Universitat de les Illes Balears

escola politècnica superior

Treball Final de Grau

GRAU D'ENGINYERIA INFORMÀTICA

Clasificación de documentos mediante información semántica

ASER AGUILAR JUÁREZ

Tutor
Isaac Lera Castro

Escola Politècnica Superior
Universitat de les Illes Balears
Palma, 11 de julio de 2017

ÍNDICE GENERAL

RESUMEN

Con el incremento de datos en internet, aumenta la necesidad de mejorar las técnicas de clasificación para acceder de manera más eficaz al contenido buscado.

En este marco, se lleva a cabo un enfoque de clasificación de documentos mediante información semántica. Este proceso requiere de dos fases. En la primera se hace uso de herramientas que permiten identificar la relación entre las palabras de una misma frase y establecer su semántica. De esta forma, se obtienen documentos de internet y se transforma su contenido, originalmente en lenguaje natural, en conceptos semánticos. La segunda fase de la propuesta trata de generar manualmente conjuntos de categorías, cada una formada por documentos de contenido semántico, y un proceso que, dado un documento sin clasificar, pueda establecer una relación con las distintas categorías creadas.

CAPÍTULO 1

INTRODUCCIÓN

Internet alberga una gran cantidad de información, descrita en lenguaje natural. Aunque recientemente se están añadiendo etiquetas semánticas en HTML5 para identificar el tipo de información que describen los documentos en la Web y facilitar la lectura de sus distintas secciones, la mayoría de documentos no disponen de estos metadatos.

Con la necesidad de poder clasificar e identificar el contenido de estos documentos y de mejorar la comunicación máquina-humano entre otros problemas relacionados con el lenguaje natural, surge un nuevo campo en las ciencias de la computación, inteligencia artificial y lingüística conocido como el procesamiento del lenguaje natural (NLP).

El NLP tiene como finalidad la comprensión del lenguaje natural y el diseño de mecanismos que permitan una comunicación eficaz computacionalmente.

El procesado de lenguaje natural resulta muy atractivo para empresas como Google, que, al ser el principal buscador a nivel mundial, realiza una gran inversión en mejorar las búsquedas. Tal y como se explica en [1], el objetivo de Google es convertirse en una especie de asistente personal, donde puedas realizar una petición, como si a otro humano se le estuviese haciendo, y que sea capaz de darte una respuesta inmediata. El ejemplo mostrado en el artículo es la realización de la pregunta "how old is justin bieber" donde se recibe la respuesta exacta de "19 years".

En 2012 lanzó Knowledge Graph para mejorar la interpretación semántica de los términos de búsqueda. Algunos de sus aportes, descritos más detalladamente en [2], son el análisis de las palabras de búsqueda en su contexto o el uso de las búsquedas de otros usuarios para mejorar los resultados.

Dentro del NLP, nos vamos a centrar en dos ramas de estudio: la sintaxis y la morfología, que nos van a dar acceso a la semántica del lenguaje.

La sintaxis es la parte de la gramática que estudia el modo en que se combinan las palabras y los grupos que estas forman para expresar significados, así como las relaciones que se establecen entre todas esas unidades [3]. Un ejemplo de cómo afecta la sintaxis a una oración y su significado es la diferencia entre las oraciones 'un hombre sobre un caballo' y 'un caballo sobre un hombre', donde, usando las mismas palabras,

se adquiere un significado totalmente distinto según el orden.

La morfología es la parte de la gramática que estudia la estructura de las palabras y de sus elementos constitutivos [4]. Hace referencia a la palabra como unidad y no su combinación para formar sintagmas, oraciones o frases. Por ejemplo, algunas palabras constituidas por la raíz *leer* son *lectura* o *legible*.

La semántica es el estudio del significado de las expresiones, no sólo en lenguaje natural como pueden ser el español o inglés, sino también un lenguaje artificial o lenguaje de programación

La morfología y la sintaxis nos van a permitir diferenciar los distintos elementos de una frase y nos ayudan a comprender la semántica. Por ejemplo, en inglés, según la gramática empleada, una misma palabra se puede usar como verbo, sustantivo o adjetivo, y su significado va a variar según esta categoría gramatical, de forma que poder disponer de ella nos acerca a la semántica. [5]. Por ejemplo la palabra inglesa *fine* como sustantivo significa *multa*, como adjetivo significa *fino, bueno* o *bonito* y como verbo significa *multar*.

Nuestro aporte se enfoca en encontrar una forma novel de mejorar la clasificación automática de documentos obtenidos de internet mediante el uso de la semántica. Este objetivo se ha conseguido mediante dos fases: la primera consiste en generar un proceso capaz de identificar la sintaxis y morfología del contenido de un documento y a partir de esta información gramatical obtener la semántica de las distintas palabras. En la segunda se obtienen manualmente conjuntos de documentos que pertenezcan a una misma categoría y se hace uso del proceso de la primera fase para cada uno de ellos obteniendo así la semántica que los describe. Una vez generadas las categorías trataremos de procesar un nuevo documento y saber cuál es su similitud con cada una de las distintas clases.

Para la realización del proyecto se ha escogido *Pyhon* como lenguaje de programación por su facilidad de uso, la gran comunidad que da soporte, y la cantidad de librerías en Python para el procesamiento de lenguaje natural.

1.1 Tareas

1. Crear un proceso que dada una URL descargue automáticamente el contenido web en documentos de texto plano

2. Análisis semántico

    a) Desglosar y procesar el contenido de los documentos por frases

        - Analizar la morfología y sintaxis
        - Realizar el análisis semántico

    b) Guardar la semántica de cada palabra y su número de apariciones en un fichero con formato JSON

3. Clasificación de documentos

    a) Procesar semánticamente grupos de documentos y clasificarlos manualmente

b) Procesar semánticamente documentos cuya categoría es desconocida inicialmente por el proceso

c) Calcular la similitud del documento a clasificar con los documentos que definen la categoría y obtener un listado ordenado de los resultados.

CAPÍTULO 2

PROPUESTA Y ESTADO DEL ARTE

2.1 Propuesta

Nuestro aporte tiene como objetivo la clasificación de documentos mediante el análisis semántico. La importancia de la semántica reside en que el significado de las palabras puede ser ambiguo y variar según el contexto, y se ha de eliminar esta ambigüedad para poder conocer cuál es el significado exacto que expresa. Para conocer el significado de las palabras, y poder comparar documentos o frases, el primer paso es disponer de una colección que a cada concepto le otorgue un significado y relación con otros creando un cuerpo formal de conocimiento, una ontología.

Las ontologías son usadas para describir de forma explícita las entidades de un dominio y sus relaciones. "Una ontología es una especificación explícita de una conceptualización", según Thomas Gruber [6]. El profesor C. Llamas nos ofrece en [7] una descripción más detallada de la construcción de ontologías y las distintas aproximaciones, y ofrece un listado de ontologías públicas como DAML, UNSPSC, RosettaNet, DMOZ o CyC. No es importante profundizar en el concepto de ontología en este proyecto. En pocas palabras, una ontología es una taxonomía de conceptos con un mayor número de relaciones.

Entre los distintos esfuerzos de la comunidad por construir una base de datos con relaciones léxicas y conceptual-semánticas encontramos WordNet, una de las ontologías lingüísticas más importantes para la investigación en el campo del análisis. Es un tesauro con una gran cantidad de sustantivos, verbos, adjetivos y adverbios agrupados en conjuntos de acepciones denominados *synsets* [8]. A lo largo del documento utilizaremos el término *synset* en vez de acepción.

Cada *synset* define un concepto semántico y tiene la estructura *nombre.X.00*

- Nombre: Nombre del *synset*.
- X: Tipo de *synset*, estando comprendido en 'v' para verbos, 'n' para sustantivos (*noun* en inglés), y 'a' o 's' para adjetivos.

- 00: Es un entero de dos dígitos que define un índice para los distintos significados con un mismo nombre de *synset* y tipo.

Por ejemplo, los *synsets* de la palabra *tree* y sus significados serían:

- tree.n.01: *a tall perennial woody plant having a main trunk and branches forming a distinct elevated crown; includes both gymnosperms and angiosperms*
- tree.n.02: *a figure that branches from a single root*
- tree.n.03: *English actor and theatrical producer noted for his lavish productions of Shakespeare (1853-1917)*
- corner.v.02: *force a person or an animal into a position from which he cannot escape*
- tree.v.02: *plant with trees*
- tree.v.03 *chase an animal up a tree*
- tree.v.04 *stretch (a shoe) on a shoetree*

A su vez, cada uno tiene un conjunto de *lemas*, que son las distintas palabras que pueden tener el significado que representa el *synset*.

Por ejemplo los *lemas* de *tree.n.02* serían

- tree
- tree_diagram

Las relaciones entre synsets pueden ser de diversos tipos, como una relación con conceptos más genéricos o más específicos como *vehículo-coche*, una relación de tipo 'forma parte de', o 'es un conjunto de' como *árbol-bosque*, siendo un árbol parte de un bosque, y estando un bosque formado por árboles. Una vez construido un corpus con esta información, nos es posible trabajar con conceptos en lugar de palabras clave.

Aunque actualmente existen tesauros para varios idiomas incluyendo el español, se ha decidido hacer uso de WordNet con documentos inglés, debido a sus cerca de 117.000 *synsets* [8] y su amplio uso en la comunidad científica centrada en el procesamiento del lenguaje natural.

2.2 Trabajo relacionado

Entre las múltiples investigaciones sobre análisis semántico, encontramos *Ontology-Based Semantic Online Classification of Documents Supporting Users in Searching the Web* [9], donde de forma semejante a nuestro proyecto, tiene como uno de sus objetivos eliminar la ambigüedad de las palabras mediante Multiwordnet, una expansión de WordNet, y poder obtener resultados más acordes a la búsqueda realizada por el usuario, o el artículo [10] donde, entre otras herramientas, hace uso de WordNet para clasificación de Tweets y menciona alguna de sus desventajas, como es la cadencia de actualizaciones y falta de abreviaciones en sus bases de conocimiento semántico.

La segunda fase de la investigación consiste en la clasificación automática de documentos sobre categorías predefinidas, constando cada una de un conjunto de

documentos sobre los que se ha analizado su semántica. A parte de las dos investigaciones mencionadas en el párrafo anterior, donde también se lleva a cabo la clasificación de documentos, encontramos otros autores con trabajos en esta misma tarea, como puedan ser *Concept-based classification for multi-document summarization* [11], donde basados en el modelo LDA (Latent Dirichlet Allocation) tratan de obtener un resumen de un conjunto de documentos obtenidos por una búsqueda de usuario, y comparan el resumen generado automáticamente con uno generado por personas.

En *A Dimensionality Reduction Approach for Semantic Document Classification* [12] usan una serie de documentos de entrenamiento introducidos por el usuario y que son representados en un espacio de conceptos. Se usa un método de reducción para este espacio y se genera una norma que defina el tópico que engloba a los documentos.

Otra investigación es *Document Classification based on Support Vector Machine using A Concept Vector Model* [13], dónde se realiza una mejora en la precisión de clasificación de documentos mediante el modelo CVM (Concept Vector Model) basándose en SVM(Support Vector Machine).

Nuestra contribución principal consiste en la transformación de un documento descrito en lenguaje natural, en un documento de conceptos semánticos que facilite su clasificación automática dentro de un conjunto de categorías predefinidas, y crear un proceso capaz de realizar esta clasificación.

CAPÍTULO 3

DESARROLLO DE LA PROPUESTA

El desarrollo de nuestra propuesta se presenta en tres apartados principales: Obtención de datos, análisis semántico, y clasificación de documentos.

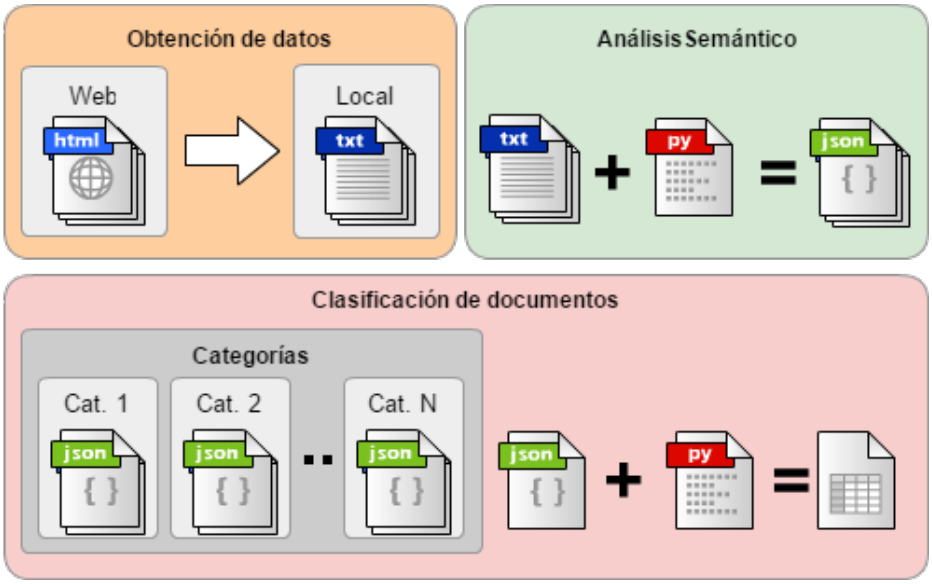


Figura 3.1: Fases del desarrollo

El apartado de obtención de datos describe las herramientas y metodología para obtener los documentos que van a servir para definir las categorías sobre las que se hará la clasificación y los documentos que van a ser clasificados.

El análisis semántico es una primera fase que permitirá que la clasificación de documentos se lleve a cabo sobre conceptos semánticos fáciles de procesar, en vez de sobre lenguaje natural con definiciones ambiguas. El resultado del análisis es un fichero

JSON con los distintos conceptos que definen el documento original y su número de apariciones.

La última etapa es la clasificación de documentos. En nuestro enfoque, esta clasificación se lleva a cabo sobre categorías predefinidas, es decir, hay un proceso manual de selección de documentos que serán agrupados para formar las categorías.

Una vez listas las definiciones de las distintas categorías, se obtienen nuevos documentos para determinar a cuál pertenecen. Antes de realizar la clasificación, el documento también es analizado semánticamente con la finalidad de poder comparar los conceptos que lo definen con los que definen las distintas categorías.

Los resultados se obtienen como una lista ordenada de mayor a menor proximidad con cada categoría.

3.1 Obtención de documentos

Wikipedia [14] es posiblemente una de las mayores enciclopedias de acceso libre que existan, con más de cinco millones de artículos en inglés y sumando varios millones en otros idiomas, tal y como se puede ver en su página de inicio, lo que la convierte en la candidata ideal para obtener múltiples documentos que hagan referencia a una misma categoría. A esto cabe sumar que la estructura de los datos en la página es sencilla, haciendo posible una extracción automática del contenido para cualquier artículo.

Para la descarga de documentos se ha usado la librería de Scrapy para Python. En su página web [15] se puede obtener toda la documentación necesaria para poder descargar contenido de forma rápida y sencilla. Aunque esta herramienta permite personalizar la obtención de datos de la web deseada, y descárgalos en objetos especificados por nosotros, en nuestro caso hemos diseñado una araña bastante sencilla a la que tras indicarle el conjunto de URL's que definen cada categoría, las recorre una a una dirigiéndose mediante una ruta CSS a la sección con la ID que indica que el contenido del artículo para posteriormente mediante XPATH obtener todo el texto. Posteriormente, para evitar errores de codificación, se hace una conversión a ASCII y se eliminan las referencias del texto.

En los últimos años, la comunidad de Wikipedia ha creado una versión semántica de sus bases de conocimiento, la dbpedia[16]. Esta herramienta ha sido descartada de la investigación para permitir extender la adquisición de documentos a otros dominios web no tan estructurados.

3.2 Análisis semántico

Para que la clasificación de documentos sea más precisa, no basta con centrarse en qué palabras definen la categoría, es necesario conocer cuál es su significado, su semántica. Con esta finalidad, tras obtener los documentos que definen cada categoría, se someten a un proceso que transforma el lenguaje natural en conceptos semánticos.

En inglés una misma palabra se puede usar como verbo, sustantivo o adjetivo, y su significado va a variar según la función sintáctica. De esta forma, descubriendo su información morfosintáctica estaremos más cerca de su información semántica. Esto hace necesario el análisis, desglose y estructuración de la información de los documentos antes de iniciar el procesado. El siguiente paso es el análisis semántico de cada palabra

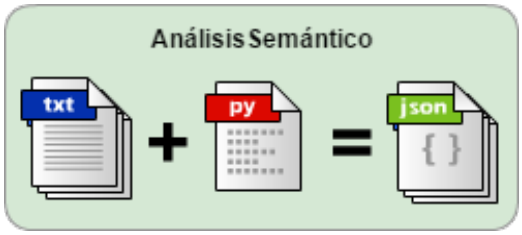


Figura 3.2: Extracción de conceptos semánticos

en su contexto transformando el lenguaje natural en conceptos semánticos. Finalmente se guardan estos conceptos con su frecuencia de aparición en cada documento. El resultado son documentos con la semántica que define cada categoría.

3.2.1 Input

En esta sección se discute la forma en la que son tratados los datos obtenidos por Scrapy para que puedan ser digeridos por el proceso encargado de asignar un significado a cada palabra. El texto del documento inicial se desglosa en frases, entendiendo que una frase es el conjunto de texto hasta encontrar un punto ".". Cada frase es procesada por CoreNLP [16], librería que provee de funciones de reconocimiento de entidades nombradas (NER) y nos permite la generación de un listado de tuplas, donde cada tupla es un conjunto *palabra-descripción morfológica*.

Entre los distintos tipos de palabras que componen el texto, las más relevantes, que nos van a permitir extraer los términos clave que definan la categoría, son los verbos, sustantivos y adjetivos. Otro punto a tener en cuenta, es que cuando se consulta a WordNet una palabra con una función sintáctica, nos va a devolver el mismo resultado a pesar de la forma morfológica indicada en la consulta. Los resultados para *car* serán los mismos que para *cars*, y los resultados de *have* serán los mismos que los de *had*. Por este motivo se hace uso de CoreNLP para normalizar las palabras de forma que los sustantivos quedan en su forma singular y los verbos en infinitivo. Esto facilita que, si una palabra aparece varias veces en el texto con morfología distinta, se pueda procesar una única vez consiguiendo tiempos de ejecución más rápidos.

El resultado final consiste en un listado de verbos, sustantivos y adjetivos normalizados, quedando ya la estructura lista para la fase de procesado.

3.2.2 Procesado

Con el listado generado en el punto anterior, se hace uso de WordNet para, de cada palabra, obtener sus *synsets* correspondientes según su función sintáctica.

Para el procesado se hace uso de un algoritmo de *backtracking* donde se generan todas las posibles combinaciones de *synsets* de cada palabra y se selecciona la combinación de mayor peso.

La cantidad de combinaciones a realizar para n palabras es la multiplicación de los distintos *synsets* aplicables a cada una. Si por ejemplo tenemos tres palabras, A con 7 *synsets*, B con 4 y C con 3, el proceso comprobaría 7 x 4 x 3 = 84 combinaciones.

Para cada par de *synsets* de una combinación, cada uno perteneciente a una palabra distinta, se suma el resultado de aplicar la función de Wu-Palmer. Esta función devuelve un valor denotando la similitud de los dos *synsets* basándose en la profundidad de ambos dentro de la taxonomía, y en su menor *synset* común. Las relaciones que usa WordNet por defecto para la búsqueda del *synset* común son las de hiperónimos e hipónimos, pero con el fin de mejorar los resultados, se han añadido nuevos tipos de relaciones tales como: *un elemento está formado por, un elemento forma parte de, la sustancia del elemento es, tiene como atributo*, y otro tipo de vinculaciones como podría ser *comer-masticar*.

Siguiendo con el ejemplo anterior, para una combinación con un *synset* de A, otro de B, y otro de C, y siendo W el resultado de aplicar Wu-Palmer para dos palabras, se sumarían los resultados de forma $W(A_i, B_i) + W(A_i, C_i) + W(B_i, C_i)$.

Supongamos la oración “insectivores eat crickets".

Synsets por palabra:

- *insectivores*: insectivore.n.01, insectivore.n.02
- *eat*: eat.v.01, eat.v.02, feed.v.06, eat.v.04, consume.v.05, corrode.v.01
- *crickets*: cricket.n.01, cricket.n.02

Algunas de las combinaciones serían:

- Combinación 1 (insectivore.n.01, eat.v.01, cricket.n.01):

  W(insectivore.n.01, eat.v.01) + W(insectivore.n.01, cricket.n.01) + W(eat.v.01, cricket.n.01) = 0 + 0.58 + 0 = 0.58

- Combinación 2 (insectivore.n.01, eat.v.01, cricket.n.02):

  W(insectivore.n.01, eat.v.01) + W(insectivore.n.01, cricket.n.02) + W(eat.v.01, cricket.n.02) = 0 + 0.11 + 0 = 0.11

- Combinación 3 (insectivore.n.01, eat.v.02, cricket.n.01):

  W(insectivore.n.01, eat.v.02) + W(insectivore.n.01, cricket.n.01) + W(eat.v.02, cricket.n.01) = 0 + 0.58 + 0 = 0.58

- Combinación 4: insectivore.n.01, eat.v.02, cricket.n.02 W(insectivore.n.01, eat.v.02) + W(insectivore.n.01, cricket.n.02) + W(eat.v.02, cricket.n.02) = 0 + 0.11 + 0 = 0.11

Como se observa en el ejemplo, el método para encontrar la mejor combinación puede no devolver un único resultado, habiendo varias combinaciones con el mismo peso. Estas situaciones son más propensas en frases cortas con verbos, los cuales en muchas ocasiones no tienen conexión en la ontología con el sustantivo o adjetivo con el que se compara, obteniendo un 0 como resultado en la similitud. Por este motivo, para establecer un desempate entre los distintos resultados posibles, para cada *synset* hemos establecido una distancia que relaciona el nombre, posición e índice del *synset* con la palabra de la que proviene (ver apéndice A.1), y que servirá para en estos casos escoger el de menor distancia.

3.2.3 Output

Los *synsets* referidos a cada palabra de cada frase procesada se guardan junto con el *lemma* de la palabra de la que proceden en una estructura tipo JSON en la que se indica su número de apariciones en el documento. De esta forma, al final del procesado de cada fichero disponemos de un documento con la semántica de cada palabra y su número de apariciones, lo que nos da la información necesaria para generar una categoría.

Además del JSON, se genera otro fichero, donde se informan estadísticas a nivel de procesado de sentencia, y de procesado general del fichero.

Para cada sentencia se obtiene información de sus *synsets* correspondientes, el tiempo de procesado, las combinaciones que se han llevado a cabo para obtener los resultados, las combinaciones estimadas, si ha habido algún error al ejecutar la sentencia, y si se ha tenido que forzar el final del proceso. Aunque en la gran mayoría de ficheros no se produce ningún error de procesado, se han recibido algunos errores en el análisis sintáctico. El proceso encargado de realizar este análisis es un servidor Java que recibe las peticiones mediante el protocolo TCP-IP de forma que el cliente envía la petición y espera recibir respuesta dentro de un tiempo de espera. Se observa que a pesar de haber ampliado significativamente esta espera, en algunos raros casos, con sentencias que incluyen datos numéricos, paréntesis y algún símbolo, se llega a producir *time-out* por el exceso de tiempo requerido por el servidor para procesar la frase. Por ejemplo en uno de los documentos que definen la categoría de armamento, se procesaron 167 líneas, y únicamente una dio error: *"The ring sold ' high - calibre ' machine guns , machine pistols , handguns , explosives (500 pounds of C - 4 plastic explosive) and thousands of rounds of ammunition to biker gangs and other criminals"* .

Finalmente, a nivel de fichero se obtiene información del tiempo total de procesado, total de combinaciones llevas a cabo, de las cuales cuántas con error, y cuantas con final de procesado forzado (ver siguiente punto 3.2.4), tiempo medio de procesado por cada mil combinaciones, y tiempo medio de procesado por sentencia. Estos datos de tiempo de ejecución sirven para darnos una idea de la eficiencia del proceso y la necesidad de optimización.

3.2.4 Eficiencia y optimización

A la hora de valorar los resultados, nuestro conocimiento nos permite saber el significado de las palabras y podemos concretar rápidamente si la idea que tenemos es la misma que nos da el proceso de análisis semántico. En cambio, llevar a cabo esta selección de conceptos semánticos es un problema P-NP y el coste computacional es considerablemente elevado.

Tras la ejecución del proceso con varios ficheros, se obtiene que, de media, para el análisis de 1.000 combinaciones se requieren 10 segundos de ejecución, además, se observa que las combinaciones por frase suelen rondar las 400.000 combinaciones, llegando a detectarse picos de cientos de millones. Suponiendo que no tuviésemos estos picos y teniendo en cuenta que de media los ficheros contienen unas 130 sentencias, en poco más de 16 años obtendríamos el resultado del primer fichero. Estos datos están muy lejos de ser aceptables, por lo que nos vemos obligados a reducir la calidad de los resultados con la finalidad de mejorarlo. Para ello, en el procesado inicial con

CoreNLP, no sólo se obtiene el tipo de cada palabra, sino que las sentencias se dividen en distintos arrays, según las subordinadas. En el procesado, se obtienen los *synsets* de las sentencias iniciales, y el resultado se usa como referencia fija para el análisis de las frases subordinadas. El proceso es iterativo, por lo que si hubiese un nivel de subordinada *n*, se procesaría por separado, teniendo como referencia los *synsets* ya obtenidos en los niveles *1..n-1*. De esta forma reducimos notoriamente el número de palabras que se procesa cada vez, con el hándicap de que las frases subordinadas no aportan contexto para definir el significado de las palabras en las frases principales.

A pesar de esta reducción, se siguen encontrando frases cuyo procesado requiere de un gran número de combinaciones debido principalmente a dos motivos, a que el número de *synsets* de algunas palabras son muy elevados, llegando a alcanzar hasta 30 significados distintos, y a que, a pesar de dividir la frase en subordinadas, no se reduce suficientemente el número de palabras en algún nivel, lo que lleva al establecimiento de dos umbrales en el procesado por frase; la limitación de *synsets* por palabra a 15, ya que no se suele superar esta cantidad, y una limitación temporal del tiempo de ejecución a 2 mil segundos (unos 33 minutos).

Inicialmente se realizó una reducción a 10 *synsets* como máximo por palabra, obteniendo una cantidad de aciertos por documento de aproximadamente un 60 %, por lo que se amplió el límite a 15, aumentando el tiempo de ejecución debido a la necesidad de realizar más combinaciones, pero mejorando los resultado en hasta un 75.4 % .

La distancia definida en A.1 también es usada para reducir el impacto de la limitación de *synsets* por palabra, y de tiempo de ejecución. En el descarte de *synsets* se ordenan de menor a mayor distancia, y se seleccionan únicamente los 15 primeros. En cuanto al umbral de tiempo, el procesado de *synsets* según esta distancia consigue que se procesen primero los mejores candidatos reduciendo el impacto de obtener un resultado antes de realizarse todas las comprobaciones.

Ahora supongamos que surge un caso en el que hay que realizar 2.000.000 de combinaciones, lo cual tardaría aproximadamente 20.000 de segundos. Si detenemos el proceso a 2 mil segundos, hemos procesado un porcentaje considerablemente bajo y estamos seleccionado los *synsets* en buena parte mediante su distancia con la palabra de la que proviene, en vez de por la semántica. Para evitar este ruido en los resultados, usando el tiempo medio de ejecución por cada 1000 combinaciones (10 segundos), y calculando el número de combinaciones requerido para encontrar la solución, se estima el porcentaje de combinaciones que se va a llevar a cabo al obtener el resultado, y se establece un mínimo de un 70 %, de forma que si el mínimo que se va a procesar es inferior, la sentencia queda descartada automáticamente no añadiendo sus *synsets* al documento generado, y evitando su tiempo de procesado.

Eliminación de verbos 'do', 'be', 'have': Estos verbos tienen 14, 15 y 15 synests distintos respectivamente, y son verbos que aportan poco a la hora de identificar la temática de un documento. Su exclusión del procesado lleva a una obtención de resultados más rápida en aquellas frases donde se da su aparición.

Una vez llevados a cabo estos cambios, de los distintos ficheros de estadísticas obtenemos la siguiente información. El tiempo medio de procesado de cada mil combinaciones de *synsets* son 10 segundos, y el tiempo medio por sentencia son 160 (2,7 minutos), por lo que de cada sentencia se hacen una media de 16000 combinaciones de *synsets* para obtener los correspondientes a cada palabra. De cada fichero se procesan de media unas 130 sentencias, a 2,7 minutos cada una, tenemos un tiempo de ejecución

por fichero de 5,85 horas, considerablemente inferior a los 16 años iniciales.

3.3 Clasificación de documentos

El proceso de clasificación se lleva a cabo sobre un conjunto de categorías generadas manualmente, y se obtiene como resultado un listado con las categorías ordenadas según su proximidad al documento a clasificar.

Figura 3.3: Proceso de clasificación

3.3.1 Creación de categorías

Con la información semántica de los ficheros JSON, se lleva a cabo un procesado conjunto a nivel de categoría, para el cuál se han descartado los synsets pertenecientes a verbos debido a su baja tasa de acierto. El resultado seguirá siendo un fichero por documento, pero esta vez con datos que lo relacionen con el conjunto. Este proceso se encarga de, para cada par de ficheros, buscar si algún *synset* aparece en ambos, y en ese caso será incluido en el fichero que definirá la categoría. Además, para cada par de *synsets*, cada uno de un fichero distinto, se comprueba si hay alguna relación en la ontología, y en caso de haberla, cuál es la distancia. Si hay una relación, y la distancia al *synset* común es menor o igual a dos, se añadirá también a la nueva estructura, de esta forma si por ejemplo tenemos en un fichero el *synset* para definir deportivo, y en otro el que define descapotable, para cada fichero se añade el *synset* que define coche, suponiendo que este está justo por encima en la ontología, heredando el número de apariciones del *synset* del que proviene en el documento dado, y consiguiendo así que si un documento a clasificar habla de otro tipo de vehículo, pueda hacerse la asociación con la categoría.

El motivo de haber escogido 2 como distancia para añadir el menor *synset* común, es que tras varias pruebas se observa que con una distancia igual o superior a 4, se añaden conceptos demasiado genéricos perjudicando la definición de las categorías. Con una distancia igual a 3 se obtienen mejoras, pero los conceptos siguen siendo genéricos. Se observa que, si el documento a clasificar pertenece por ejemplo a la categoría ”reptil", en la clasificación pueden obtenerse resultados como "felino” o ”anfibio”. Es decir, se identifica que es un animal, pero resulta complejo identificar una subcategoría, llevando así a restringir la distancia a 2, donde se han obtenido resultados más exactos.

Otro aspecto que se ha tenido en cuenta en este proceso y ha llevado a una mejora de los resultados, es que entre los distintos documentos que definen una categoría, pueden haber apartados que hablen de temas no relacionados con la categoría en sí, e introduzcan ruido en los resultados. Para reducir el impacto de estos casos se pone como condición que los conceptos escogidos aparezcan en al menos un 30 % de los documentos de la categoría.

3.3.2 Clasificación de documentos

El documento a clasificar se requiere que haya pasado previamente por un análisis semántico, de forma que sea el JSON resultante el que se procese. El método de procesado es td-idf, del inglés *term frequency, inverse document frequency*. Esta técnica consiste en la generación de una matriz donde las columnas son los documentos, en nuestro caso los que definen una categoría en concreto más el documento a clasificar, y las filas son los *synsets* existentes en los distintos documentos. Cada celda representa el número de apariciones del *synset* en el documento. Una vez generada la matriz se aplica la fórmula obtenida de [9] para establecer un peso normalizado de cada *synset* en un documento, según su aparición en el resto de documentos.

El peso w para la palabra i en el vector de documento del documento d se calcula como se indica (con N siendo el número total de documentos, y df_i siendo la frecuencia de i en el documento.

$$w_{i,d} = \frac{\overline{w}_{i,d}}{\sqrt{\sum_{j=1}^{n} \overline{w}_{j,d}^2}} \text{ with } \overline{w}_{i,d} = tf_{i,d} \cdot \log\left(\frac{N+1}{df_i}\right)$$

Tras obtener la matriz con pesos para una categoría, de la columna correspondiente al texto a clasificar se hace el coseno junto al resto de documentos y se obtienen la mediana, la media, y el máximo. Este proceso se lleva a cabo con las distintas categorías y a la hora de establecer un resultado se considera como criterio de ordenación que si la categoría A tiene al menos dos de estos valores mayores que la categoría B, A es mejor candidato que B.

En el siguiente ejemplo muestra los resultados obtenidos para *Boeing_AH-64_Apache*, un helicóptero de clase militar

	Tree	Footballer	Computer	Reptil	Anphibian	Feline	Gun	Rotorcraft	Building
Mediana	0.00577	0.00464	0.00356	0.00525	0.00532	0.00391	0.00494	0.00800	0.00628
Media	0.00559	0.00455	0.00357	0.00516	0.00529	0.00417	0.00516	0.00970	0.00611
Máximo	0.00942	0.00746	0.00671	0.00878	0.00891	0.00686	0.01001	0.01599	0.00994

Cuadro 3.1: Resultados de la clasificación del documento *Boeing_AH-64_Apache*

Según el método de ordenación descrito, la primera categoría sería *Rotorcraft*.

CAPÍTULO 4

RESULTADOS

A continuación, se presentan los resultados obtenidos en el proyecto. Este apartado se divide en dos partes correspondientes a la fase de análisis semántico y a la clasificación de documentos, respectivamente. Esta separación facilita el análisis y la extracción de conclusiones, permitiendo discernir los puntos fuertes de aquellas partes del proceso que pueden ser mejoradas o ser fruto de una futura ampliación del estudio.

4.1 Interpretación de los resultados del proceso semántico

Los ficheros resultantes del análisis sintáctico junto con las estadísticas nos permiten ver cada *synset* de qué palabra y qué frase se ha originado, lo que lleva a un trabajo manual en el que se busca en WordNet las definiciones para dicho *synset* y se comprueba si tienen sentido con el contexto del que se extrae. Aunque en la mayoría de casos resulta sencillo establecer que el resultado es o no correcto, no siempre es así. Por ejemplo, una frase hace referencia a una revista, *"according to Forbes Magazine"*. WordNet contiene dos posibles definiciones para *Magazine*.

1. *Product consisting of a paperback periodic publication as a physical object*
2. *A periodic publication containing pictures and stories and articles of interest to those who purchase it or subscribe to it*

Tras una breve búsqueda a cerca de esta revista, se deduce que el significado correcto es el segundo, mientras que el obtenido por el proceso nos ofrece el primero. Aunque este caso fue marcado como erróneo, puede resultar difícil decidir si el concepto está lo suficientemente alejado como para descartarlo.

También encontramos algún error relacionado con el análisis morfológico, donde contracciones como "can't" o "tree's bark"son separadas en dos palabras, llevando a que WordNet interprete la 't' con el significado de *Tonelada*, como unidad métrica, o la 's' como representación del elemento químico *sulfuro*.

Aunque con muy baja frecuencia, también se encuentra algún error en el que la palabra no pertenece a la categoría indicada, como en el caso de *"handles of fishing rods"*, donde CoreNLP etiqueta el sustantivo "handles" como un verbo, lo que hace imposible que se realize correctamente el análisis semántico.

Tras completar el proceso manual de evaluación de los resultados del programa de análisis se observa un porcentaje de aciertos del 75.2 % para sustantivos, 51 % para verbos, y 75.4 % para adjetivos.

Ejemplo de los resultados del análisis semántico de una frase de los documentos que definen la categoría *reptil*.

- Sentencia: *Ribs are found exclusively on the thoracic vertebrae*
- Lemas :
  - rib.n.02.rib: *any of the 12 pairs of curved arches of bone extending from the spine to or toward the sternum in humans (and similar bones in most vertebrates)*
  - find.v.01.find: *come upon, as if by accident; meet with*
  - pectoral.a.01.thoracic: *of or relating to the chest or thorax*
  - vertebra.n.01.vertebra: *one of the bony segments of the spinal column*
- Tiempo de procesado(s): 0.13
- Combinaciones hechas: 91
- Combinaciones previstas: 91
- Final forzado: No
- Error de ejecucion: No

En el ejemplo se ha definido correctamente la semántica de los sustantivos y adjetivos. En cambio, la definición del verbo no es correcta, siendo *detect.v.01* el mejor candidato entre los distintos *synsets* de la palabra, con el significado: " discover or determine the existence, presence, or fact of".

4.2 Resultados del proceso de clasificación

Debido a la cantidad de tiempo requerida para procesar semánticamente cada documento, tanto los que conforman las categorías, como los que se emplean para ser clasificados, el juego de pruebas se lleva a cabo sobre un conjunto reducido que impide poder establecer una tasa de aciertos precisa.

Para la realización de pruebas se han definido nueve categorías conformadas por entre 15 y 20 documentos: árbol, jugador de futbol, ordenador, reptil, amfibio, felino, arma, helicóptero y edificación, y para la clasificación se han seleccionado 5 documentos por cada una, 45 documentos a clasificar en total.

En la Figura 4.1 se muestran los resultados. En las abscisas están representadas las 9 categorias sobre las que se establece la relación de proximidad con el documento a clasificar y en el eje vertical el número de aciertos de cada uno de los 5 documentos para cada categoría.

Se consideran tres tipos de aciertos.

1. 1r Resultado: Si el documento a clasificar obtiene como categoría más cercana la correcta, se considera que se ha obtenido acierto en el primer resultado, marcado en azul, como sería el caso mostrado en el cuadro 3.1 para la clasificación dle documento que describe el helicóptero *Boeing_AH-64_Apache*.

2. 2º Resultado: Si el documento a clasificar obtiene como categoría más cercana una incorrecta, pero la segunda es la correcta, se considera que el acierto se ha dado en el segundo resultado, marcado en naranja. Por ejemplo la clasificación de un documento donde se describe la Boa Constrictor, un reptil, establece la similitud con cada categoría, por orden de mayor a menor, de la forma: Anfibio, reptil, árbol, felino, edificación, helicóptero, jugador de futbol, arma y ordenador, siendo reptil el segundo resultado.

3. Otros: En caso de que la categoría del documneto no se encuentre entre los dos primeros resultados, se marca como "otros", marcado en gris, consideranto la clasificación inválida.

Aunque la obtención de la categoría correcta como segundo resultado no se puede considerar dentro del correcto funcionamiento de la clasificación automática, especialmente al realizarse sobre un número reducido de categorías, estos datos nos permiten identificar un rango de resultados potencialmente correctos, que podrían incluirse en un primer resultado con las correcciones adecuadas. En el capítulo Discusión se sugieren algunas mejoras que podrían llevar a alcanzar tal meta.

De los 45 documentos, el 60 % ha obtenido su categoría como primer resultado, y un 15 % como segundo.

Del 24 % restante, un 45 % equivale a los documentos de la categoría *Feline*, ya que ninguno de ellos se ha clasificado correctamente, obteniendo siempre como primeros resultados *anfibio* y *reptil*. Tras revisar los resultados del análisis semántico para los documentos que definen la categoría *Feline*, se observa que a la palabra *cat* se le ha asignado el *synset* "computerized_tomography.n.01" un gran número de veces, habiendo documentos en el que este concepto semántico presenta hasta 92 ocurrencias. La definición del concepto es "*a method of examining body organs by scanning them with X rays and using a computer to construct a series of cross-sectional scans along a single axis*", lo que supone que una de las principales palabras para la definición de la categoría tiene un concepto erróneo.

Debido a las similitudes en las definiciones y hábitat de los reptiles y anfibios, se realiza un mayor desglose en los resultados de su clasificación.

De los 10 documentos, 8 han obtenido su categoría correcta en el primer resultado, y como segundo resultado, en el caso de los reptiles, la categoría *anfibio*, y viceversa. Los dos documentos restantes corresponden a un reptil que ha obtenido como primeros resultados *anfibio*, y *reptil*, y a un anfibio con los resultados opuestos. En la sección 3.3.1 se explicó que al generar las categorías se buscaban, para cada par de palabras de dos documentos, un hiperónimo con una distancia no mayor a dos saltos en la taxonomía entre la palabra del documento y el hiperónimo. Esta adición de conceptos a la categoría, pretende facilitar la identificación del tópico del documento a clasificar, pero intentando que la ampliación de conceptos más genéricos no dificulte el proceso cuando las categorías son semejantes. Los resultados para *anfibio*, y *reptil* indican que

la selección de esta distancia no ha añadido conceptos demasiado genéricos como para impedir que se seleccione correctamente la categoría.

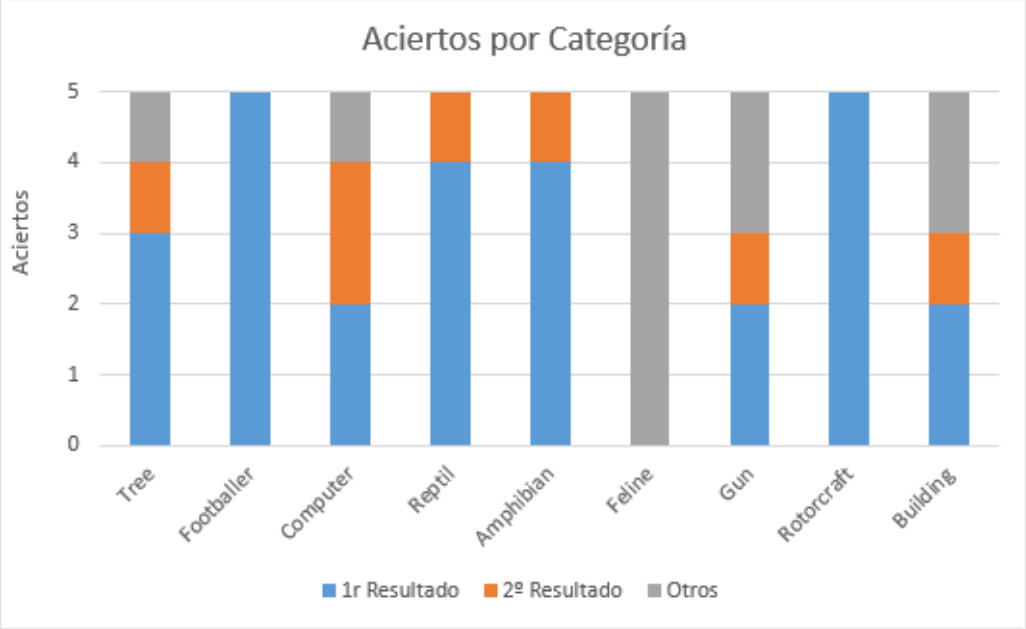


Figura 4.1: Resultados de la clasificación

Como se comentó al principio, las categorías se conforman por entre 15 y 20 documentos. Se ha buscado relación entre el número de documentos que componen cada categoría, con la tasa de aciertos, sin ver una relación clara. Por ejemplo, las categorías *gun* y *building* tienen la misma tasa de aciertos, estando la primera categoría conformada por 4 documentos más que la segunda. Aunque ver cómo afecta esta composición de las categorías a la tasa de aciertos resulta un dato interesante a la hora de crear nuevas categorías, el hecho de estar tratando con sólo nueve categorías dificulta la extracción de conclusiones.

CAPÍTULO 5

DISCUSIÓN

Los resultados del análisis semántico muestran una gran diferencia de aciertos entre los sustantivos y adjetivos, y los verbos, además de dificultades a la hora de analizar contracciones en las palabras, por lo que la falta de actualizaciones de las ontologías supone un hándicap en la búsqueda de conceptos que definan un documento mediante el uso de estas herramientas. A pesar de estas dificultades, se alcanza hasta un 75 % de aciertos en la definición de adjetivos y sustantivos. Estas categorías gramaticales son las que contienen mayor significado a la hora de realizar descripciones, permitiendo que la exclusión de los verbos en la clasificación automática no tenga un gran impacto sobre los resultados.

Aunque la mejora de aciertos mediante la actualización de los tesauros se escapa del alcance del proyecto, hay otras vías de estudio que permitirían perfeccionar los resultados.

- Posicionamiento del texto: Esta mejora consistiría en no dar peso a las palabras únicamente por su número de apariciones en el documento, sino dar mayor relevancia según su posición, como serían títulos o encabezados.

- Selección múltiple: En el modelo investigado, en caso de que más de un *synset* presente la misma similitud para una palabra en su contexto dado, se usa una función que escoge la semántica mediante otros métodos. Otra alternativa sería un modelo que en estos casos dé los distintos *synsets* como válidos y los incluya en la definición del documento en análisis.

Mejoras basadas en el rendimiento podrían ser el estudio de cómo afecta descartar algunos tipos de relaciones entre *synsets* reduciendo así el número de combinaciones, o el uso de heurísticas que permitiesen acotar la extensión del problema.

Respecto a la clasificación, habría que realizar un estudio con un mayor número de categorías y documentos que definan cada una para poder asegurar que la tasa de acierto para el primer resultado es del 60 %.

Como métodos alternativos o adicionales para mejorar la tasa de acierto o reducir los tiempos se hacen las siguientes propuestas:

- Comparación de documentos: Probar otros métodos de comparación y ordenación de resultados. Por ejemplo, la realización de pruebas con un número mayor de categorías y documentos clasificados, nos podría dar pistas de la relevancia que tiene la media, mediana y máximo durante la clasificación, permitiendo asignar distintos pesos a cada uno de estos valores y mejorar así la tasa de acierto.

- Concentración de conceptos: Entre los datos proporcionados por CoreNLP durante el análisis morfológico encontramos el posicionamiento de las distintas palabras, permitiendo así identificar si dos conceptos que definen una categoría se encuentran más o menos próximos. Añadir valor a esta proximidad podría mejorar los resultados. Por ejemplo, en un documento sobre felinos, una palabra que aparezca cerca de este concepto puede ser mejor candidata para definir la categoría, que otro que se encuentre en un párrafo más alejado.

- Tamaño de los umbrales: Se trata de realizar un estudio de cómo afectaría reducir los umbrales de tiempo de ejecución y *synsets* por palabra del análisis semántico en la clasificación de textos.

CAPÍTULO 6

CONCLUSIÓN

6.1 Conclusión

La propuesta presentada tenía como finalidad llevar a cabo el análisis semántico de documentos obtenidos de la web y su clasificación en categorías.

Ha sido una propuesta ambiciosa sobre un campo del que apenas se ha arañado su gran cantidad de contenido y posibilidades de desarrollo.

El análisis semántico se ha conseguido desglosando el contenido de los documentos, analizándolo morfosintácticamente y haciendo uso de una ontología para obtener los conceptos semánticos más aproximados a cada palabra según el contexto. El objetivo de este punto se ha alcanzado con un alto grado de satisfacción a excepción del análisis de verbos donde únicamente se consiguen aciertos en la mitad de casos. En cuanto a velocidad de procesado, a pesar de haber conseguido una gran reducción de tiempo, la investigación podría ampliarse con la búsqueda de métodos que consigan unos resultados más rápidos.

Para generar cada categoría sobre la que se hará la clasificación automática, se han agrupado documentos seleccionados manualmente. Posteriormente se han analizado para determinar la semántica que define la categoría. La semántica del documento a clasificar es comparada con la que describe las distintas categorías y se obtiene una lista ordenada según la aproximación.

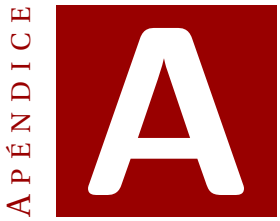
En este apartado, la baja cantidad de categorías y de documentos que conforman cada una, impide establecer una exactitud de los resultados. No obstante, los valores obtenidos con un 60 % de aciertos en el primer resultado, y un 75 % entre el primer y segundo resultado, presentan un buen punto de partida.

Respecto a las herramientas empleadas, se ha escogido *Pyhon* como lenguaje de programación y se han utilizado diferentes tecnologías como: *Scrapy* para la obtención de documentos de la red, *CoreNLP* para procesar y analizar texto sintáctica y morfológicamente, y el tesauro *WordNet*, que ha sido el núcleo del proyecto al permitir establecer la definición semántica de los documentos.

6.2 Valoración personal

A nivel personal el proyecto ha sido un constante reto. No sólo por abordarlo sin ningún conocimiento del procesamiento del lenguaje natural, sino también por desconocer el uso de las herramientas, el lenguaje de programación empleado, y la cantidad de conflictos entre versiones de las distintas herramientas y librerías.

A pesar de la baja tasa de acierto en el análisis semántico de verbos, obtener cerca de un 75 % en sustantivos y adjetivos ha superado las expectativas. Sin embargo, las casi seis horas de media requeridas para el análisis de cada documento y el tiempo empleado para optimizar los algoritmos ha supuesto que las pruebas de clasificación de documentos se llevasen a cabo con un número reducido de categorías, pocos documentos para definir cada una, y un número insuficiente de documentos clasificados como para poder detallar una tasa de acierto exacta en la clasificación.



APÉNDICE

A.1 Distancias entre una palabra y su synset

Antes de explicar la metodología llevada a cabo para establecer las distancias entre una palabra y su *synset*, hay que tener en cuenta que la finalidad de la función es llegar a desempates como candidatos ya que se están encontrando dos *synset* con la misma similitud, y por lo tanto la lógica empleada se basa en la morfología. A la hora de establecer cuál es la distancia entre una palabra y su *synset* se han tenido en cuenta 4 elementos:

1. *LEV*: La distancia de Levenshtein entre la palabra y el nombre del *synset*
2. *IND*: El índice del *synset*
3. *LEM*: El número de *lemas* del *synset*
4. *N*: La cantidad de palabras (verbos, sustantivos y adjetivos) que contiene la frase de la cual se ha obtenido el *synset*.

La distancia se define:

$$(LEV * \ 1.5) + \left(\frac{IND}{4}\right) + \left(\frac{LEM}{N}\right)$$

Con esta fórmula se valora negativamente una mayor distancia de Levenshtein, un alto índice del *synset*, o un alto número de *lemas* en la palabra. Al multiplicar la distancia de Levenshtein por 1.5 hacemos que el parecido entre la palabra y el nombre del *synset* sea un valor decisivo a la hora de escoger uno u otro. La división entre 4 del índice se debe a que se ha observado que cuanto mayor es el índice, más rebuscada o menos frecuente es la definición, pero tampoco queremos perder de vista que una palabra puede tener varias definiciones que se usen habitualmente, por lo que se busca reducir la importancia de este parámetro. Finalmente, en la división del número de *lemas* entre la cantidad de palabras del contexto, hay que tener en cuenta que la evaluación de

la palabra se está llevando a cabo en un contexto donde hay más términos que están siendo evaluados, en este sentido, cuantas menos palabras haya, más difícil será poder establecer el significado de cada una, y por lo tanto se penalizan aquellos *synsets* más ambiguos, que puedan hacer referencia a diferentes palabras (lemas).

BIBLIOGRAFÍA

[1] J. Bhattacharya, "Google hummingbird | análisis detallado de búsqueda semántica y su papel en el algoritmo colibrí," 2013. [Online]. Available: http://www.seofreelance.es/google-hummingbird-analisis-detallado-busqueda-semantica/ 1

[2] E. Checa, "Evolución semántica de google: De meta-web a knowledge graph, más web semántica," 2012. [Online]. Available: https://herramientaseo.wordpress.com/2012/05/21/evolucion-semantica-de-google-de-metaweb-a-knowledge-graph/ 1

[3] "Diccionario de la lengua española: sintaxis," 2017. [Online]. Available: http://dle.rae.es/?id=XzfiT9q 1

[4] "Diccionario de la lengua española: morfología," 2017. [Online]. Available: http://dle.rae.es/?id=Pp2aAE 1

[5] R. H. Thomason, "What is semantics?" 2012. [Online]. Available: https://web.eecs.umich.edu/~rthomaso/documents/general/what-is-semantics.html 1

[6] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220, 1993. 2.1

[7] C. Llamas, "Desarrollando una ontología sencilla," *Dpt. Informática, Universidad de Valladolid*, 2002. 2.1

[8] M. Lovett, "Wordnet. a lexical database for english," 2015. [Online]. Available: https://wordnet.princeton.edu/ 2.1

[9] E. W. De Luca and A. Nürnberger, "Ontology-based semantic online classification of documents: Supporting users in searching the web," *Programming and Coraputer Software*, vol. 26, no. 4, pp. 199–206, 2000. 2.2, 3.3.2

[10] S. S. Q. Li *et al.*, "Using paraphrases to improve tweet classification: Comparing wordnet and word embedding approaches," *2016 IEEE International Conference on Big Data (Big Data)*, pp. 4014–4016, 2016. 2.2

[11] A. C. D. Hakkani-Tür, "Concept-based classification for multi-document summarization," *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5540–5543, 2011. 2.2

[12] O. Ahlgren, P. Malo, A. Sinha, P. Korhonen, and J. Wallenius, "A dimensionality reduction approach for semantic document classification," *SPIM'11 Proceedings*

of the Second International Conference on Semantic Personalized Information Management, vol. 781, pp. 114–121, 2011. 2.2

[13] S. Deng and H. Peng, "Document classification based on support vector machine using a concept vector model," *IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)*, pp. 473–476, 2006. 2.2

[14] "Wikipedia, the free encyclopedia," 2017. [Online]. Available: https://www.wikipedia.org/ 3.1

[15] "Scrapy, an open source and collaborative framework for extracting the data you need from websites," 2017. [Online]. Available: https://scrapy.org/ 3.1

[16] "Dbpedia, towards a public data infrastructure for a large, multilingual, semantic knowledge graph," 2017. [Online]. Available: http://wiki.dbpedia.org/ 3.1, 3.2.1