palabras	1	1	100
Menor cantidad de palabras en una oración	1	4	400
Oraciones con la menor cantidad de palabras	87	1	≈ 1.15
Mayor cantidad de palabras en un artículo	1,199	671	≈ 55.96
Artículos con la mayor cantidad de palabras	1	1	100
Mayor cantidad de palabras en una oración	258	46	≈ 17.83
Oraciones con la mayor cantidad de palabras	1	1	100

*Esta cifra no son artículos en sí, sino archivos, los cuales pueden contener oraciones de varios artículos.

Tabla 4.2: Estadísticas del corpus tomado de *Medline* y del anotado.

4.2. Resultados computacionales

En la figura 4.1 se evidencia una relación entre el grado de salida de los nodos y la cantidad de estos que tienen un grado específico. Esto representa la cantidad de relaciones como las explicadas en la sección 3.3.2 en las que un nodo es parte izquierda (referenciado a través de `Arg1`).

En la figura 4.2 se evidencia una relación entre el grado de salida de los nodos y la cantidad de estos que tienen un grado específico. Esto representa la cantidad de relaciones como las explicadas en la sección 3.3.2 en las que un nodo es parte derecha (referenciado a través de `Arg2`).

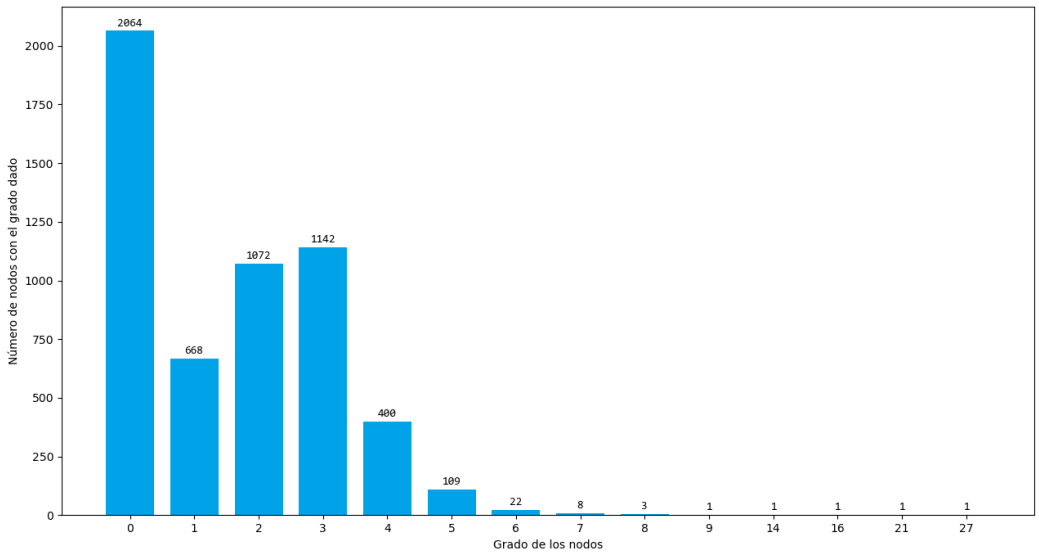


Figura 4.1: Grado de salida de los nodos del grafo.

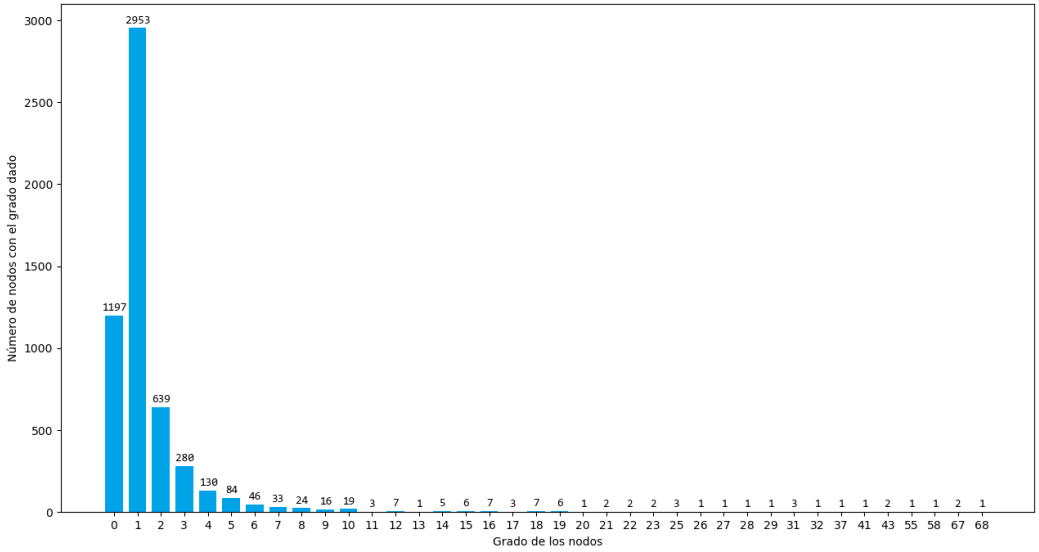


Figura 4.2: Grado de entrada de los nodos del grafo.

En la figura 4.3 se evidencia una relación entre el grado de salida de los nodos y la cantidad de estos que tienen un grado específico, pero esta vez agrupados por su rol semántico. Esto representa la cantidad de relaciones como las explicadas en la sección 3.3.2 en las que un nodo es parte izquierda (referenciado a través de `Arg1`).

En la figura 4.4 se evidencia una relación entre el grado de salida de los nodos y la cantidad de estos que tienen un grado específico, pero esta vez agrupados por su rol semántico. Esto representa la cantidad de relaciones como las explicadas en la sección 3.3.2 en las que el nodo es parte derecha (referenciado a través de `Arg2`).

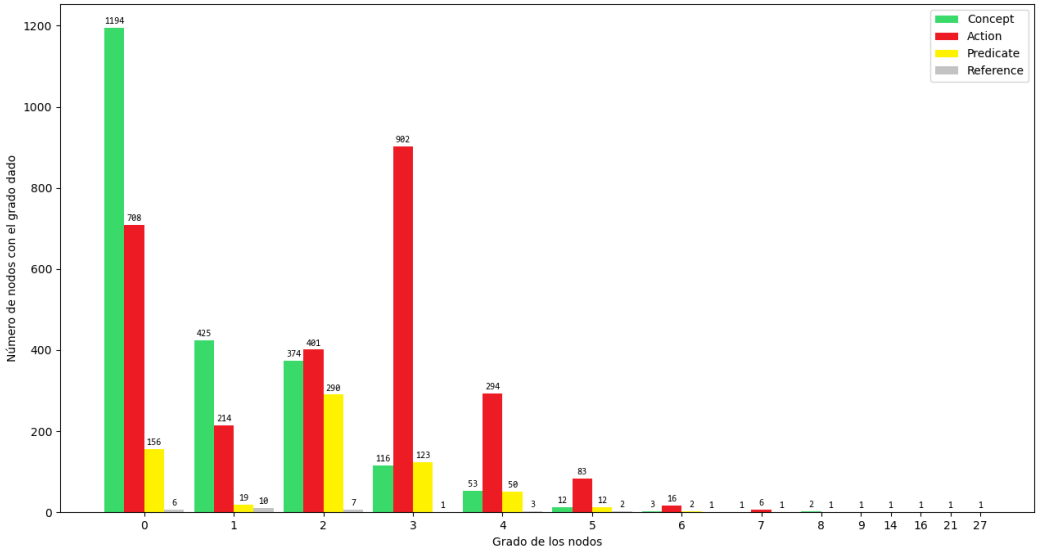


Figura 4.3: Grado de salida de los nodos del grafo por rol.

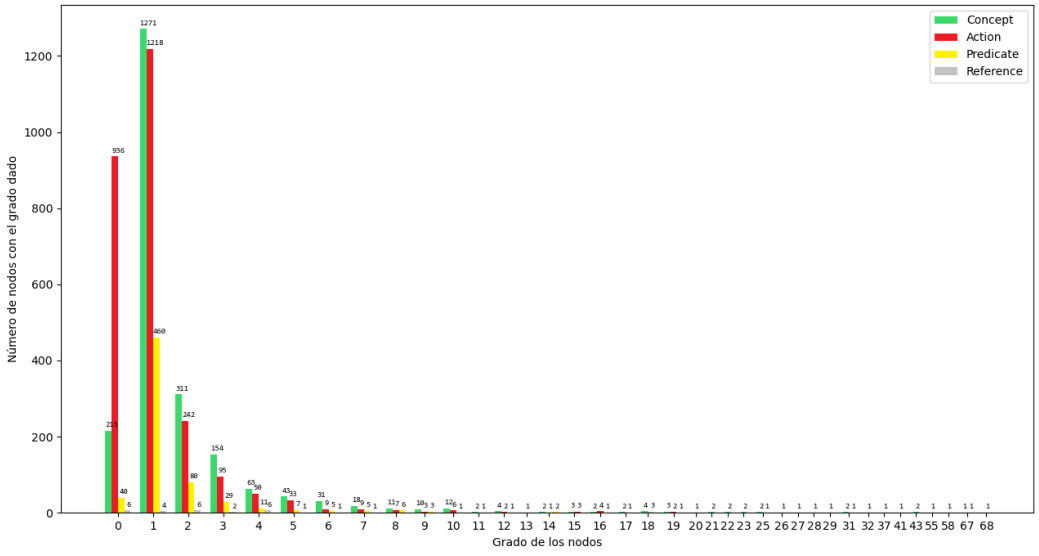


Figura 4.4: Grado de entrada de los nodos del grafo por rol.

En la tabla 4.3 se aprecia la cantidad de nodos y aristas según su tipo y a la misma vez se comparan los resultados obtenidos sin normalizar y normalizando las palabras respectivamente. La última columna representa el porcentaje de disminución en cada fila luego de normalizar.

Métrica	**Sin normalizar**	**Normalizando**	**% disminuido**
Conceptos	5,493	4,935	≈ 10.16
`Concept`	2,181	1,969	≈ 9.72
`Action`	2,625	2,335	≈ 11.05
`Reference`	35	21	40
`Predicate`	652	610	≈ 6.44
Relaciones	8,682	8,623	≈ 0.68
`Concept`	530	525	≈ 0.94
`Reference`	9	9	0
`Action`	1,902	1,875	≈ 1.42
`Subject`	923	922	≈ 0.11
`Target`	1,572	1,568	≈ 0.25
`Predicate`	501	496	≈ 1
`Domain`	298	296	≈ 0.67
`Argument`	308	307	≈ 0.32
`Is-a`	492	481	≈ 2.24
`Part-of`	89	89	0
`Same-as`	231	231	0
`Has-property`	143	143	0
`Causes`	360	360	0
`Entails`	200	199	0.5
`In-time`	154	154	0
`In-place`	361	360	≈ 0.28
`In-context`	609	608	≈ 0.16
Atributos	359	329	≈ 8.36
`Negated`	111	94	≈ 15.32
`Uncertain`	150	141	6
`Diminished`	83	80	≈ 3.61
`Emphasized`	15	14	≈ 6.67

Tabla 4.3: Estadísticas del grafo de conocimiento.

Como se ha podido observar anteriormente, una de las principales funciones de las ontologías y los grafos de conocimiento es la extracción de conocimiento implícito, el cual, como lo dice su nombre, no está representado explícitamente en el corpus.

En la tabla 4.4 puede verse un ejemplo de esto en el corpus usado en esta investigación.

Documento	**Relación explícita**
`cirugía.ann`	«`cirugía de corazón` *is-a* `operación`»
`hígado graso.ann`	«`operación` *is-a* `procedimiento médico`»
	Relación implícita
	«`cirugía de corazón` *is-a* `procedimiento médico`»

Tabla 4.4: Ejemplo de extracción de conocimiento implícito.

Para el problema en general de aprendizaje de ontologías se pueden llevar a cabo varias métricas de evaluación y metodologías, por ejemplo OntoRand [2] y OntoMetric [12]. Uno de los enfoques comúnmente utilizados en la literatura para la evaluación de ontologías [18] es la *evaluación basada en datos* (del inglés *data driven evaluation*) [1].

La evaluación *basada en datos* puede ser llevada a cabo comparando las entidades y relaciones en una ontología dado un corpus de datos representativos del mismo dominio, pero que no es usado durante la creación de la base de conocimiento.

La base de conocimientos creada puede ser evaluada teniendo en cuenta el número de entidades y relaciones pertenecientes a este nuevo corpus y que están presentes en ella.

Este enfoque ha sido usado para comparar relativamente diferentes ontologías creadas por expertos basándose en el mismo corpus y decidir qué ontología provee el mejor "ajuste" a este. [3] Sin embargo, obtener una métrica absoluta del "ajuste" entre una ontología y un corpus es una tarea difícil, principalmente porque se desconoce a priori cuál sería el mejor "ajuste".

La tabla 4.5 muestra la división que se hizo entre las oraciones del corpus anotado y los resultados alcanzados, en promedio, luego de 250 corridas de cada división.

% de división	70	75	80	85	90	95
oraciones	959	959	959	959	959	959
oraciones de entrenamiento	671	719	767	815	863	911
oraciones de verificación	288	240	192	144	96	48
entidades en el corpus	3,725	3,938	4,135	4,345	4,544	4,739
entidades de verificación	1,888	1,621	1,353	1,058	746	405
coincidencia de entidades	678	623	554	468	355	209
% de coincidencia de entidades	35.91	38.43	40.95	44.23	47.59	51.6
nuevas entidades	1,210	998	799	590	391	196
% de nuevas entidades	64.09	61.57	59.05	55.77	52.41	48.4
relaciones en el corpus	6,139	6,560	6,965	7,390	7,802	8,213
relaciones de verificación	2,728	2,283	1,845	1,390	932	469
coincidencia de relaciones	244	220	187	156	111	59
% de coincidencia de relaciones	8.94	9.64	10.14	11.22	11.91	12.58
nuevas relaciones por nuevas entidades	2,416	2,003	1,609	1,195	795	397
% de nuevas relaciones por nuevas entidades	88.56	87.74	87.21	85.97	85.3	84.65
nuevas relaciones en entidades existentes	68	60	49	39	26	13
% de nuevas relaciones en entidades existentes	2.5	2.62	2.65	2.81	2.79	2.77
% de coincidencia de relaciones válidas[7]	78.21	78.57	79.24	80	81.02	81.94

Tabla 4.5: Ejemplo de extracción de conocimiento implícito.

[7]Las relaciones válidas son aquellas entre entidades ya existentes en el corpus, pues si la entidad no existe es obvio que habrá que crearla y por tanto, la relación será un fallo seguro a la hora de comprobar si existe o no en el grafo de conocimiento de entrenamiento.

4.3. Discusión

Todos los nodos del grafo de conocimiento participan en al menos una relación, pero como se aprecia en las figuras 4.1, 4.2, 4.3 y 4.4, hay muchos nodos que tienen un grado bajo.

En el caso en que la gráfica muestra la cantidad de nodos con grado cero, esto viene dado o bien porque ese nodo no tiene relaciones de salida y en este caso tendría grado de salida cero o bien no tiene relaciones de entrada y por tanto grado de entrada cero. El hecho de que pocos nodos tengan un alto grado viene dado porque usualmente los nodos más grandes y con más palabras, al ser conocimiento más específico, pocos de ellos participan en un mayor número de relaciones.

Los roles semánticos *Concept* y *Action* que participan en una mayor cantidad de relaciones en este grafo de conocimiento son «`persona`» y «`tratamiento`» respectivamente. Dado que se trabajó con un corpus de documentos médicos, este resultado cobra sentido pues esas palabras son ampliamente empleadas en este medio.

En la tabla 4.3 se muestra el resultado de un grafo de conocimiento utilizando palabras o frases sin normalizar y normalizadas. Normalizar las palabras no solo reduce la cantidad de nodos y aristas en el grafo sino que también potencialmente aumenta la cantidad de conocimiento implícito que puede ser extraído. Esto sucede debido a que al fusionar nodos, la cantidad de caminos en el grafo de conocimiento aumentan. Por otra parte, todas las relaciones explícitamente escritas en el corpus representan dos nodos y una arista entre estos. Por tanto, si dos nodos no tienen aristas entre ellos, pero existe un camino que los conecta, esto es conocimiento implícito descubierto a través del grafo.

El hecho de normalizar implica, por ejemplo, que todas las conjugaciones de un mismo verbo deben resultar en el propio verbo sin conjugar. Lo mismo sucede para el resto de palabras del idioma. Mientras mayor sea la cantidad de palabras normalizadas, hay una mejor representación en el grafo de su conocimiento expuesto en el corpus. Obviamente, es de vital importancia la normalización de familias de palabras a su primitiva, para poder dar continuidad al significado que ellas representan, puesto que no tiene sentido normalizar «`glóbulos rojos`» a «`globo rojo`» o «`glóbulo blanco`».

Una deficiencia clara sucedió a la hora de hallar los resultados de la tabla 4.5. Para este tipo de evaluación, lo ideal es tener un grafo de conocimiento formado a partir de una ontología y de un corpus preferentemente grande. A su vez, el grafo

debe ser revisado con otro corpus perteneciente al mismo tema y de mediano o gran tamaño. Muchas veces esto es difícil de lograr, y en efecto, es lo que sucedió.

Para llevar a cabo esta tarea, se dividieron las oraciones anotadas en dos grupos, un grupo de *training* (*entrenamiento* en español) con el cual se realizará el grafo de conocimiento y un grupo de *testing* (*verificación* en español), con el que se revisará la existencia de las anotaciones de texto y las relaciones respecto a las que ya están construidas en el grafo. Como puede verse en dicha tabla, a medida que el corpus de entrenamiento tiene más entidades y relaciones, se espera una mayor coincidencia respecto a nueva información que se quiera añadir a la base de conocimiento ya formada.

Conclusiones

Esta investigación propone un conjunto de elementos orientados al descubrimiento de conocimiento en textos del lenguaje natural. La propuesta se centra en el idioma español y el dominio de la salud, pero es generalizable en ambos aspectos. Entre las contribuciones fundamentales de esta investigación destacan:

(1) La definición de un modelo de anotación de propósito general que logra capturar los rasgos semánticos más relevantes contenidos en documentos de texto plano. El mismo es usado como base en la construcción de la ontología propuesta.

(2) La definición de un formato de anotación de archivos para el esquema conceptual previamente definido.

(3) Se diseñó una propuesta de ontología donde se puede representar un corpus de documentos escritos en lenguaje natural.

(4) Se implementó un algoritmo computacional para representar un corpus anotado como grafo de conocimiento a través de dicha ontología.

A menudo, una ontología de un dominio no es un objetivo en sí misma. Desarrollarla es similar a definir un conjunto de datos y su estructura para que los utilicen otros programas. Los métodos de resolución de problemas, las aplicaciones independientes del dominio y los usuarios, a menudo las utilizan como datos, en conjunto con bases de conocimiento creadas a partir de las mismas. Por ejemplo, en esta investigación se desarrolla una ontología de dominio médico, la cual se puede utilizar como base para algunas aplicaciones que ofrezcan un conjunto de herramientas de gestión de la salud.

En los resultados se demostró el descubrimiento de conocimiento implícito en el corpus. Esto trae consigo disímiles ventajas, desde el propio descubrimiento de

este conocimiento, hasta la interpretación y aprendizaje de un gran número de textos escritos en lenguaje natural, en apenas segundos, mediante el uso de un equipo de cómputo y las propuestas ofrecidas en esta investigación. El conocimiento extraído podría ser usado posteriormente por especialistas en el tema o usuarios, y de esta manera ahorrar el tiempo que tomaría la lectura e interpretación del propio corpus usado para esta tarea.

Teniendo en cuenta que un humano debe tener una gran capacidad de memorización para poder recordar todo lo aprendido en un corpus de documentos, la traba que puede ocasionar tenerlo escrito en un idioma que no se domine, y el factor de no olvidar lo aprendido de él al pasar el tiempo, es clave la utilización de un equipo de cómputo para la creación de la base de conocimiento respectiva al corpus. La misma puede ser fácilmente guardada, leída y usada en el propio sistema o incluso en otros, aportando gran versatilidad al uso de las técnicas mostradas en este estudio.

El descubrimiento automático de conocimiento en el dominio médico tiene especial importancia, pues permitiría identificar interacciones ocultas en la literatura. Además, a pesar de que los recursos médicos disponibles en idioma español son abundantes, los recursos necesarios para la creación de sistemas de extracción automática son más escasos que en otros idiomas, por lo cual la construcción de una ontología y un grafo de conocimiento basado en un corpus del propio dominio, constituyen un hecho relevante para el desarrollo de nuevos sistemas y la continuación de esta investigación en un futuro.

Recomendaciones

A pesar de que esta investigación está orientada hacia el descubrimiento de conocimiento en documentos médicos y del idioma español, el modelo de anotación y la ontología propuesta son de propósito general. Esto permite su aplicación en otros dominios e idiomas.

Se propone la anotación de corpus de otros dominios en el modelo de anotación definido en este estudio. Al mismo tiempo, tendría gran connotación la creación de grafos de conocimiento mediante la utilización de estos corpus y basados en el modelo ontológico ofrecido.

Se propone comprobar la efectividad de la propuesta de solución ofrecida, pues como se mencionó anteriormente, una deficiencia clara a la hora de calcular las estadísticas expuestas en la tabla 4.5 es que se usó un corpus muy pequeño para crear el grafo de conocimiento y la no existencia de uno independiente pero del mismo dominio y anotado con el formato de modelo propuesto para la posterior validación de la base de conocimiento creada.

La resolución de referencias y correferencias mejoraría en gran medida el descubrimiento de conocimiento implícito en el grafo y debe mejorar los resultados obtenidos. Esta es una tarea que usualmente se intenta resolver usando inteligencia artificial y es un reto que se propone para trabajo futuro, dando continuidad a la línea de investigación presentada en este trabajo.

Otra de las propuestas consideradas es la creación de un grafo de conocimiento a partir de un corpus de dominio específico, siguiendo la línea de investigación de este estudio. A su vez, fomentar el análisis de este corpus en un grupo de expertos en el dominio, y de esta manera corroborar cuán relevante es el conocimiento implícito descubierto a través del grafo resultante en comparación al extraído por los especialistas.

Además se propone la creación de una aplicación para computadoras, móviles

y/o páginas web, la cual podría ofrecer sugerencias de enfermedades dados los síntomas especificados por el usuario, y a su vez, tratamientos para las mismas. Esto sería posible mediante la utilización del grafo de conocimiento creado a partir del corpus usado en esta investigación, el cual es de dominio médico.

Bibliografía

[1] BRANK, J., MLADENIC, D., AND GROBELNIK, M. A survey of ontology evaluation techniques. (Citado en la página 46).

[2] BRANK, J., MLADENIC, D., AND GROBELNIK, M. Gold standard based ontology evaluation using instance assignment. In *Workshop on Evaluation of Ontologies for the Web, EON* (2006). (Citado en la página 46).

[3] BREWSTER, C., ALANI, H., DASMAHAPATRA, S., AND WILKS, Y. Data driven ontology evaluation, 2004. (Citado en la página 46).

[4] BRICKLEY, D., AND GUHA, R. V. Resource Description Framework (RDF) Schema Specification. World Wide Web Consortium: http://www.w3.org/TR/PR-rdf-schema. (Citado en la página 5).

[5] CONSUEGRA AYALA, J. P. Descubrimiento de Conocimiento en Documentos Clínicos: Un Enfoque Basado en Aprendizaje Profundo, Dec. 2019. (Citado en la página 29).

[6] EXPLOSION AI. spaCy. `https://spacy.io`. (Citado en la página 40).

[7] GRUBER, T. R. A Translation Approach to Portable Ontology Specification. *Knowledge Acquisition 5* (1993), 199–220. (Citado en las páginas 5 y 6).

[8] GUARINO, N. Formal ontology in information systems. Proceedings of the first international conference (FOIS'98), June 1998. (Citado en la página 2).

[9] HEARST, M. A. Automatic acquisition of hyponyms from large text corpora. *Association for Computational Linguistics* (1992), 539–545. (Citado en la página 15).

[10] HENDLER, J., AND MCGUINNESS, D. L. The DARPA Agent Markup Language. *IEEE Intelligent Systems 16*, 6 (2000), 67–73. (Citado en la página 5).

[11] HUMPHREYS, B. L., AND LINDBERG, D. A. B. The UMLS project: making the conceptual connection between users and the information they need. *The UMLS project: making the conceptual connection between users and the information they need. Bulletin of the Medical Library Association 81*, 2 (1993), 170. (Citado en la página 6).

[12] LOZANO-TELLO, A., AND GÓMEZ-PÉREZ, A. Ontometric: A method to choose the appropiate ontology. (Citado en la página 46).

[13] MCGUINNESS, D. L., FIKES, R., RICE, J., AND WILDER, S. An Environment for Merging and Testing Large Ontologies. (Citado en la página 7).

[14] MCGUINNESS, D. L., AND WRIGHT, J. Conceptual Modeling for Configuration: A Description Logic-based Approach. *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing - special issue on Configuration* (1998). (Citado en la página 7).

[15] MEDLINEPLUS. `https://medlineplus.gov`. (Citado en la página 3).

[16] MUSEN, M. A. Dimensions of knowledge sharing and reuse. *Computers and Biomedical Research 25* (1992), 435–467. (Citado en la página 6).

[17] NATURAL LANGUAGE TOOLKIT. NLTK. `https://www.nltk.org`. (Citado en la página 40).

[18] PETASIS, G., KARKALETSIS, V., PALIOURAS, G., KRITHARA, A., AND ZAVITSANOS, E. Ontology population and enrichment: State of the art. (Citado en la página 46).

[19] PIAD-MORFFIS, A., ALMEIDA-CRUZ, Y., GUTIÉRREZ, Y., AND MUÑOZ, R. A Computational Ecosystem to Support eHealth Knowledge Discovery Technologies in Spanish. *Journal of Biomedical Informatics* (2020), 103517. (Citado en la página 29).

[20] PIAD-MORFFIS, A., ALMEIDA-CRUZ, Y., GUTIÉRREZ, Y., AND MUÑOZ, R. [dataset] eHealth-KD v2, Mar. 2020. (Citado en la página 30).

[21] PIAD-MORFFIS, A., ESTEVEZ-VELARDE, S., ALMEIDA-CRUZ, Y., GUTIÉRREZ, Y., MUÑOZ, R., AND MONTOYO, A. Overview of the eHealth Knowledge Discovery Challenge at IberLEF 2020. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020)* (2020). (Citado en la página 29).

[22] PIAD-MORFFIS, A., ESTEVEZ-VELARDE, S., CONSUEGRA-AYALA, J. P., ALMEIDA-CRUZ, Y., GUTIÉRREZ, Y., MUÑOZ, R., AND MONTOYO, A. Overview of the eHealth Knowledge Discovery Challenge at IberLEF 2019. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019). CEUR Workshop Proceedings, CEUR-WS.org* (2019). (Citado en la página 29).

[23] PISANELLI, D. M., GANGEMI, A., AND STEVE, G. Ontologies and information systems: the marriage of the century? *New Trends in Software Methodologies, Tools and Techniques* (2002), 125–133. (Citado en la página 1).

[24] PRICE, C., AND SPACKMAN, K. SNOMED clinical terms. *BJHC&IM-British Journal of Healthcare Computing & Information Management 17*, 3 (2000), 27–31. (Citado en la página 6).

[25] PYTHON SOFTWARE FOUNDATION. Python. `https://www.python.org/downloads/release/python-382`. (Citado en la página 40).

[26] ROTHENFLUH, T. R., GENNARI, J., ERIKSSON, H., PUERTA, A., TU, S., AND MUSEN, M. Reusable ontologies, knowledge-acquisition tools, and performance systems: PROTÉGÉ-II solutions to Sisyphus-2. *International Journal of Human-Computer Studies 44* (1996), 303–332. (Citado en la página 7).

[27] RUSSELL, B. 2 ed. W. W. Norton & Company, Berlin, 1996. Reprint, first published in 1903. (Citado en la página 17).

[28] UNITED NATIONS DEVELOPMENT PROGRAM. United Nations Standard Products and Services Code. `https://www.unspsc.org`. (Citado en la página 6).

[29] WEISSTEIN, E. W. Grelling's Paradox. `https://mathworld.wolfram.com/GrellingsParadox.html`. (Citado en la página 17).