Anales de Lingüística. Segunda época
n.7 – AGO-DIC 2021. Mendoza, Argentina ISSN 0325-3597 (impreso) - ISSN 2684-0669 (digital)
https://revistas.uncu.edu.ar/ojs3/index.php/analeslinguistica
pp. 175-201

KIND: Un proyecto de inducción automática de taxonomías léxicas

KIND: A project for the automatic induction of lexical taxonomies

Rogelio Nazar[1] https://orcid.org/0000-0002-8853-1353

Instituto de Literatura y Ciencias del Lenguaje
Pontificia Universidad Católica de Valparaíso
rogelio.nazar@pucv.cl

Resumen

Este artículo presenta una descripción del Proyecto Kind, un algoritmo para inducción automática de taxonomías léxicas a partir de corpus. La inducción de taxonomías consiste en el descubrimiento de relaciones de hiperonimia entre pares sustantivos, ya sea mono o poliléxicos, y en la integración de estos pares en estructuras mayores. La metodología propuesta es fundamentalmente estadística y tiene mínimo requerimiento de recursos lingüísticos, característica que facilita la reproducción de experimentos en distintas lenguas. Las lenguas con las que se ha experimentado hasta ahora son castellano, inglés y francés. La implementación del algoritmo y un demostrador en línea se encuentran disponibles como código abierto en el sitio web del proyecto[1].

Palabras clave: inducción automática de taxonomías léxicas, relaciones de hiperonimia, descubrimiento de hiperonimia, lexicografía computacional, semántica léxica

Abstract

This paper presents a description of the Kind Project, an algorithm for automatic induction of lexical taxonomies from corpora. Taxonomy induction consists of the discovery of hypernymy relations between single or multiword noun pairs, and the

[1] Recibido: 30.11.2021 | Aceptado: 07.01.2021

integration of these pairs into larger structures. The proposed methodology is fundamentally statistical and the requirement of linguistic resources is minimal, a characteristic that facilitates the reproduction of experiments in different languages. The languages for which results have been obtained so far are Spanish, English and French. The implementation of the algorithm and an online demo are available as open source on the projects' website[1].

Keywords: automatic induction of lexical taxonomies, hypernym relations, hypernymy discovery, computational lexicography, lexical semantics

Introducción

A pesar de ser un problema bastante antiguo en la historia del procesamiento del lenguaje natural (PLN), la inducción automática de taxonomías léxicas continúa dejando un amplio margen de mejora. El presente artículo describe uno de los intentos por hacer frente a dicho problema, el proyecto Kind[1], que en sí mismo tiene ya cierta antigüedad también, puesto que es el resultado de más de 10 años de trabajo colaborativo de investigadores de distintos países.

Una taxonomía léxica es una ordenación de las palabras de una lengua (típicamente, de sustantivos mono o polilexemáticos) dada por la relación semántica de hiperonimia. En una estructura de este tipo, a cada palabra le corresponde una cadena taxonómica que la vincula con las que denotan conceptos más generales. Así, si tenemos, por ejemplo en el caso del inglés, el sustantivo de entrada *airplane,* la cadena correspondiente sería la siguiente:

1. -- > *entity*
2> physical object
3> inanimate
4> artifact
5> vehicle

[1] http://www.tecling.com/kind

6> aircraft
7> airplane

Las taxonomías léxicas, en tanto estructuras jerárquicas, no poseen ciclos y están constituidas por relaciones asimétricas. Esto es, un *avión* es un tipo de *vehículo,* pero un *vehículo* no puede ser un tipo de *avión*. También poseen la propiedad transitiva, es decir que, si un *avión* es un tipo de *vehículo* y a su vez un *vehículo* es un tipo de *artefacto*, entonces un *avión* es un tipo de *artefacto*.

El objetivo del sistema Kind es construir este tipo de estructuras de manera automática y sin información previa, es decir, sin necesidad de que un programador ingrese en el sistema una base de conocimiento. La ventaja principal de esta propuesta metodológica, en comparación con otras que se mencionan más adelante en este artículo, es que se trata de un diseño relativamente simple. Uno de los aspectos centrales en este sentido es que no está basado en aprendizaje automático. Si se puede hablar de aprendizaje, sería en todo caso un aprendizaje autónomo o no guiado. Esto representa una ventaja ya que los sistemas de aprendizaje automático requieren gran cantidad de material de entrenamiento, y este material no puede contener errores, ya que estos producirían un sesgo en el aprendizaje.

Frente a los sistemas de aprendizaje automático, el que se presenta aquí es por tanto preferible por ser más simple, más económico en términos conceptuales y computacionales, y a su vez más robusto con respecto a condiciones iniciales desfavorables, como puede ser la insuficiente cantidad de recursos en una determinada lengua. Precisamente, una ventaja importante de un sistema de clasificación que no requiere de una etapa de entrenamiento es que facilita la reproducción de los experimentos en distintas lenguas. El algoritmo implementado en Kind utiliza muy pocos recursos externos: solo requiere un corpus de gran tamaño, un etiquetador morfosintáctico e, idealmente, un listado breve de

patrones lexicosintácticos, todos ellos elementos básicos en cualquier operación de PLN.

Dicho en términos más generales, lo que se pretende en este proyecto es poner en orden todos los sustantivos de la lengua, es decir, encontrar el orden que subyace en el aparente desorden de las palabras tal como se presentan en la superficie del corpus, y hacer esto solamente por medio de ecuaciones matemáticas. Evidentemente, todo esto representa una idea atractiva desde un punto de vista lingüístico.

En su estado actual, Kind funciona en castellano, inglés y francés, y la idea es ir incluyendo paulatinamente distintas lenguas. Las taxonomías que posee van creciendo de manera autónoma y en un ciclo virtuoso, ya que mientras mayor es la base de conocimiento, más precisas resultan las clasificaciones. El sistema es, además, completamente abierto, ya que el código fuente se encuentra disponible en la web del proyecto, que además ofrece un demostrador en línea y la posibilidad de navegar por las taxonomías ya creadas. La implementación del algoritmo consiste en un único *script* en el lenguaje Perl, sin dependencias.

1. La inducción de taxonomías léxicas en PLN

Con distintos objetivos, son varios los intentos que se han llevado a cabo para el desarrollo de taxonomías léxicas. El término que se ha utilizado más frecuentemente, particularmente en el ámbito de la inteligencia artificial, es el de *ontologías*. Pero, en estricto rigor, ambos términos no refieren a lo mismo, puesto que una ontología es una estructura de conceptos, no de palabras. Esto es, las ontologías son estructuras elaboradas desde un punto de vista onomasiológico en lugar de semasiológico (Sager 1990) y, además, pueden incluir otro tipo de relaciones aparte de hiperonimia.

La mayor parte de los primeros esfuerzos en esta línea, dentro de los campos de la inteligencia artificial y de la lingüística computacional, fueron bases de conocimiento elaboradas de manera manual. Algunos ejemplos conocidos de estos proyectos, aunque muy distintos entre sí, son las

ontologías CYC (Lenat 1995), WordNet (Fellbaum 1998), SNOMED-CT (Cornet y de Keizer 2008) o DOLCE (Borgo y Masolo 2009), entre otras.

Además de estos esfuerzos por construir estas taxonomías u ontologías de manera manual, ya desde la década de 1970 han existido esfuerzos para construirlas de manera automatizada. Los primeros intentos fueron a partir del procesamiento de datos de diccionarios electrónicos (Calzolari 1973; Chodorow Byrd y Heidorn 1985). Las estrategias que usan Wikipedia para extraer estas relaciones, como BabelNet (Navigli y Ponzetto 2010), podrían considerarse una extensión de este enfoque.

En una etapa posterior, se trasladó el mismo tipo de procedimiento hacia la extracción de relaciones de hiperonimia a partir de corpus. Esta transición se llevó a cabo por medio de la búsqueda de patrones lexicosintácticos tales como *X es un tipo de Y*, entre otros. (Hearst, 1992; Snow, Jurafsky y Ng, 2006).

En paralelo, otra dirección de trabajo con corpus estuvo basada en estrategias de análisis cuantitativo, más específicamente aplicando el criterio de la similitud distribucional o paradigmática (Pereira, Tishby y Lee 1993; Grefenstette 1994; Lin 1998; Weeds y Weir 2003, Bullinaria 2008). Con estas estrategias, los investigadores buscaban no tanto el establecimiento de relaciones de hiperonimia como el agrupamiento de cohipónimos, es decir, palabras que tienen un hiperónimo común.

En la actualidad existe un número cada vez mayor de publicaciones al respecto, y la tendencia general es la de una progresiva hibridación, aprovechando las ventajas y fortalezas de cada una para una óptima complementariedad (Panchenko et al. 2016; Shwartz, Santus y Schlechtweg 2017; Sarkar, McCrae y Buitelaar 2018). El método que se describe en este artículo encaja en esta tendencia ya que presenta una combinación de distintas estrategias, pero muchas de estas son nuevas y resulta novedosa también la manera de combinarlas.

2. El proyecto Kind

2.1. Historia del proyecto

El primer antecedente del proyecto fue el análisis de las relaciones asimétricas de coocurrencia léxica para el descubrimiento de hiperónimos (Nazar 2010; Nazar, Vivaldi y Wanner 2012), descrito en el apartado 2.2.4 de este artículo. En paralelo, se realizaron investigaciones en el campo de la asociación paradigmática (Nazar y Renau 2015), que serían el antecedente del módulo descrito en el apartado 2.2.5. También se realizaron tentativas de establecimiento de relaciones de hiperonimia por medio del análisis estadístico de diccionarios (Nazar y Janssen 2010, Renau y Nazar 2012), aunque esta línea se abandonó por la necesidad de recursos que implica. Finalmente, se ensayó con distintas formas de ensamblaje de las estrategias en un único algoritmo (Nazar y Renau 2016; Nazar et al. 2020). Estas representan etapas previas al desarrollo actual, que parece ser el definitivo.

El proyecto ha tenido, además, distintas fuentes de financiamiento gubernamental. La primera fue el Proyecto Fondecyt[1] 11140686 en 2014, seguido del Proyecto Ecos Sud-Conicyt[2] C16H02 en 2016 y, finalmente, el Proyecto Fondecyt Regular[3] 1191481, en 2019.

2.2. Estado actual del algoritmo

Kind está compuesto por una serie de módulos que se ejecutan en paralelo: un módulo de asociación asimétrica (apartado 2.2.4), otro de similitud paradigmática (apartado 2.2.5), uno de similitud morfológica (apartado

[1] Proyecto Fondecyt Iniciación 11140686: "Inducción automática de taxonomías de sustantivos generales y especializados a partir de corpus textuales desde el enfoque de la lingüística cuantitativa". Investigador principal: Rogelio Nazar. Duración: 2014-2017.

[2] Proyecto Ecos Sud-Conicyt C16H02: "Inducción automática de taxonomías del español y el francés mediante técnicas cuantitativas y estadística de corpus". Directores: Irene Renau y Rafael Marín. Duración: 2016-2019.

[3] Proyecto Fondecyt Regular 1191481: "Inducción automática de taxonomías de marcadores discursivos a partir de corpus multilingües". Investigador principal: Rogelio Nazar. Duración: 2019-2021.

2.2.6), uno de reglas morfosemánticas (apartado 2.2.7), uno de patrones de lexicosintácticos (apartado 2.2.8), uno de detección de núcleos (apartado 2.2.9) y, finalmente, uno de poda para la detección de enlaces incorrectos (apartado 2.2.10). El trabajo de estos módulos está coordinado por un procesador central que es el encargado de producir una cadena taxonómica completa para un sustantivo de entrada (apartado 2.2.11) e integrarla a una estructura de base, descrita en el apartado siguiente.

2.2.1. La estructura de base

El punto de partida del proceso de inducción de taxonomías es una estructura de base consistente en una disposición en forma de árbol de alrededor de 300 sustantivos que designan los conceptos más generales de una lengua, comenzando por aquellos como *entidad*, *propiedad*, *evento*, *grupo*, etc. Esta estructura está inspirada en el proyecto de la Ontología CPA[1] (Hanks y Pustejovsky 2005), pero ha sido libremente adaptada para los propósitos del presente proyecto, además de traducida al castellano y al francés.

2.2.2. Corpus: indización y extracción de concordancias

Como la mayoría de los módulos están basados en corpus, es necesario contar con un corpus de gran tamaño para poder operar. Los corpus de los que se extrajo el material fueron las versiones en francés, inglés y castellano de la serie TenTen (Jakubíček et al. 2013), que cuentan con aproximadamente 10^{10} palabras por lengua y representan actualmente la mayor muestra de texto disponible para estas y otras lenguas. Los corpus están compuestos por páginas web descargadas aleatoriamente, convertidas a texto plano, verticalizadas, lematizadas y etiquetadas con categoría gramatical.

Para extraer contextos de aparición de palabras a partir de un corpus tan grande es necesario contar con un extractor de concordancias lo suficientemente robusto. La opción utilizada en este proyecto es el

[1] http://pdev.org.uk

software Kwico[1], que es el módulo de extracción de concordancias desarrollado en el contexto de Jaguar[2], un software para explotación de corpus. En esta tarea, Kwico es rápido y resulta por ello ideal para ser utilizado con corpus de gran tamaño. Además, se encuentra también libremente disponible para la comunidad, en código abierto.

2.2.3. Construcción y ponderación de vectores

La representación de palabras y categorías de palabras como vectores es una operación que comparten muchos de los módulos. La construcción de los vectores no es idéntica en todos los casos, ya que los elementos que se utilizan como componentes varía según el módulo. Sin embargo, el proceso de selección y ponderación de los componentes es común.

En el caso de los módulos basados en aspectos distribucionales, dado un sustantivo de entrada *a*, se construye un vector $\vec{a}$ utilizando como componentes palabras (adjetivos, sustantivos y verbos) que tienen una significativa frecuencia de coocurrencia con *a* en la ventana de contexto. Para ello, se extraen las concordancias de *a* a partir del corpus y se representan como un conjunto $C(a) = \{c(a)_1, \ldots c(a)_n\}$ (con $n <= 5000$, para permitir un menor tiempo de procesamiento). La ventana de contexto es de diez palabras a cada lado del término de entrada *a* y cada contexto $C(a)_i$ se representa como una bolsa de palabras, es decir, sin orden ni repeticiones.

El vector $\vec{a}$ incluye al propio término de entrada, ya que tiende a ser la palabra más frecuente en sus propios contextos de aparición. Para simplificar, la selección de componentes está restringida a unidades monoléxicas, pero el procedimiento sería el mismo si se incluyeran secuencias de palabras. También por simplicidad, la dimensionalidad $|\vec{a}|$ se limitó a 100, un umbral definido de manera empírica.

[1] http://www.tecling.com/kwico
[2] http://www.tecling.com/jaguar

En el caso del módulo descrito en el apartado 2.2.6, en cambio, la construcción de los vectores es más sencilla, ya que los componentes representan rasgos morfológicos de las palabras de una determinada categoría semántica. Estos, a su vez, se presentan como secuencias de tres, cuatro y cinco letras al final de la palabra. Así, el segmento final *-itis* sería uno de los componentes del vector correspondiente a la categoría *enfermedad*.

Una vez construidos los vectores según el procedimiento que corresponda a cada módulo, estos son sometidos a un único proceso de ponderación que tiene por objetivo la selección de los componentes más discriminantes. Esta ponderación es común a los distintos módulos con el fin de mantener uniformidad de criterio y mayor simplicidad. La que se aplica es la que se muestra en la ecuación 1:

$$pond(F_i, T_j) = \frac{f(F_i|T_j)}{\sqrt{|T_j|}} \cdot \frac{1}{\sqrt{D(F_i)}} \quad (1)$$

Aquí, el símbolo F_i puede representar un componente cualquiera. Por ejemplo, en el caso de un rasgo morfológico, el segmento final *-itis*. A su vez, T_j puede ser una categoría semántica, como por ejemplo, *enfermedad*; f(F_i | T_j) denota la frecuencia del rasgo F_i en la categoría T_j, que viene dada por el número de hipónimos que tienen este componente en dicha categoría. Finalmente, $D(F_i)$ representa la dispersión de dicho rasgo F_i, es decir, el número de categorías semánticas en las que este rasgo está presente. Esta ponderación favorece así los rasgos que se concentran en pocas categorías y que, al mismo tiempo, son productivos en cada una de ellas. Siguiendo con el mismo ejemplo, este sería el caso de encontrar una gran proporción de palabras terminadas en el segmento *-itis* entre la población de palabras de la categoría de enfermedades, lo cual convierte a este rasgo en un componente valioso para el vector.

La selección de los mejores componentes se registra en una matriz $FM_{m,h,f}$ (ecuación 2), donde *m* representa el nombre del módulo, *h* el hiperónimo

o categoría semántica, f el componente o característica y u es una constante o parámetro definido de manera empírica. Esto se recalcula periódicamente a medida que la taxonomía se desarrolla.

$$F_i \in FM_{m,h,f} \iff pond(F_i, T_j) > u \quad (2)$$

2.2.4. Módulo de asociación asimétrica

El módulo de asociación asimétrica es posiblemente el más interesante desde un punto de vista lingüístico. Está basado en la intuición de que, en general, los sustantivos tienen una tendencia a coocurrir con sus hiperónimos de manera no recíproca (Nazar, 2010), y esto es un aspecto que no había sido observado con anterioridad.

Considérese el siguiente ejemplo para ilustrar esta idea. Si examinamos el vector de coocurrencias de una palabra como *motocicleta*, observaremos que el segundo sustantivo más frecuente después de *motor* será probablemente *vehículo.* Sin embargo, si examinamos las palabras más frecuentes en el vector de *vehículo*, encontraremos que *motocicleta* probablemente no estará entre ellas. Del mismo modo, las palabras también coocurren frecuentemente con sus cohipónimos, y entonces en el vector de coocurrencia de *motocicleta* veremos elementos frecuentes como *automóvil*, *bicicleta*, *coche*, etc., cada uno de los cuales probablemente mostrará también el mismo patrón con respecto a *vehículo*.

Habiendo observado este comportamiento, es posible derivar una estrategia que utilice vectores de coocurrencia para recuperar, para un sustantivo de entrada *a*, un vector de coocurrencia de primer orden y luego uno de segundo orden, compuesto por palabras que coocurren con las palabras que coocurren con *a*. Los hiperónimos correctos para *a* se encontrarán normalmente entre los elementos coocurrentes de primer y segundo orden más frecuentes de *a*. Podemos representar este razonamiento en una matriz:

$$\begin{bmatrix} a & b & c & \mathbf{d} & \dots \\ b & \mathbf{d} & a & e & \dots \\ c & f & g & h & \dots \\ d & i & \mathbf{d} & j & \dots \\ \dots & \dots & \dots & \dots & \end{bmatrix}$$

$$a \rightarrow d \iff d \in T$$

La primera fila representa el vector de coocurrencia de primer orden, correspondiente a la palabra de entrada *a*. La primera columna muestra las mismas palabras que en la primera fila, es decir, las palabras de primer orden de coocurrencia con *a*. Las filas siguientes representan la coocurrencia de primer orden del elemento de la primera columna de cada fila. En este caso, si *d* es la palabra más frecuente en la matriz para *a*, y además resulta ser una categoría perteneciente a la taxonomía *T*, entonces este módulo promoverá la candidatura de *d* como hiperónimo de *a*.

2.2.5. Módulo de asociación paradigmática

El módulo de asociación paradigmática se basa en la noción de similitud distributiva. Esta noción fue descrita por primera vez por distribucionalistas como Harris (1954), que observaron que las palabras semánticamente similares comparten características comunes en el tipo de contexto en el que aparecen. En el caso de este módulo, por características comunes entendemos palabras coocurrentes.

Para obtener el cálculo de asociación paradigmática se crea un vector de coocurrencias para un sustantivo de entrada *a*, de la misma manera que en el apartado 2.2.3. La diferencia en este caso es que el módulo también crea vectores de coocurrencia para cada categoría semántica de la taxonomía, porque la clasificación de *a* se basará en el solapamiento entre $\vec{a}$ y cada uno de los vectores de las categorías. De este modo, el módulo crea un vector de coocurrencia para cada uno de los hiperónimos de la taxonomía, y cada uno de ellos es un vector-categoría. Se diferencia del vector de coocurrencia de una sola palabra en que no es simplemente el vector de

coocurrencia de la palabra que designa la categoría. En cambio, el vector-categoría se calcula como la suma de todos los componentes de los vectores de cada hipónimo de dicho hiperónimo.

Para ilustrar esta idea con un ejemplo, si *a* es la palabra *piano*, entonces $\vec{a}$ tendrá como componentes palabras coocurrentes como *sonata*, *orquesta*, *voz*, *violín*, *concierto*, *guitarra*, etc. El hiperónimo o tipo semántico al que pertenece *piano* es *instrumento musical*. Para este hiperónimo existe un vector-categoría, y este vector-categoría tendrá como componentes la suma de todos los componentes de los vectores de cada sustantivo perteneciente a dicha categoría, es decir, todos los nombres de instrumentos musicales ya capturados.

La lógica de este módulo, por tanto, es que se compara a $\vec{a}$ con los vectores de cada una de las categorías de la taxonomía. Por término medio, el vector de coocurrencias de un sustantivo tendrá un cierto grado de solapamiento con el vector de categorías padre. Tal como se explicó en el apartado 2.2.3, la matriz $FM_{m,h,f}$ representa los rasgos que están altamente asociados con *h*, un hiperónimo dado. En el caso de este módulo, con palabras que muestran una frecuencia de coocurrencia significativa con palabras que son hipónimos de *h*. De esta manera, dado un sustantivo de entrada *a*, este módulo decide si *h* es un buen candidato a hiperónimo para *a* ($a \in h$) calculando el solapamiento entre $\vec{a}$ y $\vec{h}$, el vector de $FM_{m,h}$, como se muestra en la ecuación 3.

$$\forall a \notin h \wedge \forall h \in T : paradigm(\vec{a}, \vec{h}) = \frac{|\vec{a} \cap \vec{h}|}{|\vec{a}| + 1} \quad (3)$$

2.2.6. Módulo de similitud morfológica entre cohipónimos

A diferencia de los anteriores, este módulo de similitud morfológica no utiliza corpus ni medidas de coocurrencia. En cambio, consiste en detectar la coincidencia de rasgos morfológicos entre cohipónimos. Los rasgos se definen simplemente como secuencias de 3 a 5 letras al final de cada palabra, lo que significa que no es necesario contar con analizadores

morfológicos. Estos rasgos se extraen de la taxonomía y se ponderan exactamente igual que en los demás módulos, es decir, con la ecuación 1 de el apartado 2.2.3. Ya tenemos, por tanto, una matriz $FM_{m,h,f}$ con las características mejor puntuadas.

Para continuar con el ejemplo ya mencionado, si los otros módulos van clasificando distintos términos como tipos de enfermedades, este módulo va aprendiendo entonces a relacionar segmentos finales de esos términos (como *-itis*, *-osis*, etc.) con sus hiperónimos correspondientes. De esta manera, si un término de entrada *a* no aparece en corpus con la suficiente frecuencia, sus rasgos morfológicos podrían servir para apoyar la clasificación. Si *a* es *poliomielitis* y un candidato a hiperónimo *h* es *enfermedad*, podremos decidir si $a \in h$ según la ecuación 4, donde el símbolo $\vec{h}$ denota un vector de enegramas de caracteres asociados con el hiperónimo *h*, y $M_{a,f}$ denota un rasgo *f* de *a* como, por ejemplo, el segmento *-itis*.

$$MS(a \rightarrow h) = \begin{cases} true & M_{(a,f)} \in \vec{h} \\ false & \text{otherwise} \end{cases} \quad (4)$$

Este módulo funciona como un motor de inferencia y no como una simple medida de similitud morfológica porque no comprueba la similitud entre hipónimo e hiperónimo. También es agnóstico en cuanto a lengua, ya que no depende de conocimiento externo como léxicos, reglas o información morfológica, y permite capturar patrones repetidos de diferentes tamaños sin necesidad de realizar un análisis morfológico que sería costoso y dependiente de lengua. Por supuesto, la selección de las secuencias se verá afectada por el tipo filogenético de la lengua que se procesa (afijos iniciales frente a afijos finales, lenguas semíticas, etc.). Sin embargo, el patrón seleccionado es común a un gran subconjunto amplio de lenguas y puede adaptarse con facilidad para acomodar otras estructuras morfológicas. Como estrategia individual, la de este módulo posiblemente no sea muy útil, pero sí lo es en combinación con los demás módulos.

2.2.7. Módulo de reglas morfosemánticas

Este módulo está basado en un conjunto de reglas morfológicas elaboradas de manera manual con el propósito de codificar la asociación entre ciertos rasgos morfológicos y ciertas clases semánticas. Naturalmente, el módulo está limitado a las categorías más generales, ya que sería excesivamente laborioso proceder de esta manera con todas las categorías existentes. En el caso del inglés, los siguientes son algunos ejemplos de estas reglas, muy similares a las del francés y el castellano:

```
`property' => `ment|ness|ity|sion|[ea]nce|ncy',
`science' => `ics|logy|omy|graphy|[es]try|mics',
`treatment' => `therapy|scopy',
`surgery' => `surgery|ctomy',
`instrument' => `graph|scope|phone|meter',
`disease' => `[ao]sis|itis|pathy|emia|oma',
...
```

Por medio de estas reglas es posible determinar que si una palabra de entrada termina con alguno de los patrones del lado derecho (por ejemplo, *psychotherapy*), el elemento del lado izquierdo será promovido como hiperónimo.

Por el momento solo hay entre 20 y 30 reglas de este tipo por lengua, pero también se hacen extensivas a los hipónimos de los elementos de la izquierda. Por ejemplo, en el caso del hiperónimo *enfermedad*, los hipónimos de dicho elemento también se aceptarán como candidatos a hiperónimo (por ejemplo, *trastorno, patología, aflicción, inflamación, infección,* etc.). Si bien es cierto que esta estrategia sólo afectará a un número muy limitado de términos, sí será eficaz al menos en ese subconjunto.

2.2.8. Módulo de patrones lexicosintácticos

Tal como se mencionó en la sección 1, el uso de patrones léxicos como medio para extraer relaciones de hiperonimia del corpus se remonta al

trabajo de Hearst (1992), pero tiene sus raíces en trabajos realizados anteriormente sobre diccionarios (Chodorow, Byrd y Heidorn, 1985). Un patrón léxico es una secuencia como *HIPO es un tipo de HIPER*, entre diversos otros ejemplos.

En su artículo original, Hearst (1992) sugería buscar este tipo de patrones para extraer relaciones de hiperonimia entre los sustantivos o sintagmas nominales que se encontraran a cada lado del patrón, y muchos investigadores utilizaron este enfoque. Sin embargo, los patrones léxicos aplicados de esta manera son propensos a errores. Es en este contexto en el que se ha trabajado para tratar de identificar qué patrones son más fiables (Potrich y Pianta, 2008; Seitner et al., 2016).

En el caso de los patrones lexicosintácticos, la diferencia fundamental entre lo que se hace en el proyecto Kind y lo que hacen otros autores es que aquí no se busca directamente el patrón en el corpus. Por el contrario, lo que se hace es buscar primero el término de entrada en el corpus y recién entonces buscar los patrones sobre los contextos de aparición de dicho término. Estos patrones darán lugar a una serie de candidatos a hiperónimo, y la clave del procedimiento es la observación de la frecuencia de aparición de estos candidatos. Habrá un candidato a hiperónimo *h* con la frecuencia más alta, pero si la frecuencia no es lo suficientemente alta ($f(h) < 3$), entonces el candidato es rechazado y el módulo no devuelve ningún resultado. Esto significa que los resultados se producirán sólo en algunas ocasiones, pero si efectivamente se producen, tendrán una alta probabilidad de ser correctos.

A continuación, se muestran algunos ejemplos de los patrones recopilados para el castellano, que son similares a los que existen para el inglés y el francés.

HIPO es un HIPER
HIPO es un tipo de HIPER
HIPO, un tipo de HIPER

HIPO y otros HIPER

...

Por una cuestión de economía de esfuerzo, muchos de estos patrones se unen utilizando la sintaxis de las expresiones regulares, de tal manera que una línea de código puede incluir varios patrones, como en el siguiente ejemplo, donde "?" significa que la letra anterior puede no estar:

es una? (tipo|forma|clase) de

2.2.9. Módulo de identificación del núcleo sintáctico

Si el término de entrada es una expresión poliléxica, este módulo intentará identificar el núcleo sintáctico de la expresión. El procedimiento para ello es rudimentario pero eficaz en la gran mayoría de los casos: el último componente de la secuencia es el núcleo en el caso del inglés o el primero en el caso del francés y el castellano. La única regla añadida es que, en el caso del inglés, se comienza el proceso eliminando todos los elementos que se encuentran después de la primera aparición de la preposición *of*. Por ejemplo, en el caso de un término como *acute infectious disease*, el núcleo será *disease*, pero en el caso de *disease of lung*, el núcleo que se devuelve será de nuevo *disease* y no *lung*.

La idea detrás de este proceso no es simplemente identificar el núcleo sintáctico del término de entrada con su hiperónimo, ya que esto no sería de mayor utilidad. La razón de ser del módulo es, en cambio, que si el término de entrada no se encuentra con suficiente frecuencia en el corpus, entonces la identificación del núcleo sintáctico puede servir para iniciar una nueva instancia del proceso de descubrimiento de hiperónimos. Esto es, para obtener el hiperónimo del núcleo, ya que es probable que el hiperónimo del núcleo sea válido también para el término completo.

2.2.10. Módulo de poda

La idea de incorporar este módulo era que sirviera como medio de control de la consistencia interna de la taxonomía, es decir, para identificar y destruir enlaces incorrectos. La entrada de este módulo es una lista de

pares hipónimo-hiperónimo y la salida es un booleano para cada par. Las aserciones con probabilidad de ser verdaderas se representan como una matriz $T_{a,h}$, donde a es un elemento miembro (un hipónimo) y h es la categoría semántica (un hiperónimo).

La poda se realiza en función de una medida de asociación sintagmática. Así, la salida de este primer módulo para $T_{a,h}$ se obtiene a partir del estimador $M_{a,h}$, que mide la frecuencia de coocurrencia en el corpus de la supuesta pareja hipónimo- hiperónimo a - h. Puede decirse que $M_{a,h}$ es una medida de la importancia relativa de h respecto a a, y es por tanto apropiada para relaciones asimétricas como la hiperonimia (ecuación 5).

$$M_{a,h} = \frac{\sum_{i=1}^{|C|} \begin{cases} 1 & h \in C_i \\ 0 & \text{otherwise} \end{cases}}{|C| + 1} \quad (5)$$

$M_{a,h}$ se calcula mediante el análisis de una muestra aleatoria de contextos de aparición de cada hipónimo a en el corpus. Puede utilizarse para calcular una clasificación de las afirmaciones más fiables o simplemente para devolver un booleano si $M_{a,h} > p$, con p como parámetro arbitrario. Además, un valor verdadero para un par $M_{a,h}$ solo se emite si pasa la prueba de asimetría (ecuación 6).

$$S_{a,h} = \begin{cases} \text{true} & M_{a,h} > M_{h,a} \\ \text{false} & \text{otherwise} \end{cases} \quad (6)$$

2.2.11. Módulo de procesamiento central

Para producir las cadenas taxonómicas, un procesador central articula la salida de los diferentes módulos utilizando una estrategia descendente y otra ascendente. Estas estrategias no son más que una disposición particular de los módulos ya presentados.

El procedimiento descendente –*td(),* por *top-down*– del procesador central consiste en formular las preguntas más generales. Dado un sustantivo de entrada *a*, se pregunta primero si *a* es una entidad, una propiedad o un evento. Una vez que se encuentra una respuesta a esta primera pregunta, se pasa al siguiente nivel. Si se trata de una entidad, por ejemplo, la siguiente pregunta sería si es concreta o abstracta. Si resultara ser concreta, la siguiente pregunta es si es animada o inanimada. Es en este sentido que se trata de una estrategia descendente, porque va descendiendo a través de una estructura arbórea y debe decidir en cada etapa por qué rama continuar. Este proceso continúa hasta llegar a un nivel medio de abstracción como, por ejemplo, si se determina que el término de entrada es un tipo de arma o un tipo de pez. Estos últimos son sustantivos lo suficientemente generales como para ser incluidos como nodo terminal de la estructura de base (apartado 2.2.1).

Es en este punto donde comienza el procedimiento inverso, es decir, la estrategia ascendente –bu(), por *bottom-up*–, que tiene como objetivo encontrar el hiperónimo más inmediato o específico del dominio. El procedimiento ascendente será lo suficientemente robusto como para producir resultados incluso si el procedimiento descendente falla, porque a partir de un hiperónimo inmediato podemos seguir construyendo una cadena taxonómica sometiendo recursivamente el hiperónimo obtenido a una nueva instancia del procedimiento descendente. Por ejemplo, si el término de entrada *a* es *fluoxetina*, el resultado de procedimiento ascendente debería ser *inhibidor selectivo de la recaptación de serotonina*, *ISRS* o *antidepresivo*. Con estos candidatos es posible recomenzar el proceso para vincular dicho resultado con un término más general ya incluido en la taxonomía. De esta forma, el algoritmo consigue conectar los segmentos en una única cadena taxonómica que va desde el nodo terminal hasta el vértice de la estructura (*top node*).

A continuación, se ofrece una explicación más detallada de la manera en que se articulan ambas estrategias. Dado un término de entrada *a* (y

suponiendo[1] que $a \notin T$): si *a* es poliléxico, primero se obtiene su núcleo sintáctico *nu(a),* y si $nu(a) \in T$, entonces el proceso concluye. En caso contrario, se llama a la función *td(a)*. De ello resulta un primer candidato a hiperónimo h_1. A continuación, procede con la función *bu(a),* que resulta en uno o más candidatos bh_i ($i < 4$). Si alguno de estos, según la misma función $bu(bh_i)$, es hipónimo de h_1, se selecciona y el proceso termina. Caso contrario, si pertenecen a la taxonomía ($bh_i \in T$ v $bu(bh_i) \in T$), entonces bh_i pasa a formar parte del conjunto de candidatos a hiperónimo *h*. Si no es así, se aplica la función descendente con el núcleo sintáctico (*td(nu(a))*), y el resultado se asimila al conjunto *h.*

Si existe más de un candidato ($|h| > 1$), se toma una decisión final mediante la función *best(h)*, basada en la cantidad de módulos que hayan votado a cada uno. El sistema computa esos valores en una matriz *R* para gestionar la contribución de cada módulo y obtener así la puntuación final de los pares *a* y *h* (*Z(a,h)*) como la suma de cada contribución (ecuación 7).

$$Z(x, y) = \sum_{i=1}^{|R|} R_i \quad (7)$$

Por último, y antes de decidir la promoción de la candidatura de un determinado hiperónimo, el sistema llama a la función *poda(a,best(h))*, descrita en el apartado 2.2.10. Esta función toma como argumentos el término de entrada y el mejor candidato a hiperónimo y decide si destruye o no esa candidatura. Si el mejor candidato es destruido, se pasa al segundo mejor. Si este no existe o también es destruido, se pasa al tercero, y si tampoco hay éxito con este, el sistema declara que no hay resultados.

3. Resultados

La evaluación de los resultados del proyecto se lleva a cabo por medio pruebas de poblamiento de la taxonomía a partir de listados de términos

[1] Reanalizar términos ya incluidos permitiría descubrir nuevos significados de estos términos, y esta es una de las líneas de trabajo futuro.

de entrada, mono y poliléxicos, especializados y generales. Se proporciona al sistema este material y luego se registra si ha sido capaz de clasificar las unidades correctamente.

La manera en la que se suele proceder con la evaluación es mediante la comparación automática con listados de vocabulario o *gold-standards*. Estos son listados de pares hipónimo-hiperónimo, y lo que se hace es comparar el hiperónimo que viene en el listado con los que propone el sistema. Para la evaluación fueron utilizados listados de vocabulario general y especializado, en inglés, francés y castellano. Los de vocabulario general se elaboraron de manera manual en el marco del proyecto Kind. En cuanto a los vocabularios especializados, en el caso del inglés y francés fueron tomados de otros autores (Bordea, Lefever y Buitelaar, 2016; Camacho-Collados et al., 2018), y consisten en términos que designan ciencias o áreas de especialidad. En castellano, a falta de algo similar, se constituyó uno con nombres de fármacos tomados del Vademecum[1].

Si bien este protocolo de evaluación evita el voluminoso trabajo que representaría una evaluación manual de los resultados y el sesgo inherente a la subjetividad del evaluador, también presenta el problema de que muchas veces el hiperónimo que devuelve el algoritmo evaluado es correcto; pero a la vez distinto del que está en el vocabulario debido, entre otras cosas, al problema de la polisemia. Este sería el caso, por ejemplo, de una palabra como *tomahawk*, que el algoritmo clasifica correctamente como un tipo de *herramienta*, pero que en las estadísticas cuenta como incorrecto debido a que no coincide con el vocabulario, en el que aparece como *arma* debido al nombre del misil. Algo similar ocurre en el caso de *fax*, que por polisemia regular sería también un tipo de *documento*, pero en el vocabulario figura como un tipo de *máquina*.

[1] https://www.vademecum.es/

Tabla 1. Resultados de la evaluación con vocabularios

Lengua	Dominio	Términos	Pre	Rec	F1
Es	Especializado	1767	.79	.63	.70
Es	General	358	.82	.82	.82
Fr	Especializado	449	.70	.61	.65
Fr	General	291	.93	.89	.90
En	Especializado	451	.73	.70	.71
En	General	273	.77	.76	.76

La **tabla 1** muestra la evaluación de los resultados con distintos listados de términos en las tres lenguas. Tal como se puede apreciar, los resultados no son homogéneos entre sí. Uno de los motivos que pueden explicar esta variabilidad es que algunos de los vocabularios contienen altas tasas de error, a pesar de que se utilizan en competencias internacionales. Más allá de esta circunstancia, los resultados de Kind, según esta evaluación, son comparables con los informados por otros investigadores que proponen sistemas considerablemente más complejos y costosos (Panchenko et al. 2017; Sarkar, McCrae y Buitelaar 2018, Qiu et al. 2018).

Otro de los aspectos que interesaba evaluar es el desempeño individual de los distintos módulos de Kind, pero como no es posible que un módulo funcione con independencia del resto del conjunto, lo que se hizo fue repetir varias veces los experimentos desconectando cada vez un módulo distinto. El resultado fue que no se advirtieron diferencias significativas en el desempeño general del sistema, lo cual indica que la calidad del resultado está en el conjunto y no el desempeño individual de alguno de los módulos.

Conclusiones

Este artículo ha tenido el propósito de ofrecer un resumen del estado actual del proyecto de inducción de taxonomías Kind, trabajo que ha ido evolucionando lentamente en el transcurso de la última década. Si bien todavía hay tarea pendiente, sí es posible señalar que el proyecto ha alcanzado un punto de maduración y estabilidad en su diseño metodológico, y las vías de investigación futura apuntan a mejoras puntuales.

Desde un punto de vista teórico, el proyecto ofrece aspectos interesantes, particularmente en el caso de algunos de los módulos, como el que explota la coocurrencia asimétrica, una propiedad natural del lenguaje que no había sido suficientemente descrita hasta ahora. Además, el proyecto puede servir también para otros estudios de corte teórico, como ha sido ya el caso de análisis de los patrones de uso de verbos en función de las categorías semánticas de sus argumentos (Renau et al. 2019).

Tal como se presenta en la actualidad, el sistema ofrece además algunas posibilidades de aplicación práctica. Algunos ejemplos podrían ser analizar un texto para determinar su tópico, lo cual puede a su vez ser útil para la clasificación de documentos. Otra aplicación concreta podría ser la de hacer búsquedas en un corpus por hiperónimos en lugar de por términos, como propuso O'Donnell (2008).

En cuanto a líneas de trabajo futuro, la primera es continuar con el poblamiento de la taxonomía y la alineación multilingüe de las categorías y unidades. Más adelante, sería de interés también continuar incorporando otras lenguas. Otra línea concreta de trabajo futuro podría ser la de aplicar un preproceso a los términos de entrada. Esto sería posible en caso de que la entrada del proceso esté constituida por un conjunto de términos que incluyen unidades poliléxicas y que además pertenecen a un mismo dominio temático. En un caso así, el preproceso podría consistir en la detección de repeticiones, dentro de este conjunto, de unidades que sirven como núcleo sintáctico, lo cual sería una forma de obtener categorías

semánticas relevantes del dominio específico. Sería el caso, por ejemplo, de *trastorno* en un dominio médico.

También está pendiente hacer frente a algunas de las limitaciones que presenta actualmente el sistema. Un problema concreto que no está bien resuelto todavía es el tratamiento de los nombres propios. Existe todavía confusión con respecto a los nombres propios incluso en las taxonomías desarrolladas de manera manual, tal como señalan Miller y Hristea (2006) en el caso de WordNet. Un nombre propio no debe formar parte de una taxonomía ya que no puede establecer una relación de hiperonimia. Contrariamente a lo que indica WordNet, Sócrates no es un tipo de filósofo, sino una instancia del tipo filósofo. Por ello, todo sistema de inducción de taxonomías debería ser capaz de detectar si un término de entrada es un nombre propio y rechazarlo.

Otro problema no resuelto es el tratamiento de la polisemia, una dificultad típica de los sistemas de inducción de taxonomías, advertido ya por Amsler (1981). Si bien la identificación de las relaciones entre hipónimos e hiperónimos es relativamente estable en el marco de este proyecto, el conflicto se produce en las cadenas mayores cuando uno de los eslabones está ocupado por un término polisémico. De hecho, esta es actualmente una de las principales fuentes de error del algoritmo. Se está explorando una vía para solucionar este problema (Nazar, Obreque y Renau 2020), pero queda trabajo por hacer.

Esta última es una línea de trabajo interesante porque conecta con el problema de la detección de neología semántica. Se trataría en este caso de detectar que una palabra ha desarrollado un nuevo significado o ha pasado por un proceso de resemantización, ya sea por especialización, generalización o por otros procesos. La idea aquí sería acusar el cambio de significado al advertir que el término adquiere un nuevo hiperónimo. Es una línea de trabajo interesante pero también más compleja, porque añade la dimensión temporal a un problema que hasta ahora solo había sido tratado mediante el corte sincrónico.

Referencias bibliográficas

Amsler, R. (1981). A taxonomy for English nouns and verbs. En *Proceedings of the 19th annual meeting on Association for Computational Linguistics*, pp. 133–138.

Bordea, G.; Lefever, E. y Buitelaar, P. (2016) SemEval-2016 Task 13: Taxonomy extraction evaluation (texeval-2). En *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, Association for Computational Linguistics, pp. 1081–1091.

Borgo, S., y Masolo, C. (2009). Foundational choices in DOLCE. En: S. Staab, y R. Studer (eds.), *Handbook on ontologies*. Berlín: Springer, pp. 361–382.

Bullinaria, J. (2008). Semantic Categorization Using Simple Word Co-occurrence statistics. En *ESSLLI Workshop on Distributional Lexical Semantics*, Hamburg, Alemania.

Camacho-Collados, J.; Delli Bovi, C.; Espinosa Anke. L.; Oramas, S.; Pasini, T.; Santus, E.; Shwartz, V.; Navigli, R.; Saggion, H. (2018). SemEval-2018 Task 9: Hypernym Discovery. En *Proceedings of The 12th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT*, Nueva Orleans, Louisiana, 2018, pp. 712–724.

Chodorow, M.; Byrd, R; y Heidorn G. (1985). Extracting semantic hierarchies from a large on-line dictionary. En *Proceedings of the 23rd annual meeting on ACL,* Chicago, Illinois, pp. 299–304.

Cornet, R. y de Keizer, N. (2008). Forty years of SNOMED: a literature review. *BMC Med Inform Decis Mak* 8, S2.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. Cambridge: MIT Press.

Grefenstette, G. (1994). Explorations in Automatic Thesaurus Discovery. Norwell, MA: Kluwer Academic Publishers.

Hanks, P. y Pustejovsky, J. (2005). A Pattern Dictionary for Natural Language Processing. *Revue Francaise de Langue Appliquée* 10.

Harris, Z. (1954). Distributional Structure. *Word*. 10, pp. 146–162.

Hearst, M. (1992). Automatic Acquisition of Hyponyms from Large Text Corpora. En *Proceedings of the 14th Conference on Computational Linguistics* – Vol. 2, COLING '92, pp. 539–545.

Jakubíček, M.; Kilgarriff, A.; Kovář, V.; Rychlý, P. y Suchomel, V. (2013). The TenTen Corpus Family. En *7th International Corpus Linguistics Conference CL 2013*, pp. 125–127.

Lenat, D. (1995). CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38 (11), 33-38.

Lin, D. (1998). Automatic Retrieval and Clustering of Similar Words. En: Proceedings of the 17th International Conference on Computational Linguistics - Volume 2, COLING '98, pp. 768–774.

Miller, G. y Hristea, F. (2006). Squibs and Discussions: WordNet Nouns: Classes and Instances, American Journal of Computational Linguistics 32: 1–3.

Navigli, R., & Ponzetto, S. (2010). BabelNet: Building a very large multilingual semantic network. En: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 216–225.

Nazar, R. (2010). *A Quantitative Approach to Concept Analysis*. Tesis doctoral, Universitat Pompeu Fabra.

Nazar, R.; Balvet, A.; Ferraro, G.; Marín, R.; Renau, I. (2020). Pruning and repopulating a lexical taxonomy: experiments in Spanish, English and French. *Journal of Intelligent Systems*, 30(1): 376–394.

Nazar, R.; Janssen, M. (2010). Combining Resources: Taxonomy Extraction from Multiple Dictionaries. En *Proceedings of The 8th edition of the Language Resources and Evaluation Conference (LREC 2010)*, pp. 1055–1061.

Nazar, R.; Obreque, J.; Renau, I. (2020). Tarántula –> araña –> animal : asignación de hiperónimos de segundo nivel basada en métodos de similitud distribucional. *Procesamiento del Lenguaje Natural*, (64): 29–36.

Nazar, R.; Renau, I. (2015). Agrupación semántica de sustantivos basada en similitud distribucional: implicaciones lexicográficas. En María Pilar Garcés Gómez (ed.): *Lingüística y diccionarios. Anexos Revista de Lexicografía*, vol. 2, pp. 272–295.

Nazar, R.; Renau, I. (2016). A taxonomy of Spanish nouns, a statistical algorithm to generate it and its implementation in open source code. En *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, pp. 1485–1492.

Nazar, R.; Vivaldi, J; Wanner, L. (2012). Co-occurrence Graphs Applied to Taxonomy Extraction in Scientific and Technical Corpora. *Procesamiento del Lenguaje Natural*, (49): 67–74.

O'Donnell, M. (2008). The UAM CorpusTool: Software for corpus annotation and exploration. En Bretones Callejas, Carmen M. et al. (eds), *La lingüística aplicada hoy: comprendiendo el lenguaje y la mente*. Universidad de Almería, pp. 1433–1447.

Panchenko, A.; Faralli, S.; Ruppert, E.; Remus, S.; Naets, H.; Fairon, C.;Ponzetto, S. y Biemann, C. (2016). TAXI at SemEval-2016 Task 13: a Taxonomy Induction Method based on Lexico-Syntactic Patterns, Substrings and Focused Crawling. En *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 1320–1327.

Pereira, F.; Tishby, N. y Lee, L. (1993). Distributional clustering of English words. En *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pp. 183–190.

Potrich, A. y Pianta, E. (2008). L-ISA: Learning Domain Specific Isa-Relations from the Web. En *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, pp. 2368–2375.

Qiu, W.; Chen, M.; Li, L. y Si, L. (2018). NLP_HZ at SemEval-2018 Task 9: a Nearest Neighbor Approach. En *Proceedings of The 12th International Workshop on Semantic Evaluation*, pp. 909–913.

Renau, I.; Nazar, R.; Castro, A.; López, B.; Obreque, J. (2019). Verbo y contexto de uso: un análisis basado en corpus con métodos cualitativos y cuantitativos. *Revista Signos*, 52(101): 878–901.

Renau, I.; Nazar, R. (2012). Hypernymy relations from definiens-definiendum co-occurrence in multiple dictionary definitions. *Procesamiento del Lenguaje Natural*, (49): 83–90.

Sager, J. (1990). *A Practical Course in Terminology Processing*, Amsterdam/Philadelphia: John Benjamins.

Sarkar, R.; McCrae, J. y Buitelaar, P. (2018). A supervised approach to taxonomy extraction using word embeddings. En *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018),* pp. 2059–2064.

Seitner, J; Bizer, C.; Eckert, K.; Faralli, S.; Meusel, R.; Paulheim, H. y Ponzetto, S. (2016). A Large DataBase of Hypernymy Relations Extracted from the Web. En Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC'16), pp. 360–367.

Shwartz, V.; Santus, E. y Schlechtweg, D. (2017). Hypernyms under Siege: Linguistically-motivated Artillery for Hypernymy Detection. En *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, vol. 1, pp. 65–75.

Snow, R.; Jurafsky, D. y Ng, A. (2006). Semantic Taxonomy Induction from Heterogenous Evidence. En Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pp. 801–808.

Weeds, J. y Weir, D. (2003). A General Framework for Distributional Similarity. En *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pp. 81–88.

Nota biográfica

Rogelio Nazar

Rogelio Nazar es Licenciado en Comunicación Social por la Universidad Nacional de Cuyo (Argentina) y Doctor en Ciencias del Lenguaje y Lingüística Aplicada por la Universidad Pompeu Fabra (España). Actualmente es profesor del Instituto de Literatura y Ciencias del Lenguaje de la Pontificia Universidad Católica de Valparaíso (Chile). Su área de investigación es la lingüística computacional y trabaja en lexicología, terminología y análisis computacional del discurso.