

FunGramKB y el conocimiento cultural

Carlos Perrián Pascual, Universidad Politécnica de Valencia (España)

María de los Llanos Carrión Varela, UNED (España)

Índice

- 1 Introducción
- 2 Bases de conocimiento cultural
 - 2.1 YAGO y YAGO2
 - 2.2 DBpedia
- 3 Metodología para la importación de conocimiento cultural a FunGramKB
 - 3.1 Fase de construcción
 - 3.1.1 Relevancia y pertinencia de las propiedades
 - 3.1.2 Identificación del tipo de bioestructura
 - 3.1.3 Identificación de la unidad de medida
 - 3.1.4 Construcción de la plantilla
 - 3.2 Fase de proyección
- 4 Motivación del proyecto
- 5 Conclusiones
- Agradecimientos
- Referencias
- Apéndice 1. Las plantillas COREL para la clase Place de DBpedia.

Resumen

Una base de conocimiento provista de información sobre entidades nombradas, i.e. entidades con nombre propio, resulta de suma importancia para algunas tareas del procesamiento del lenguaje natural, p.ej. la resolución de la anáfora. Por esta razón, en los últimos años han proliferado los proyectos orientados a la extracción automática de ese tipo de entidades, además de recuperar información semántica sobre ellas. En este sentido, dentro del marco de FunGramKB, nosotros pretendemos importar por medio de plantillas de proyección los datos almacenados en repositorios sobre el conocimiento cultural, el cual está configurado principalmente por entidades nombradas. Así, el objetivo de este artículo es describir el proceso semiautomático de migración del conocimiento cultural de DBpedia a los constructos cognitivos del Onomasticón de FunGramKB.

Palabras clave: FunGramKB, Onomasticón, conocimiento cultural, entidades nombradas, procesamiento del lenguaje natural

1 Introducción

El término “entidad nombrada”, acuñado en *The Sixth Conference on Message Understanding* (1995), suele referirse a “una palabra o secuencias de palabras que se identifican con un nombre de persona, organización o lugar”, aunque a veces también se incluyen términos tan variados como las fechas o las fórmulas químicas. Estas entidades nombradas desempeñan un papel crucial en el procesamiento del lenguaje natural (PLN). Por ejemplo, con el fin de mantener la cohesión en los textos traducidos automáticamente, es preciso resolver los problemas ocasionados por la correferencia. Además, en la indización automática de materiales textuales, los términos más informativos suelen ser las entidades nombradas; por ejemplo, resulta muy fácil averiguar cuál es el tema de un documento si aparecen términos como Pearl Harbor y

Japón. Y evidentemente ni mencionar cabe la importancia del papel de las entidades nombradas en las tareas de comprensión del lenguaje. Desde el punto de vista informativo, las entidades nombradas son expresiones lingüísticas que actúan como anclajes referenciales y que denotan objetos ontológicos en los documentos (Nédellec y Nazarenko 2009). Por ello, aunque tuviéramos un recurso con todos los nombres propios, la mayoría de los problemas radicarían en la falta de información semántica, o más bien de conocimiento cultural, el cual se precisa para realizar las tareas del PLN descritas anteriormente.

En este sentido, Wikipedia¹ es sin duda una de las más profusas fuentes de información que en la actualidad almacena conocimiento cultural sobre millones de entidades. Wikipedia, la cual pertenece a la Fundación Wikimedia, se concibe como una enciclopedia multilingüe que posee artículos en más de 275 lenguas, libremente accesibles en la Web y editados de manera cooperativa por voluntarios de todo el mundo, bajo la supervisión de un grupo de administradores que velan por el mantenimiento de los artículos y el cumplimiento de las normas. A pesar de su organización jerárquica, el conocimiento almacenado en Wikipedia carece de estructuración conceptual, por lo cual parece razonable pensar que reutilizar dicho conocimiento con el fin de dotarlo de este tipo de contenido y así poder integrarlo en una base de conocimiento es una tarea que merece la pena acometer. Partiendo de esta línea de actuación, surgieron proyectos como YAGO² y DBpedia,³ los cuales reconocen el gran valor informativo que Wikipedia puede ofrecer siempre y cuando pueda ser organizado bajo una ontología.

Nuestro objetivo no es realizar un proyecto de “reconocimiento de entidades nombradas”, tal y como hicieron YAGO y DBpedia, donde a cada entidad identificada se le asocia una forma canónica además de asignarle una categoría semántica. Más bien, proponemos importar a FunGramKB⁴ —y más concretamente al Onomasticón— el conocimiento cultural ya almacenado en otras bases de datos existentes. El Onomasticón de FunGramKB almacena el conocimiento sobre instancias de entidades y eventos, las cuales pueden ser descritas sincrónica o diacrónicamente. Por tanto, podemos presentar el conocimiento cultural atemporalmente (i.e. retrato), p.ej. la descripción física de un personaje público, o inserto en un paradigma temporal (i.e. historia), p.ej. la secuencia de acontecimientos que tuvieron lugar en un evento histórico. Al igual que en el resto de módulos cognitivos de FunGramKB, se utilizará el lenguaje de representación conceptual COREL (cf. Perrián Pascual y Mairal Usón, 2010) para la descripción formal de las entidades nombradas, lo cual no sólo nos permitirá integrar ese conocimiento cultural con el conocimiento del sentido común de la Ontología Nuclear o del Cognición, sino también podremos reutilizar el mismo motor de razonamiento.

2 Bases de conocimiento cultural

2.1 YAGO y YAGO2

El proyecto YAGO⁵ (Suchanek *et al.*, 2007, 2008; De Melo *et al.*, 2008; Kasneci *et al.*, 2008), cuya ontología contiene más de un millón de entidades, pretende unificar la

¹ www.wikipedia.org

² www.mpi-inf.mpg.de/~suchanek/downloads/yago/

³ dbpedia.org

⁴ Desde 2004, se ha ido acumulando una extensa literatura sobre FunGramKB (cf. Perrián-Pascual y Arcas-Túnez 2004, 2005, 2007a, 2010a, 2010b; Perrián-Pascual y Mairal-Usón 2010; Mairal-Usón y Perrián-Pascual 2009; Jiménez Briones y Luzondo Oyón 2011; Jiménez Briones y Pérez Cabello de Alba 2011), a la cual remitimos al lector con el fin de ampliar la información sobre esta base de conocimiento.

⁵ Este acrónimo representa las palabras *Yet Another Great Ontology*.

información extraída de manera automática de Wikipedia y complementarla con la base de datos léxica WordNet⁶, de manera que esta última se enriquezca mediante la inclusión de información relativa a entidades nombradas. Como se muestra en la Tabla 1, YAGO se limita a describir los hechos a través de catorce relaciones semánticas.

Relation	Domain	Range	# Facts
SUBCLASSOF	class	class	143,210
TYPE	entity	class	1,901,130
CONTEXT	entity	entity	~40,000,000
DESCRIBES	word	entity	986,628
BORNINYEAR	person	year	188,128
DIEDINYEAR	person	year	92,607
ESTABLISHEDIN	entity	year	13,619
LOCATEDIN	object	region	59,716
WRITTENINYEAR	book	year	9,670
POLITICIANOF	organization	person	3,599
HASWONPRIZE	person	prize	1,016
MEANS	word	entity	1,598,684
FAMILYNAMEOF	word	person	223,194
GIVENNAMEOF	word	person	217,132

Tabla 1. Las relaciones semánticas en YAGO (Suchanek *et al.*, 2007).

Mostramos a continuación algunos ejemplos que ilustran la manera en la que YAGO codifica la información (Suchanek *et. al.*, 2007: 698-699):

- (1) a) "*Einstein*" MEANS AlbertEinstein
- b) AlbertEinstein TYPE physicist
- c) physicist SUBCLASSOF scientist
- d) subClassOf TYPE transitiveRelation

En (1a), tiene lugar una proyección de la realización lingüística, i.e. "*Einstein*",⁷ hacia el objeto ontológico almacenado en YAGO, i.e. *AlbertEinstein*,⁸ por medio de la relación *MEANS*. En los ejemplos (1b) y (1c), en cambio, se reflejan dos de las múltiples propiedades de la entidad *AlbertEinstein* que YAGO almacena: en primer lugar, la entidad *AlbertEinstein* es una instanciación de la clase *physicist*, una relación que se expresa mediante la relación *TYPE*, mientras que la clase *physicist* está subordinada a su vez a la clase *scientist* a través de la relación *SUBCLASSOF*. Finalmente, en (1d) se observa que las propias relaciones también se consideran entidades; por ejemplo, la relación *subClassOf* arriba mostrada, en minúscula en esta ocasión por ser tratada como una entidad, es un subtipo de la clase *transitiveRelation*. Formalmente, todos estos tripletes de información se almacenan en RDF:

⁶ wordnet.princeton.edu

⁷ "Einstein" se encuentra entrecomillado en YAGO para expresar que se trata de una palabra.

⁸ Suchanek *et. al.* (2007) reconocen la posibilidad de que una misma palabra, en este caso "Einstein", se refiera a más de una entidad, como por ejemplo el científico Albert Einstein o el músico Alfred Einstein. Estos autores también proporcionan el ejemplo de la palabra "*capital*", entendida principalmente como "*capital city*" en Wikipedia y confundida frecuentemente con "*financial assets*", siendo este último sentido el más frecuente en WordNet. Todo esto implica que en algún punto del proceso sea preciso efectuar manualmente una desambiguación léxica, ya que los diferentes sentidos de un término deberían estar vinculados a unidades ontológicas diferentes. A este respecto, el tratamiento de la ambigüedad semántica es más eficaz en YAGO2 mediante la inclusión de las *reglas factuales* (Hoffart *et al.*, 2010: 9).

```
<rdf:Description
  rdf:about="http://mpii.mpg.de/yago#Albert_Einstein">
  <yago:bornInYear rdf:ID="f1">1879</yago:bornInYear>
</rdf:Description>
```

Si comparamos YAGO con el Onomasticón de FunGramKB, encontramos una diferencia notoria. YAGO es una base de datos relacional, no de semántica profunda, lo cual sitúa a FunGramKB desde el primer momento en una posición de ventaja sobre ella. Según Perrián Pascual (en prensa), la descripción de la semántica léxica puede ser realizada desde tres enfoques diferentes de acuerdo con la granularidad del contenido cognitivo:⁹

- (i) enfoque superficial, por medio de asociaciones semánticas con otras unidades léxicas,
- (ii) enfoque intermedio, por medio de relaciones conceptuales binarias formando una simple matriz de atributo-valor, o
- (iii) enfoque profundo, por medio de rasgos semánticos que proporcionen una verdadera definición, p.ej. a través de una representación semántica proposicional.

A pesar de que una base de conocimiento relacional (i.e. enfoques superficial e intermedio) puede presentar ventajas a la hora de representar el conocimiento, dada la relativa sencillez de su representación, este tipo de modelo no es suficiente para los sistemas automáticos de comprensión del lenguaje, ya que su poder expresivo se ve drásticamente mermado (cf. Perrián-Pascual y Arcas-Túnez, 2007b).

Futuros desarrollos de YAGO han dado como fruto una versión mejorada y enriquecida denominada YAGO2 (Hoffart *et al.*, 2010). Se trata de una extensión de la versión anterior, donde la principal mejora radica en que tanto las entidades como los eventos contenidos en esta base de conocimiento han sido ubicados en un plano espacio-temporal. Para ello, YAGO2 toma ahora datos de manera automática de tres fuentes principales: Wikipedia, WordNet y GeoNames.¹⁰ De esta manera, la dimensión espacial de las entidades adopta la forma de coordenadas geográficas, las cuales se obtienen de Wikipedia y de GeoNames. No obstante, FunGramKB no considera que las coordenadas de ubicación geográfica de la instancia de una entidad resulten de gran utilidad para el PLN, razón por la cual se desestima este tipo de conocimiento.

Por otra parte, a las entidades en YAGO2 se les asigna un rango temporal para denotar su existencia en el tiempo. Por ejemplo, Elvis Presley está asociado con 08-01-1935 y 16-08-1977 como sus fechas de nacimiento y muerte respectivamente; en cambio, Barack Obama sólo está vinculado a su fecha de nacimiento, 04-08-1961. Además, los hechos pueden o bien tener un punto concreto de ocurrencia si se consideran instantáneos, o bien tener un rango temporal cuando posean una fecha de inicio y otra de fin conocidas. Por ejemplo, el hecho de que Bob Dylan publicara el álbum *Blonde on Blonde* está asociado al punto temporal 16-05-1966, pero que estuviera casado con Sara Lownds está asociado con el rango temporal de 1965-##-

⁹ En este sentido, no debemos confundir entre la granularidad de la descripción semántica (o contenido cognitivo) y la granularidad de los papeles semánticos. Por ejemplo, FrameNet (Baker *et al.*, 1998; Ruppenhofer *et al.*, 2006) utiliza papeles semánticos de granularidad muy fina, pero sus marcos semánticos son bastante superficiales desde el punto de vista cognitivo. En cambio, los papeles temáticos de COREL en FunGramKB son bastante generales, pero la granularidad descriptiva de los constructos cognitivos puede ser muy fina. Por tanto, la granularidad en la descripción del contenido cognitivo no depende tanto del inventario de papeles semánticos, sino más bien del poder expresivo del lenguaje de representación conceptual.

¹⁰ www.geonames.org

a 1977-##-##. Este fenómeno puede entrañar cierto paralelismo con las macroestructuras cognitivas de FunGramKB, si bien con matices. A la hora de conferir una dimensión temporal, en FunGramKB encontramos dos tipos de constructos cognitivos (i.e. retratos e historias) que pueden ser aplicados a cada uno de los conceptos nombrados. Los retratos son llamados así por el hecho de asemejarse a una fotografía instantánea que se toma en el momento actual. Las historias, por el contrario, contienen entidades y eventos que han de ser integrados dentro de un esquema temporal determinado. En FunGramKB, cada propiedad de una entidad o un hecho de YAGO2 se trataría como una predicación de un retrato o de una historia. Por ejemplo, la entidad BobDylan está descrita por las propiedades birthDate o birthName, entre otras. Ahora bien, la propiedad birthDate formaría parte de una historia, ya que se enmarca dentro de un rango temporal determinado, mientras que birthName formaría parte de un retrato. Por tanto, nuestra entidad %BOB_DYLAN_00 estaría provista de un retrato y una historia independientes aunque interrelacionados.

Con respecto al contenido informativo en YAGO2, éste mantiene un enfoque conservador, puesto que, en caso de duda, aboga por dejar sin asignar el valor espacial o temporal a algunas entidades, en lugar de proporcionar una información errónea. En este sentido, YAGO2 comparte con FunGramKB el compromiso de no asignar ningún tipo de conocimiento (p.ej. semántico, temporal, etc.) del que no se tenga la certeza de su validez, incluso a riesgo de provocar un vacío conceptual. De esta forma, se puede afirmar que quizás no esté todo, pero todo lo que está es cierto.

2.2 DBpedia

Otro proyecto relevante para el desarrollo del Onomástico de FunGramKB es DBpedia (Auer *et al.*, 2007; Bizer *et al.*, 2009). De hecho, si bien DBpedia tiene en común con YAGO el hecho de que obtiene sus datos de manera automática a partir de Wikipedia,¹¹ la estructuración del conocimiento resultante es diferente. DBpedia almacena información acerca de 3,5 millones de entidades —incluyendo personas, lugares, obras musicales, películas y empresas, entre otros muchos tipos— tomadas principalmente de los artículos de Wikipedia, almacenando toda esta información de manera jerárquica. De hecho, esta jerarquía de clases se organiza en la denominada *Ontología de DBpedia*, cuyas clases más comunes se presentan en la Tabla 2.

¹¹ Ambas bases de datos están estrechamente relacionadas, ya que YAGO sirve incluso como fuente de contenidos a DBpedia, formando así parte de ella.

Ontology Class	Instances	Example Properties
Person	198,056	name, birthdate, birthplace, employer, spouse
Artist	54,262	activeyears, awards, occupation, genre
Actor	26,009	academyaward, goldenglobeaward, activeyears
MusicalArtist	19,535	genre, instrument, label, voiceType
Athlete	74,832	currentTeam, currentPosition, currentNumber
Politician	12,874	predecessor, successor, party
Place	247,507	lat, long
Building	23,304	architect, location, openingdate, style
Airport	7,971	location, owner, IATA, lat, long
Bridge	1,420	crosses, mainspan, openingdate, length
Skyscraper	2,028	developer, engineer, height, architect, cost
PopulatedPlace	181,847	foundingdate, language, area, population
River	10,797	sourceMountain, length, mouth, maxDepth
Organisation	91,275	location, foundationdate, keyperson
Band	14,952	currentMembers, foundation, homeTown, label
Company	20,173	industry, products, netincome, revenue
Educ.Institution	21,052	dean, director, graduates, staff, students
Work	189,620	author, genre, language
Book	15,677	isbn, publisher, pages, author, mediatype
Film	34,680	director, producer, starring, budget, released
MusicalWork	101,985	runtime, artist, label, producer
Album	74,055	artist, label, genre, runtime, producer, cover
Single	24,597	album, format, releaseDate, band, runtime
Software	5,652	developer, language, platform, license
TelevisionShow	10,169	network, producer, episodenummer, theme

Tabla 2. Las clases ontológicas más comunes en DBpedia.

En la Ontología de DBpedia, todas las clases se aglutinan en torno a una sola categoría común, *Thing*, superordinado del cual parten todas las demás clases. Es evidente que esta ingente cantidad de entidades, ya clasificadas de manera sistemática, resulte de extremo interés para el Onomasticón de FunGramKB. Asimismo, otra de las características interesantes de DBpedia es que está conectada con otras fuentes de datos en la Web (Tabla 3), de acuerdo con los principios de los Linked Data (Bizer *et al.*, 2007).

Data Source	No. of Links	Data Source	No. of Links
Freebase	2,400,000	WikiCompany	25,000
flickr wrapp	1,950,000	MusicBrainz	23,000
WordNet	330,000	Book Mashup	7,000
GeoNames	85,000	Project Gutenberg	2,500
OpenCyc	60,000	DBLP Bibliography	200
UMBEL	20,000	CIA World Factbook	200
Bio2RDF	25,000	EuroStat	200

Tabla 3. Vínculos RDF desde DBpedia a otras bases de datos (Bizer *et al.*, 2009).

Esto contribuye a que la información almacenada sea, además de más completa que la disponible en Wikipedia, constantemente actualizada a la vez que lo hacen sus fuentes.

Dentro de cada artículo de Wikipedia, la principal fuente de información para DBpedia son los *info-boxes*. Un *info-box*, situado generalmente a la derecha de un artículo, contiene la información más relevante que se puede encontrar dentro de ese artículo, estructurada en forma de relaciones atributo-valor. Por ejemplo, como muestra la Figura 1, para una ciudad se presenta información sobre el país al que pertenece, la población, la extensión, las coordenadas geográficas, etc.

```

{{Infobox Town AT |
name = Innsbruck |
image_coa = InnsbruckWappen.png |
image_map = Karte-tirol-I.png |
state = [[Tyrol]] |
regbzkg = [[Statutory city]] |
population = 117,342 |
population_as_of = 2006 |
pop_dens = 1,119 |
area = 104.91 |
elevation = 574 |
lat_deg = 47 |
lat_min = 16 |
lat_hem = N |
lon_deg = 11 |
lon_min = 23 |
lon_hem = E |
postal_code = 6010-6080 |
area_code = 0512 |
licence = I |
mayor = Hilde Zach |
website = [http://innsbruck.at] |
}}

```

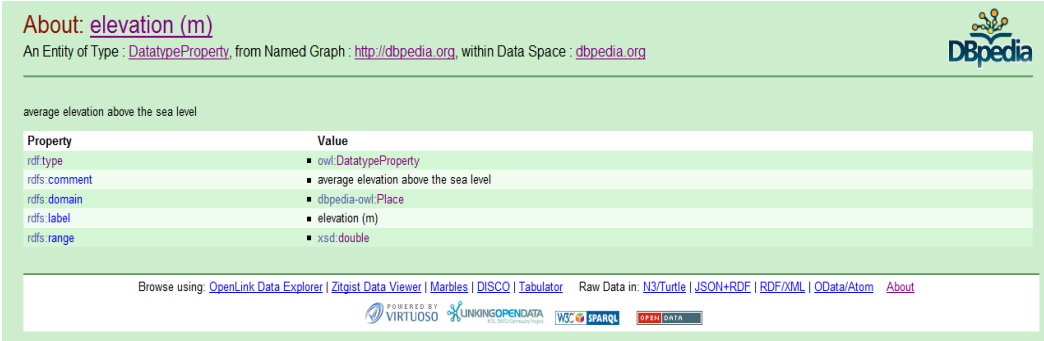


Innsbruck

Country: Austria
State: Tyrol
Administrative region: Statutory city
Population: 117,342 (2006)
Area: 104.91 km²
Population density: 1,119 /km²
Elevation: 574 m
Coordinates: 47°16' N 11°23' E
Postal code: 6010-6080
Area code: 0512
Licence plate code: I
Mayor: Hilde Zach
Website: www.innsbruck.at

Figure 1. Ejemplo de *info-box* en Wikipedia.

Cada clase de DBpedia está estructurada en una serie de propiedades, las cuales se asignarán por defecto a cada entidad que se instancie bajo esa clase y cuyos valores serán tomados a partir de la información que DBpedia extraiga de los *info-boxes* de Wikipedia. Si se accede a cada una de estas propiedades a través de DBpedia, en ocasiones se puede obtener más información acerca de las mismas. Por ejemplo, la unidad de medida en la que se expresa un valor resultará relevante a la hora de construir la plantilla que definirá la propiedad en forma de una predicación dentro del Onomasticón de FunGramKB.¹² A modo de ilustración, las Figuras 2 y 3 muestran los datos que DBpedia proporciona acerca de las propiedades *elevation* y *areaTotal* pertenecientes a la clase *Place*.



About: [elevation \(m\)](#)
An Entity of Type: [DatatypeProperty](#), from Named Graph: <http://dbpedia.org>, within Data Space: [dbpedia.org](#)

average elevation above the sea level

Property	Value
rdf:type	<ul style="list-style-type: none"> owl:DatatypeProperty
rdfs:comment	<ul style="list-style-type: none"> average elevation above the sea level
rdfs:domain	<ul style="list-style-type: none"> dbpedia-owl:Place
rdfs:label	<ul style="list-style-type: none"> elevation (m)
rdfs:range	<ul style="list-style-type: none"> xsd:double

Browse using: [OpenLink Data Explorer](#) | [Zitgist Data Viewer](#) | [Marbles](#) | [DISCO](#) | [Tabulator](#) Raw Data in: [N3/Turtle](#) | [JSON+RDF](#) | [RDF/XML](#) | [OData/Atom](#) | [About](#)

POWERED BY: [VIRTUOSO](#) | [LINKINGOPENDATA](#) | [W3C](#) | [SPARQL](#) | [OPEN DATA](#)

Figura 2. Propiedad *elevation* dentro de la clase *Place*.

¹² Véase apartado 3.1.4.

The screenshot shows a DBpedia page for the property 'area total (m2)'. At the top, it says 'About: [area total \(m2\)](#)' and 'An Entity of Type: [DatatypeProperty](#), from Named Graph: [http://dbpedia.org](#), within Data Space: [dbpedia.org](#)'. The DBpedia logo is in the top right. Below this is a table with two columns: 'Property' and 'Value'. The table contains the following rows:

Property	Value
rdf:type	owl:DatatypeProperty
rdfs:domain	dbpedia-owl:Place
rdfs:label	area total (m2)
rdfs:range	xsd:double
owl:sameAs	http://sw.opencyc.org/concept/Mx8Ngh4rY_n0YBt-QdS5seAT9e7DQ-laHR0cDovL2RicGVkaWEub3Jnl29udG9sb2d5L2FyZWFu63RhbA

At the bottom, there are links for 'Browse using: [OpenLink Data Explorer](#) | [Zigzag Data Viewer](#) | [Marbles](#) | [DISCO](#) | [Tabulator](#)' and 'Raw Data in: [N3/Turtle](#) | [JSON+RDF](#) | [RDF/XML](#) | [QData/Atom](#) | [About](#)'. There are also logos for 'POWERED BY VIRTUOSO', 'LINKING OPENDATA', 'W3C SPARQL', and 'OPEN DATA'.

Figura 3. Propiedad *areaTotal* dentro de la clase *Place*.

Observamos que los valores de las propiedades *elevation* y *areaTotal* se expresan en metros y metros cuadrados respectivamente. Además, como muestra la Figura 2, se puede incluir una breve descripción a modo de aclaración sobre la propiedad en forma de *comment* (p.ej. *average elevation above the sea level*). Como se indicará en el siguiente apartado, toda esta información añadida acerca de las propiedades de una clase es de gran importancia para la representación del conocimiento en COREL.

3 Metodología para la importación de conocimiento cultural a FunGramKB

Una vez que hemos demostrado la importancia del conocimiento cultural, a través de las entidades nombradas, y cómo este conocimiento puede ser encontrado de manera estructurada en bases de datos que han explotado Wikipedia, es preciso describir la metodología que nos permitirá importar dicho conocimiento a FunGramKB. Más concretamente, el proceso de población del Onomasticón de FunGramKB se divide en tres fases:

- i) Fase de construcción: a través de un editor en FunGramKB Suite, se construyen plantillas COREL que permitan plasmar el conocimiento almacenado en DBpedia.
- ii) Fase de proyección: a partir de estas plantillas, se generan los retratos y las historias del Onomasticón.
- iii) Fase de actualización: ya que las fuentes de conocimiento evolucionan, los esquemas conceptuales del Onomasticón también podrán ser actualizados.

Se trata de un enfoque semiautomático de adquisición de conocimiento, donde la fase (i) es asistida, a través de una de las herramientas de FunGramKB Suite (i.e. *DBpedia Mapper*), pero las fases (ii) y (iii) son automáticas.

3.1 Fase de construcción

La piedra angular de todo este proceso es la fase de construcción de las plantillas, en la cual podemos secuenciar los siguientes pasos:

- 1) Relevancia y pertinencia de las propiedades
- 2) Identificación del tipo de bioestructura
- 3) Identificación de la unidad de medida
- 4) Construcción de la plantilla

Aunque el proceso de construcción que describimos en este apartado es válido para proyectar el conocimiento cultural de cualquier recurso al Onomasticón de

FunGramKB, nosotros lo ilustramos con las diecinueve propiedades de *Place* en DBpedia: *areaLand*, *areaTotal*, *areaWater*, *coordinates*, *depth*, *elevation*, *height*, *length*, *location*, *maximumDepth*, *maximumElevation*, *minimumElevation*, *nativeName*, *nearestCity*, *nickname*, *otherName*, *percentageOfAreaWater*, *type* y *width*.

3.1.1 Relevancia y pertinencia de las propiedades

Todo el proceso de población se realiza por medio de un editor conceptual diseñado para este propósito, al cual hemos denominado *DBpedia Mapper* (Figura 4).

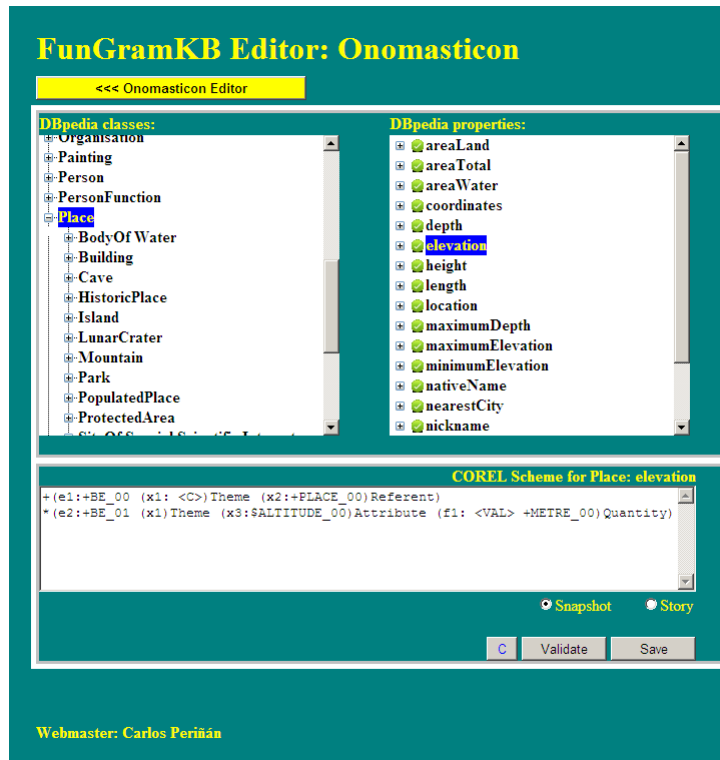


Figura 4. *DBpedia Mapper*.

Observamos que *DBpedia Mapper* se divide en dos partes: una zona de navegación y una zona de edición. En la parte superior, la zona de navegación se divide a su vez en dos listas: la lista de la izquierda contiene la Ontología de clases de DBpedia, mientras que la lista de la derecha presenta un listado de las propiedades vinculadas a la clase seleccionada. En la parte inferior, la zona de edición nos permite almacenar la plantilla COREL asignada a una determinada propiedad, previa validación sintáctico-semántica de la representación formal de la plantilla.

En una primera fase, descartamos aquellas propiedades, dentro del inventario que DBpedia proporciona para cada clase, que no resulten relevantes o pertinentes para FunGramKB. Por una parte, la relevancia de la propiedad se relaciona estrechamente con la posibilidad de que la información pueda ser reutilizada en varios tipos de aplicaciones del PLN, i.e. evaluamos su potencial utilidad en el marco del PLN. Por ejemplo, los datos sobre las coordenadas geográficas de algunos lugares (i.e. *coordinates*) no resultan generalmente relevantes para el PLN. Por otra parte, la pertinencia de la propiedad se relaciona principalmente con su adecuación en la modelización del conocimiento a través del lenguaje COREL. Por ejemplo, los diversos nombres que se puedan utilizar para referirnos a una misma entidad (p.ej. *nativeName*, *nickName*, o *otherName*) son útiles para el PLN, pero no se colocan

dentro de la representación conceptual sino más bien como unidades léxicas vinculadas al concepto en cuestión. La propiedad *type*, la cual indica las clases que sirven como superordinado en la Ontología de DBpedia, tampoco resulta pertinente, debido a que esta información ya está presente en la Ontología de FunGramKB. En definitiva, sólo las propiedades relevantes y pertinentes constituirán un motivo suficiente para la construcción de las plantillas COREL.

3.1.2 Identificación del tipo de bioestructura

Una vez depurado el inventario de propiedades, identificaremos en cada caso si la propiedad en cuestión contribuye a la construcción de un retrato o de una historia. Por ejemplo, las plantillas asignadas a las propiedades *elevation* o *areaTotal* sirven para construir predicaciones pertenecientes a un retrato, ya que el valor que esta propiedad adopte en relación a una entidad determinada tendrá lugar en un marco atemporal. En cambio, propiedades como *birthDate* o *deathDate*, las cuales pueden encontrarse en la clase *Person*, ayudan a construir historias, ya que estas propiedades sirven para enmarcar una determinada entidad en un marco temporal.

3.1.3 Identificación de la unidad de medida

Cuando los valores de las propiedades son numéricos, resulta imprescindible conocer si existe alguna unidad de medida vinculada. Como ya se comentó en el apartado 2.2, los valores de las propiedades *elevation* y *areaTotal* se expresan en metros y metros cuadrados respectivamente en DBpedia. En la plantilla COREL, esta unidad de medida se representaría por medio de un concepto de la Ontología de FunGramKB, ya que todos los conceptos que configuren una plantilla deben estar previamente incluidos en este módulo cognitivo. Esto demuestra que se establece una estrecha relación entre la Ontología, la cual representa el conocimiento del sentido común, y el Onomasticón, el cual representa el conocimiento del modelo del mundo.

Con el fin de identificar el concepto que sirve para representar la unidad de medida, el investigador puede utilizar el buscador conceptual ontológico (Figura 5).

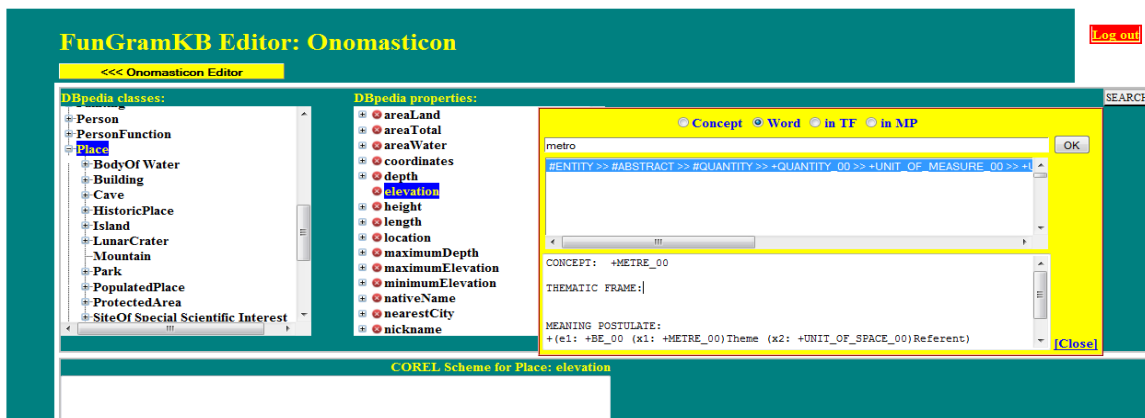


Figura 5. Buscador conceptual ontológico en *DBpedia Mapper*.

Por ejemplo, podemos realizar una búsqueda del concepto a partir de una unidad léxica en inglés o español. Así, podemos saber rápidamente que el concepto al cual se vincula *metro* es +METRE_00. En el hipotético caso de que tal concepto no existiera, entonces sería preciso crearlo en la Ontología de FunGramKB antes de crear la

plantilla COREL en *DBpedia Mapper*.¹³ De esta forma, mantenemos una correlación consistente entre ambos módulos cognitivos.

3.1.4 Construcción de la plantilla

En la fase final del proceso de importación, se construye la plantilla COREL para cada una de las propiedades relevantes y pertinentes de una clase. Una plantilla COREL, la cual está formada por una o más predicaciones, muestra un aspecto similar al de un postulado de significado, con la única diferencia de que se permiten las variables <C> (i.e. la entidad poseedora de la propiedad) y <VAL> (i.e. el valor que se asigna a dicha propiedad). La estructura lógica en que se organiza una plantilla es la siguiente:

- a) La primera predicación contiene la proyección de la clase de la DBpedia a un concepto ontológico de FunGramKB. De esta forma, se establece una plena integración entre el conocimiento del mundo (i.e. Onomasticón) y el conocimiento del sentido común (i.e. Ontología). En el caso de la propiedad *elevation*, la cual pertenece a la clase *Place*, esta primera predicación tendría la siguiente forma:

(3) +(e1: +BE_00 (x1: <C>)Theme (x2: +PLACE_00)Referent)

Podemos observar que el operador de razonamiento + que encabeza la predicación indica que se trata de una predicación estricta, ya que esta propiedad siempre se asigna a entidades de tipo lugar.

- b) Las siguientes predicaciones codifican la descripción de la propiedad, incluyendo, si es necesario, la unidad de medida. Siguiendo con nuestro ejemplo, la propiedad *elevation* se representaría de la siguiente forma:

(4) *(e2: +BE_01 (x1)Theme (x3: \$ALTITUDE_00)Attribute (f1: <VAL> +METRE_00)Quantity)

En esta segunda predicación, ha tenido lugar tanto la conceptualización de la propiedad (\$ALTITUDE_00) como de su unidad de medida (+METRE_00). Por tanto, la variable <VAL> ocupa la posición de un operador de cuantificación que acompaña a la unidad de medida. Observamos también que el operador de razonamiento es ahora rebatible (*), ya que el valor de verdad de esta predicación es susceptible de variar a lo largo del tiempo. Por ejemplo, la altitud de una ciudad sobre el nivel del mar puede verse reducida debido a causas naturales.

Tras validar la representación formal de acuerdo con la gramática de COREL (cf. Perrián Pascual y Arcas Túnez, 2004; Perrián Pascual y Mairal Usón, 2010), la plantilla se almacena finalmente en FunGramKB a través del editor de *DBpedia Mapper* (Figura 4). El Apéndice 1 contiene el inventario de todas las plantillas pertenecientes a las propiedades de *Place* en DBpedia.

¹³ Con el fin de introducir un nuevo concepto ontológico, se requiere acceder al editor de la Ontología de FunGramKB, donde se creará un concepto terminal subordinado a un concepto básico ya existente, además de especificar todas sus propiedades conceptuales, p.ej. el postulado de significado.

3.2 Fase de proyección

Una vez almacenadas las plantillas COREL, éstas permitirán construir automáticamente los retratos e historias que configurarán nuestro Onomasticón. De esta forma, a partir de la plantilla asignada a la propiedad *elevation* de *Place*, *DBpedia Mapper* permite proyectar el conocimiento que DBpedia tiene de esta propiedad a través de las variables <C> y <VAL>, obteniendo así predicaciones como (5-7).

- (5) +(e1: +BE_00 (x1: %TOLEDO_SPAIN_00)Theme (x2: +PLACE_00)Referent)
*(e2: +BE_01 (x1)Theme (x3: \$ALTITUDE_00)Attribute (f1: 529
+METRE_00)Quantity)

Toledo (España) es un lugar ubicado a 529 m de altitud.

- (6) +(e1: +BE_00 (x1: %SAN_FRANCISCO_00)Theme (x2: +PLACE_00)Referent)
*(e2: +BE_01 (x1)Theme (x3: \$ALTITUDE_00)Attribute (f1: 15.8496
+METRE_00) Quantity)

San Francisco es un lugar ubicado a 15,8496 metros de altitud.

- (7) +(e1: +BE_00 (x1: %CASTELLON_DE_LA_PLANA_00)Theme (x2:
+PLACE_00)Referent)
*(e2: +BE_01 (x1)Theme (x3: \$ALTITUDE_00)Attribute (f1: 30 +METRE_00)
Quantity)

Castellón de la Plana es un lugar ubicado a 30 m de altitud.

4 Motivación del proyecto

La construcción de una base de conocimiento de forma automática o semiautomática a partir de recursos ya existentes es mucho más fácil y rápida que un proceso de construcción completamente manual. No obstante, el verdadero beneficio de importar conocimiento a FunGramKB no radica evidentemente en el cambio del sistema notacional, sino más bien en (i) la integración del conocimiento nuevo con todo el conocimiento que ya tenemos almacenado tanto a nivel léxico como conceptual, y en (ii) la reutilización de los mecanismos de razonamiento diseñados para los módulos cognitivos.

La mayoría de ontologías están provistas de una semántica superficial expresada generalmente por medio de relaciones binarias. En cambio, los conceptos ontológicos en FunGramKB están orientados a la semántica profunda, donde no sólo se expresan a través de un lenguaje de representación conceptual más expresivo, sino además los conceptos ontológicos se conectan con las unidades conceptuales de otros módulos cognitivos, i.e. el Cognición y el Onomasticón. Por ejemplo, a partir del postulado de significado de \$ALTITUDE_00, el cual presentamos en (8), el sistema puede inferir que Toledo está a una altura de 529 metros desde el mar.

- (8) +(e1: +BE_00 (x1: \$ALTITUDE_00)Theme (x2: +HEIGHT_00)Referent)
+((e2: +BE_01 (x3)Theme (x1)Attribute)(e3: +MEASURE_00 (x3)Theme
(x4)Attribute(f1: +SEA_00)Origin (f2: +LAND_00)Goal))

O también el conocimiento semántico podría ayudar al sistema a convertir la altitud a kilómetros a través del postulado de significado de \$KILOMETRE_00 (9).

(9) +(e1: +BE_00 (x1: \$KILOMETRE_00)Theme (x2:
+UNIT_OF_SPACE_00)Referent)
+(e2: +BE_00 (x1)Theme (x3: 1000 +METRE_00)Referent)

A esto debemos añadir el hecho de que los conceptos de FunGramKB también están vinculados a unidades léxicas dotadas de entradas con una rica información lingüística. Además de atributos tradicionalmente presentes en los recursos lexicográficos, las unidades léxicas, y más concretamente los verbos, están provistos de una gramática nuclear fundamentada en el aparato teórico de la Gramática del Papel y la Referencia (cf. Mairal Usón y Perrián Pascual, 2009).

Además, ya que los esquemas conceptuales almacenados en los diversos módulos cognitivos de FunGramKB están representados a través del mismo lenguaje formal, i.e. COREL, podemos diseñar un mismo motor de razonamiento que sirva para los conocimientos semántico, procedimental y episódico, facilitando así la integración del conocimiento.

También debemos destacar la posibilidad de reutilización tanto del modelo de plantilla COREL como del motor de migración conceptual. En otras palabras, la metodología descrita en el apartado 3 sirve igualmente para importar conocimiento conceptual de diversa naturaleza. Por ejemplo, podríamos implementar una metodología similar para importar al Onomasticón el conocimiento cultural de otras bases de datos (p.ej. YAGO2), pero también podríamos proyectar hacia una Ontología Satélite el conocimiento especializado de repositorios terminológicos como EcoLexicon (Faber y León Araúz, 2010).

5 Conclusiones

En este artículo, hemos descrito la categorización del conocimiento cultural en proyectos como YAGO, YAGO2 y DBpedia, de los cuales el último —debido a su riqueza de datos y a su capacidad de actualización— ha sido elegido para ayudar a poblar de forma semiautomática el Onomasticón de FunGramKB. Más concretamente, mediante la creación de plantillas de proyección, el ingeniero del conocimiento puede explotar todos los datos de la DBpedia de manera bastante eficiente. Esto es posible porque las plantillas se vinculan a las propiedades de las clases, a las cuales se vinculan las entidades nombradas. Por tanto, una sola plantilla de proyección permite la construcción de una o varias predicaciones para los constructor cognitivos, i.e. retratos o historias, de miles de entidades nombradas.

Agradecimientos

Este trabajo forma parte de dos proyectos de investigación financiados por el Ministerio de Ciencia y Tecnología, códigos FFI2008-05035-C02-01 y FFI2010-15983.

Referencias

- Auer, Sören, Bizer, Christian, Kobilarov, Georgi, Lehmann, Jens, Cyganiak, Richard e Ives, Zachary (2007): "DBpedia: a nucleus for a Web of open data". En: *Proceedings of the 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference*. Berlín-Heidelberg: Springer.
- Baker, Collin F., Fillmore, Charles J. y Lowe, John B. (1998): "The Berkeley FrameNet project". En: *Proceedings of the COLING-ACL*. Montreal, 86-90.
- Bizer, Christian, Cyganiak, Richard y Heath, Tom (2007): "How to publish Linked Data on the Web". [<http://sites.wiwiw.fu-berlin.de/suhl/bizer/pub/LinkedDataTutorial/>]

- Bizer Christian, Lehmann, Jens, Kobilarov, Georgi, Auer, Sören, Becker, Christian, Cyganiak, Richard y Hellmann, Sebastian (2009): "DBpedia: a crystallization point for the Web of data". *Journal of Web Semantics: Science, Services and Agents on the World Wide Web* 7, 154-165.
- De Melo, Gerard, Suchanek, Fabian y Pease, Adam (2008): "Integrating YAGO into the Suggested Upper Merged Ontology". En: *Proceedings of the 20th IEEE International Conference on Tools with Artificial Intelligence*, Ohio.
- Faber, Pamela y León Araúz, Pilar (2010): "Dinamismo conceptual en las bases de conocimiento terminológico: el caso de EcoLexicon". *Íkala. Revista de Lenguaje y Cultura* 15 (25), 75-100.
- Hoffart, Johannes, Suchanek, Fabian, Berberich, Klaus y Weikum, Gerhard (2010): "YAGO2: a spatially and temporally enhanced knowledge base from Wikipedia". Informe técnico MPI-I-2010-5-007, Max-Planck-Institut für Informatik.
- Jiménez Briones, Rocío y Luzondo Oyón, Alba (2011): "Building ontological meaning in a lexico-conceptual knowledge base". *Onomázein* 23, 11-40.
- Jiménez Briones, Rocío y Pérez Cabello de Alba, Beatriz (2011): "An account of selection restrictions in Role and Reference Grammar". *Revista Canaria de Estudios Ingleses* 62, 99-122.
- Kasneći, Gjergji, Ramanath, Maya, Suchanek, Fabian y Weikum, Gerhard (2008): "The YAGO-NAGA approach to knowledge discovery". *SIGMOD Record* 37 (4), 41-47.
- Mairal Usón, Ricardo y Periñán Pascual, Carlos (2009): "The anatomy of the lexicon component within the framework of a conceptual knowledge base". *Revista Española de Lingüística Aplicada* 22, 217-244.
- Nédellec, Claire y Nazarenko, Adeline (2009): "Information extraction". En: Steffen Staab y Rudi Studer (eds.) *Handbook on Ontologies*. Berlín-Heidelberg: Springer, 663-685.
- Periñán Pascual, Carlos (en prensa): "The situated common-sense knowledge in FunGramKB". *Review of Cognitive Linguistics*.
- Periñán Pascual, Carlos y Arcas Túnez, Francisco (2004): "Meaning postulates in a lexico-conceptual knowledge base". En: *Proceedings of the 15th International Workshop on Databases and Expert Systems Applications*. Los Alamitos (California): IEEE, 38-42.
- Periñán Pascual, Carlos y Arcas Túnez, Francisco (2005): "Microconceptual-Knowledge Spreading in FunGramKB". En: *Proceedings on the 9th IASTED International Conference on Artificial Intelligence and Soft Computing*. Anaheim-Calgary-Zurich: ACTA Press, 239-244.
- Periñán Pascual, Carlos y Arcas Túnez, Francisco (2007a): "Cognitive modules of an NLP knowledge base for language understanding". *Procesamiento del Lenguaje Natural* 39, 197-204.
- Periñán Pascual, Carlos y Arcas Túnez, Francisco (2007b): "Deep semantics in an NLP knowledge base". En: *Proceedings of the 12th Conference of the Spanish Association for Artificial Intelligence*. Salamanca: Universidad de Salamanca, 279-288.
- Periñán Pascual, Carlos y Arcas Túnez, Francisco (2010a): "Ontological commitments in FunGramKB". *Procesamiento del Lenguaje Natural* 44, 27-34.
- Periñán Pascual, Carlos y Arcas Túnez, Francisco (2010b): "The architecture of FunGramKB". En: *Proceedings of the Seventh International Conference on Language Resources and Evaluation*. European Language Resources Association, 2667-2674.
- Periñán Pascual, Carlos y Mairal Usón, Ricardo (2010): "La gramática de COREL: un lenguaje de representación conceptual". *Onomázein* 21, 11-45.
- Ruppenhofer, Josef, Ellsworth, Michael, Petruck, Miriam R.L., Johnson, Christopher R. y Scheffczyk, Jan (2006): "FrameNet II: extended theory and practice". Informe

técnico. International Computer Science Institute, University of California at Berkeley.

Suchanek, Fabian, Kasneci, Gjergji y Weikum, Gerhard (2007): "YAGO: a core of semantic knowledge —unifying WordNet and Wikipedia". En: *Proceedings of the 16th International World Wide Web Conference*, 697-706.

Suchanek, Fabian, Kasneci, Gjergji y Weikum, Gerhard (2008): "YAGO: a large ontology from Wikipedia and WordNet". *Web Semantics: Science, Services and Agents on the World Wide Web* 6 (3), 203-217.

Apéndice 1. Las plantillas COREL para la clase Place de DBpedia.

1	TYPE: snapshot
CLASE	Place
PROPIEDAD	/areaLand
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) +(e2:+BE_02 (x3:+LAND_00)Theme (x1)Location (f1:+IN_00)Position) *(e3:+BE_01 (x3)Theme (x4:+BIG_00)Attribute (f2: <VAL> \$SQUARE_METRE_00)Quantity)

2	TYPE: snapshot
CLASE	Place
PROPIEDAD	/areaTotal
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_01 (x1)Theme (x3:+BIG_00)Attribute (f1: <VAL> \$SQUARE_METRE_00)Quantity)

3	TYPE: snapshot
CLASE	Place
PROPIEDAD	/areaWater
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) +(e2:+BE_02 (x3:+AREA_OF_WATER_00)Theme (x1)Location (f1:+IN_00)Position) *(e3:+BE_01 (x3)Theme (x4:+BIG_00)Attribute (f2: <VAL> \$SQUARE_METRE_00)Quantity)

4	TYPE: snapshot
CLASE	Place
PROPIEDAD	/coordinates
ESQUEMA COREL	Se prescinde de esta propiedad por no resultar relevante para el PLN.

5	TYPE: snapshot
CLASE	Place
PROPIEDAD	/depth
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_01 (x1)Theme (x3:+DEEP_00)Attribute (f1: <VAL> +METRE_00)Quantity)

6		TYPE: snapshot
CLASE	Place	
PROPIEDAD	/elevation	
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_01 (x1)Theme (x3:\$ALTITUDE_00)Attribute (f1: <VAL> +METRE_00)Quantity)	

7		TYPE: snapshot
CLASE	Place	
PROPIEDAD	/height	
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_01 (x1)Theme (x3:+HIGH_00)Attribute (f1: <VAL> +METRE_00)Quantity)	

8		TYPE: snapshot
CLASE	Place	
PROPIEDAD	/length	
ESQUEMA COREL	+(e1: +BE_00 (x1: <C>)Theme (x2: +PLACE_00)Referent) *(e2: +BE_01 (x1)Theme (x3: +LONG_00)Attribute (f1: <VAL> +METRE_00)Quantity)	

9		TYPE: snapshot
CLASE	Place	
PROPIEDAD	/location	
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_02 (x1)Theme (x3:<VAL>)Location (f1:+IN_00)Position)	

10		TYPE: snapshot
CLASE	Place	
PROPIEDAD	/maximumDepth	
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_01 (x1)Theme (x3: +DEEP_00)Attribute (f1: <VAL> +METRE_00)Quantity (f2: (e3:+BE_01 (x1)Theme (x4: m +DEEP_00)Attribute))Scene)	

11		TYPE: snapshot
CLASE	Place	
PROPIEDAD	/maximumElevation	
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_01 (x1)Theme (x3: \$ALTITUDE_00)Attribute (f1: <VAL> +METRE_00)Quantity (f2: (e3:+BE_01 (x1)Theme (x4: m \$ALTITUDE_00)Attribute))Scene)	

12		TYPE: snapshot
CLASE	Place	
PROPIEDAD	/minimumElevation	
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_01 (x1)Theme (x3: \$ALTITUDE_00)Attribute (f1: <VAL> +METRE_00)Quantity (f2: (e3:+BE_01 (x1)Theme (x4: p \$ALTITUDE_00)Attribute))Scene)	

13		TYPE: snapshot
CLASE	Place	
PROPIEDAD	/nativeName /nickname /otherName	
ESQUEMA COREL	No se precisa creación de regla, ya que es una redirección a la que se accederá a través del "alternative name box" que se creará a tal fin en el Onomasticón de FunGramKB.	

14		TYPE: snapshot
CLASE	Place	
PROPIEDAD	/nearestCity	
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_02 (x3:<VAL>)Theme (x1)Location (f1: m +NEAR_00)Position)	

15		TYPE: snapshot
CLASE	Place	
PROPIEDAD	/percentage Of AreaWater	
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) +(e2:+BE_02 (x3:+AREA_OF_WATER_00)Theme (x1)Location (f1:+IN_00)Position) *(e3:+BE_01 (x3)Theme (x4:+BIG_00)Attribute (f2: <VAL> \$PERCENTAGE_00)Quantity)	

16		TYPE: snapshot
CLASE	Place	
PROPIEDAD	/type	
ESQUEMA COREL	Se prescinde de regla para esta propiedad, ya que resulta redundante enumerar de nuevo el tipo, puesto que aparecerá en la jerarquía de la Ontología de DBpedia (clase y subclase) presente en el Onomasticón de FunGramKB a través de <i>DBpedia Mapper</i> .	

17

TYPE: snapshot

CLASE	Place
PROPIEDAD	/width
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_01 (x1)Theme (x3:+WIDE_00)Attribute (f1: <VAL> +METRE_00)Quantity)