UNIVERSIDAD DE DEUSTO

HACIA UN SISTEMA DE MARKETING DIRIGIDO MÁS EFICAZ Y PERSONALIZADO EN REDES SOCIALES

Tesis doctoral presentada por
PATXI GALÁN GARCÍA
para el título de
DOCTOR EN INGENIERÍA INFORMÁTICA Y TELECOMUNICACIONES

Directores
Dr. CARLOS LAORDEN GÓMEZ
y
Dr. PABLO GARCÍA BRINGAS

Bilbao, 14 de Junio de 2016

UNIVERSIDAD DE DEUSTO

HACIA UN SISTEMA DE MARKETING DIRIGIDO MÁS EFICAZ Y PERSONALIZADO EN REDES SOCIALES

Tesis doctoral presentada por
PATXI GALÁN GARCÍA
dentro del Programa de Doctorado en
INGENIERÍA EN INFORMÁTICA Y TELECOMUNICACIÓN

Dirigida por el
Dr. CARLOS LAORDEN GÓMEZ
y por el
Dr. PABLO GARCÍA BRINGAS

Author Co-advisor Co-advisor

Bilbao, 14 de Junio de 2016

———————————-

Raquel

Eres mi estrella polar en las oscuras noches, mi brújula en la locura, mi calma en la tormenta, el mechero de mi gasolina, mi uno y mi cero, mi pepito grillo, mi mayor manía y mi mejor respuesta, mi numero par, mi compañera, mi vida....

- - - -

Te quiero

———————————-

Mama, abuela y abuelo

Una familia se da, no se elije. Yo tengo suerte de que la fortuna me haya sonreído. Sin vosotros nada de esto nunca hubiera sido posible.

- - - -

Espero que os sintáis orgullosos.

———————————-

Cuadrilla

Toda aventura necesita amigos y compañeros, se que he elegido bien.

- - - -

Ultras.

———————————-

Abstract

The e-commerce paradigm has completely changed the world, globalizing the impact of companies and allowing the creation of new types of businesses with presence exclusively on the digital world. Being advertising one of the most lucrative forms of business, mainly on-line advertising, it has also become a differentiating and required tool for most organizations to promote their services and products.

One of the advertising channels that has been increasing in usage is the Internet. The number of ads in this medium has grown impressively. The reason is simple, using the Internet is cheap and allows one to easily reach millions of people every day. Unfortunately, the perfect announcement may not be sent to the most receptive users if the target user's preferences have not been taken into account.

Furthermore, with the growth of social networking, the era of smartphones and the boom of the big data paradigm, information retrieval (IR) systems have flourished in analyzing and extracting conclusions from the vast amount of freely available data on the Internet, mostly personal information from users. Some companies gather the users data to create profiles based on users' preferences, behavior or interactions to offer advertisements or even to directly sell users' information.

Being the key factor to "know" the users, these IR systems gather as much information about them. As possible, to generate profiles based on a very large temporal window. Although the general basis allows for the creation of very detailed user profiles, it may generate a problem of obsolescence if the weight of real-time interactions are not taken into account. Because one of the main objectives of user profiling is to send a directed advertisement, not being able to successfully target the users' needs leads to a spam problem (i.e., causes users to consider the message as disturbing).

A solution to obtain a more up-to-date profile from the user resides in a relatively old communication system, short message service (SMS), which is currently more popular in its adaptation to new technologies

and is known as instant messaging (IM) systems, including WhatsApp, Telegram, Facebook Messenger, Hangouts or Line. These systems, despite not having a limitation in terms of interaction length, are commonly used in a dialogue manner, that is, using short sentences. Those interactions represent an important and up-to-date source of information about users but entail an important drawback, which is the difficulty to interpret them due to the insufficient context about each interaction.

Additionally, linking a user's preferences with ad information is an area that connects marketing with natural language processing, and currently, this advertising approach generates ads for users that are higher in quality and have more impact than ads generated using other approaches.

In light of this background, this work proposes one methodology to extract the context from short sentences, based on a newly created DBpedia topics taxonomy, and to offer more directed and real-time advertising, based on an Amazon Bookstore category taxonomy with the aim of increasing the effect of the delivery system. We also have created a resource (map) that links Amazon Book Store categories with DBpedia article topics and created 2 new datasets of short sentences: one labeled using DBpedia topics, to link them with their context, and the other labeled using Amazon book categories, giving suggestions about products related to specific contexts. Moreover, we generated definitions about DBpedia topics and Amazon books categories, which were either missing or not descriptive enough, to help subsequent works in the task of using DBpedia's and Amazon's categorization systems. In addition, we provide a baseline to compare new aproaches using these metodologies and resources. Finally, we have created a proof of concept, in the form of a gift wizard, that, from one sentence, word or term, and by using deep learning techniques, make 3 steps. i) Offers DBpedia topics, as a context, according with the given input. ii) With selected contexts topics, filter Amazon Books Store categories according with our created relational map. iii) Give the categories that have books with information so close to the input and the access to that category on Amazon.

Resumen

El paradigma del comercio electrónico ha cambiado por completo el mundo, globalizando el impacto de las empresas y permitiendo la creación de nuevos tipos de negocios con presencia, exclusivamente, en el mundo digital. Teniendo en cuenta que la publicidad es una de las formas más lucrativas de negocio, principalmente la publicidad en línea, también se ha convertido en un diferenciador y herramienta necesaria para la mayoría de las organizaciones para dar a conocer sus servicios y productos.

Uno de los canales de publicidad que ha ido en aumento es Internet. El número de anuncios en este medio ha crecido de manera impresionante. La razón es simple, el uso de Internet es barato y permite llegar fácilmente a millones de personas todos los días. Desafortunadamente, el anuncio perfecto puede no ser enviado al o a los usuarios más receptivos si no se han tomado las preferencias del usuario en cuenta.

Por otra parte, con el crecimiento de las redes sociales, la era de los teléfonos inteligentes y el auge del paradigma “Big Data”, los sistemas de recuperación de información (RI) han florecido explotando el análisis y la extracción de conclusiones a partir de una gran cantidad de datos. Estos datos están disponibles libre y gratuitamente en Internet, sobre todo información personal de los usuarios. Algunas empresas recopilan estos tipos de datos para crear perfiles basados en las preferencias, comportamiento o interacciones de los usuarios para ofrecerles anuncios o incluso para vender directamente a otras empresas esta información.

Siendo el objetivo “conocer” a los usuarios, estos sistemas de RI reúnen la mayor cantidad de información posible acerca de los objetivos, para generar perfiles basados en los datos generados en una ventana temporal muy amplia. Aunque es viable y óptima la creación de perfiles muy detallados de usuarios, existe un gran problema de obsolescencia si el peso de las interacciones en tiempo real no se tiene en cuenta. Debido a que uno de los principales objetivos de la generación de perfiles de usuario, es enviar anuncios dirigidos, al no ser capaces de dirigirse con éxito a las necesidades actuales que los usuarios reclaman, la mayoría

de los sistemas desembocan a un problema de SPAM (es decir, hace que los usuarios consideren el mensaje como no deseado).

Una solución para obtener un perfil de usuario más actualizado reside en un sistema de comunicación relativamente antiguo, el servicio de mensajes cortos (SMS). Este sistema se ha modernizado y actualmente es más popular en su adaptación a las nuevas tecnologías. Ahora se conocen como sistemas de mensajería instantánea (MI). Algunos ejemplos de estos sistemas son WhatsApp, Telegrama, Facebook Messenger, Hangouts, Telegram o Line. Estos sistemas, a pesar de no tener una limitación en cuanto a la longitud de interacción, se utilizan comúnmente en forma de diálogo, es decir, utilizando frases cortas. Esas interacciones representan una importante fuente de datos actualizada sobre los usuarios. El problema es que conllevan una desventaja importante, la dificultad para interpretar las conversaciones debido al insuficiente contexto sobre cada interacción.

Por otra parte, el vínculo entre las preferencias del usuario con los datos del anuncio es un área que conecta la comercialización o marketing con el procesamiento del lenguaje natural. En la actualidad, este enfoque genera anuncios más efectivos para los usuarios que los anuncios generados mediante otros enfoques.

A la luz de estos antecedentes, en este trabajo se presenta una metodología para extraer el contexto de frases cortas, basadas en una nueva taxonomía temática de DBpedia, capaz de ofrecer publicidad en base a una taxonomía de las categorías de libros existentes en la tienda on-line de libros de Amazon. También se ha creado un recurso (llamado mapa) que une las categorías de la tienda de libros de Amazon con las temáticas o “topics” de los artículos que contiene DBpedia. Además, se han creado dos nuevos conjuntos de datos que contienen frases cortas etiquetadas. La primera utiliza las temáticas de DBpedia para vincular las frases con su contexto, y el segundo utiliza las categorías de la tienda de libros de Amazon para vincular las frases con los posibles productos sugerentes. Por otra parte, se han generado definiciones sobre las temáticas de DBpedia y las diferentes categorías de libros de la tienda on-line de Amazon. Estas definiciones, en el caso de Amazon no existían o no eran públicas y en el caso de DBpedia, no existían o no eran lo suficientemente descriptivas. El motivo para crearlas es ayudar en los posteriores trabajos que requieran la tarea de utilizar sistemas de categorización de Amazon y DBpedia. Además, se proporciona con este trabajo un “baseline” para comparar los nuevos enfoques planteados en este campo y los posibles futuros enfoques que surjan y que utilicen las metodologías y

recursos que puedan derivar de esta u otras investigaciones. Por último, se ha creado una prueba de concepto en forma de un asistente para reglaos que, partiendo de una frase, palabra o término, y mediante el uso de técnicas de aprendizaje profundas o “deep learning” hace 3 pasos. i) Ofrecer temáticas de DBpedia como un contexto determinado en base a la entrada. ii) Con las temáticas seleccionadas filtrar y obtener las categorías de libros de Amazon más afines a dicho o dichos contextos en base al mapa relacional creado. iii) Ofrecer estas categorías de libros de Amazon al usuario porque la información que contienen sus libros son los que tienen más relevancia con el texto de entrada.

Agradecimientos

Es muy difícil acordarse de todas las personas que han ayudado a llevar este trabajo a buen puerto. Sobre todo para una persona que se suele olvidar de si ha cerrado bien o no la puerta de casa esta mañana. Pero aun así, voy a hacerlo, pero pido perdón a todas esas personas de las cuales me he olvidado. Lo siento.

En primer lugar quiero dar las gracias a Pablo García Bringas por haberme dado la oportunidad formar parte del S3lab y ser director de mi tesis. Me acuerdo cuando vinisteis metiendo miedo a clase Javi y tú para reclutar a estudiantes. Participar en este grupo me ha dado mucho conocimiento y nuevos compañeros y amigos. Gracias Pablo. A Carlos Laorden, el otro director de esta tesis, le doy las gracias, no solo por haberme conducido por los diferentes caminos de la investigación, si no por todo el tiempo que ha dedicado e invertido en esta tesis y, en general, en mí. Sus locuras con la domótica, las discusiones sobre seguridad, formas de actuar, la ayuda con la conferencia de Hawái, el detalle del *born to be discover*, sembrar el interés por los dispositivos rooteados y creer en mi me ha ayudado bastante. Gracias Carlos.

Con el equipo que conforman o han participado en el S3lab, en general, siempre he aprendido algo de cada uno de ellos. Sobre todo con los becarios que han estado por el lab. De cada uno de ellos he aprendido algo. Individualmente, tengo que dar las gracias a Sergio Huerta, por la ayuda con las formulas, por los partidos de pádel, por recomendarme restaurantes en Burgos y por escucharme divagar sobre clases y patrones. A Xabi Cantero por su inmenso conocimiento de Linux, su sinceridad y toda la ayuda que me ha dado con los millones de archivos y los *conjuros* de Shell, con los php para los servicios que hemos levantado y para el *PoC* entre otras cosas. Eres un grande, eso que comes debe ser impresionante, comparte tu secreto. A Irene Díez, Guillermo Mariscal, Xabi Ugarte e Iskander Sanchez gracias por haber sido mis expertos en la tediosa tarea de etiquetar los datasets. Los chupa chuses los teníais bien merecidos.

A Javi Nieves, has sido un referente para mí en varios aspectos, tanto técnicos como personales y por sus recomendaciones para programar.

Me ayudaron muchísimo durante la tesis, pero para recomendar ciertas películas, ejem. A Borja Sanz, gracias por tu ayuda en la RECSI, por darme la oportunidad de dar clase en el máster, por estar siempre un paso por delante mío en información hasta que el poder te llamo, por lo de la estancia, y por seguramente más cosas que ni sepa. Se echa de menos las charlas de tecnológica. A Iván García por esas charlas sobre seguridad, además de las conversaciones a 3 con Carlos y los "buenos días". Se nos quedó en el tintero lo de los coches, entre otras cosas. Así ¿cómo lo vamos a conseguir?

A Mikel S. por esos grandísimos posters que haces. No eres un ingeniero, eres un artista. A Asier G. que siempre estaba con una sonrisa cuando le iba a pedir una máquina virtual o que mejorase las que ya tenía. A Jorge D. por los consejos y conversaciones que hemos mantenido, además de por ser el "otro" de NLP en el lab. Ninguno de los dos tenemos ni idea de lo que realmente es esto, pero salimos adelante. A José G. por toda la información y ayuda que me ha ofrecido para la estancia y en general. A Iker P. al igual que al resto, por los consejos y por azuzarme a hacer SPAM. A Igor S. por darme la oportunidad de participar en varias investigaciones, estar en el *Reading group* de seguridad y por darme ciertas pautas a tener en cuenta para escribir la tesis. A Agustin Z. por esos ánimos y los diferentes consejos. A Ricardo M. por las largas conversaciones a la hora de comer y siempre estar dispuesto a ayudar. A Leticia A. por su grandísima profesionalidad y paciencia con nosotros y en especial, conmigo. A Álvaro H. por ayudar con los *SO* y las diferentes credenciales. A Kepa Z. por atenderme siempre con una sonrisa y ayudar en todo lo que se podía. A Sara B.L. por preguntar siempre por mí en el chat. Gracias a todos.

También tengo que agradecer enormemente a Hermann Ney su amabilidad, paciencia y hospitalidad durante mi estancia en Aachen. Aprendí muchas cosas sobre *Machine Translation* y gestión de personas. A Jan-Thorsten Peter (master beer opener), Stephani J., Parnia Bahar, Weiyue W., Tobias M., Stephan P., Dhenya S., Dewi Suryani y al resto de integrantes del grupo de *Human Language Technology and Pattern Recognition* del i6 de RWTH de Aachen por su grandísima amabilidad, paciencia y por toda la ayuda que me dieron. *Vielen Dank für alles.*

Ahora lo mismo, pero en inglés.

In addition, I have to give enormous thanks to Prof. Dr.-Ing. Hermann Ney for him kindness, patience and hospitality during my stay in Aachen. I learned a lot about Machine Translation and people

management. Also, I thank to Jan-Thorsten Peter (master beer opener), Stephani J., Parnia Bahar, Weiyue W., Tobias M., Stephan P., Dhenya S., Dewi Suryani and the other members of the *Human Language Technology and Pattern Recognition* group of i6 in RWTH Aachen for they great kindness, patience and for all the help that they gave me. Vielen Dank für alles.

A mi madre, mi abuela y a mi abuelo por siempre estar ahí. Me habéis apoyado y ayudado tanto que necesitaría un tiempo infinito para daros las gracias. Somos un desastre monumental, pero aun así, conseguimos salir adelante.

A la *Cuadrilla*, Aitzi, Eloy, Esdras, Itxas, Jorge, Kapi, Les, Liher, Marta G., Marta M., Mono, Peter y Xabi. Sois la familia que he elegido. Gracias por vuestra amistad, apoyo y ánimos.

A Marian, Asier y Mikel. Todavía no me doy cuenta de la suerte que tengo de que estéis en mi vida y de poder contar con vosotros. Gracias por todo.

A Sara, Dory, Dorita, Toñi y Miguel gracias por vuestro apoyo continuo y los consejos. Os aseguro que no han caído todos en saco roto.

Y a la persona con la que comparto mi camino y el resto de él. Raquel, infinitas gracias por apoyarme, aconsejarme y nunca perder la esperanza en mí. Tú has hecho que esta etapa no haya sido tan difícil. Gracias preciosa.

Gracias a todos de todo corazón.

Índice general

Índice de figuras

Índice de tablas

"¿Cómo se supone que la educación me va a hacer más listo? Al contrario, cada vez que aprendo algo nuevo, algo que ya sabía desaparece de mi cerebro. ¿Recuerdas cuando hice ese curso de fabricación de vino en casa y se me olvidó conducir?"

Homer Jay Simpson
(1956 –)

1

Introducción

Rara es la palabra, esa herramienta que puede hacer que una persona se convierta en leyenda o que un imperio caiga.

Desde el principio de los tiempos, la comunicación mediante la palabra entre seres es un síntoma de inteligencia, socialización y supervivencia. Gracias a esta potentísima herramienta, los seres humanos hemos logrado llegar a transmitir nuestras vivencias, secretos, descubrimientos y demás sucesos que han acaecido nuestra vida. Ahora mismo, mediante la escritura de esta tesis, yo estoy compartiendo con el mundo los conocimientos, experiencias, investigaciones y resultados que han copado una época de mi vida. Y todo mediante la palabra, en este caso, escrita.

Hoy en día mediante diferentes tipos de canales, todo el mundo escribe, comparte u opina sobre todos los aspectos que rodean nuestra sociedad cada vez más tecnologizada. Se utilizan expresiones del lenguaje hablado, se modifican expresiones ya existentes o, simplemente, se crean nuevas expresiones o formas de hablar dentro de ciertos círculos sociales. La riqueza del lenguaje es tal, que se adapta a cualquier circunstancia, entorno y lugar generando muchísima información, tanto de nosotros mismos como del entorno que nos rodea.

Esta fuente de información y opiniones se convierte en un gran conjunto de conocimiento, desestructurado en ciertos aspectos, pero con un grandísimo potencial para comprender el mundo en el que vivimos. Este basto saber que se genera cada día, muestra la importancia de la comunicación. En ella mostramos nuestras necesidades, esperanzas, amores, hábitos, éticas, ideologías, rencores y demás sentimientos, para comunicarlos a otra persona que conozcamos. Toda esta información o conteni-

do generado, sirve a todas las áreas relacionadas con la sociedad para establecer su posición en esta pero, en concreto, en el área empresarial, puede ayudar a mejorar el posicionamiento de una empresa o producto. Con la ayuda del canal de comunicación digital más utilizado del mundo, Internet, las empresas promocionan sus productos y servicios. Sin embargo, en muchas ocasiones las empresas recurren a modelos de publicidad tradicionales y unidireccionales, más propios de otros medios como la televisión o la radio. La adaptación de las empresas, tanto a nivel de modelo de negocio como de estrategia de marketing, a las nuevas características de esta red de comunicación es fundamental de cara a asegurar el éxito.

El conocimiento de lo que la gente piensa, tanto sobre una empresa, un producto, una persona o cualquier aspecto que sea imagen de algún ente social, puede propiciar el éxito o el fracaso de las acciones que quiera llevar a cabo dicha entidad. Actualmente, el flujo para comunicarse de las personas ha evolucionado hasta llegar a utilizar las nuevas tecnologías y, dentro de ellas, nuevos sistemas de mensajería instantánea para realizar estas comunicaciones. El análisis de estas comunicaciones permite obtener datos relevantes sobre las opiniones de los usuarios, pero el problema es que esta desestructurada, con múltiples coloquialismos, faltas de ortografía, modismos y referencias a elementos que están dentro de un contexto dado. La comprensión de esta información no es una tarea simple. Aunque actualmente existan herramientas que ayudan a un sistema informático a comprender la estructura de las palabras y las frases, no suelen ser tan flexibles ni abstractas para comprender los contextos si contienen poco texto y, encima, no formal.

Empresas como Google, utilizan principalmente las palabras para ofrecer anuncios. Esta táctica les funciona a las mil maravillas, pero esto se debe a que tienen millones de palabras y elementos etiquetados. La cantidad, en este caso, si marca una diferencia considerable. Además, estos recursos no siempre están disponibles para todas las empresas, por lo que buscar métodos que mejoren el alcance de sus ventas, teniendo en cuenta las limitaciones que tienen, es vital.

Además, el procesamiento de esta información desestructurada, de poco y corto contenido, con expresiones específicas de ciertos lenguajes producidos por avances tecnológicos puede ayudar, en este caso, a las pequeñas y medianas empresas que quieran vender productos específicos a usuarios que realmente quieren dichos productos, sin la necesidad de molestarles mediante sistemas de SPAM o por bombardeo de anuncios que no les agradan, utilizando recursos gratuitos y los propias fuentes de conversaciones de las redes sociales o de sus propias plataformas digitales. Por otro lado, también puede servir para conocer su posición en la sociedad y la visión que tienen los ciudadanos sobre ella.

Teniendo en cuenta todo lo mencionado, para realizar este trabajo, la principal motivación que hemos tenido ha sido querer terminar de asestar la última estocada al

SPAM masivo e indiscriminado. Esta lacra se está expandiendo por todos los sistemas de mensajería instantánea y es bastante molesto. El problema es que este sistema, debido a su bajo coste y alto impacto, es difícil de terminar con él. Por eso hemos pensado que, si al menos proporcionando herramientas que mejoren el alcance de las campañas de marketing mediante la generación de sistemas de metodologías y recursos que, a priori sean más eficaces que el SPAM, las empresas dejen de utilizarlo, pasando al marketing personalizado. Para lograr nuestro objetivo principal, hemos planteado varios objetivos secundarios.

1. Plantear un mecanismo para poder obtener el contexto de las frases y otro mecanismo para detectar los productos relacionados con ese contexto.
2. Que los mecanismos generados sigan una metodología para poder replicarse.
3. Plantear diferentes caminos a seguir, viendo la viabilidad de cada uno de ellos para futuros experimentos.
4. Ver si alguno de los planteamientos viables puede tener un enfoque comercial real.

En definitiva, esta investigación se centra en el análisis de textos desestructurados y con todo tipo de sublenguajes tecnológicos obtenidos de conversaciones, teniendo en cuenta la privacidad de los usuarios, para lograr obtener el tema u objetivo de la conversación y, en este caso, ofrecer un producto que puede estar buscado y que sea adecuado a sus gustos.

Con lo previamente mencionado, se ha propuesto la siguiente hipótesis. *¿Es posible detectar el contexto de frases cortas para ofrecer productos, ofertas, o cualquier elemento referente al marketing o área de negocio de una entidad mediante la utilización de recursos lingüísticos, análisis de las frases, el ámbito de uso de los productos y elementos relacionados con contextos determinados y que todo este proceso siga una metodología y pueda adaptarse a las nuevas modas lingüísticas y plataformas como las redes sociales o sistemas de mensajería instantánea?*

"Mira, pueden aceptar la ciencia y enfrentar la realidad, o pueden creer en ángeles y vivir en un mundo infantil de fantasías"

Lisa Marie Simpson
(1980 –)

2

Estado del Arte

Al comienzo de toda gran aventura, lo primero es explicar el entorno en el cual se va a desarrollar la historia. En los siguientes apartados, se darán pinceladas de cómo han ido creciendo y evolucionando las áreas en las que toma parte esta historia.

2.1 Procesamiento del Lenguaje Natural

El lenguaje es la herramienta y/o arma más importante y potente de que dispone la humanidad.

A simple vista, puede no parecer una herramienta con tanto peso como un sistema informático, o tan mortífera y destructiva como las armas nucleares, pero para la creación de ambas es necesario el lenguaje.

El lenguaje es utilizado para transmitir ideas y conocimientos, lo cual lo convierte en una herramienta de comunicación. Esta información puede ser almacenada y procesada en sistemas informáticos para su posterior análisis. Y es precisamente eso, la posibilidad de trasmitir y almacenar conocimiento, lo que convierte al lenguaje en una herramienta tan potente que ayuda al avance de la sociedad.

Por otra parte, en innumerables etapas de la historia, múltiples personalidades han utilizado el lenguaje para transmitir sus ideales o maneras de comprender la realidad. El resultado de esta transmisión de mensajes, ha desembocado en grandes

movimientos con resultados impresionantes, tanto buenos como malos, repercutiendo en el curso de la historia y en el futuro de la humanidad.

En la actualidad, las personas utilizan nuevos canales, tales como Internet o redes sociales, para comunicarse o transmitir sus ideas o conocimientos. Debido a este nuevo paradigma y a la necesidad de poder procesar y buscar conocimiento dentro de toda esa nueva información, nace una nueva área dentro de las ciencias computacionales: el procesamiento del lenguaje natural (PLN).

Pero para llegar a comprender qué es y como funciona esta área, primero es importante definir ciertos aspectos del propio lenguaje y su uso.

2.1.1 Introducción al estado del Arte del PLN

La capacidad de comunicarse es un rasgo que otorga inteligencia, prosperidad, seguridad y supervivencia al ser vivo que lo logra realizar de manera continuada en el tiempo. La mayoría de los seres vivos que habitan nuestro planeta, poseen ciertos mecanismos para la comunicación, lo que les otorga cierta ventaja a la hora de sobrevivir.

El proceso para lograr la comunicación con seres de la misma especie es largo y necesita de muchos años de evolución. Nosotros, los Seres Humanos, hemos requerido de millones de años para lograr comunicarnos de manera ordenada, mediante un formato especifico y unas reglas determinadas para que cualquier individuo pueda entender el mensaje que queremos trasmitir, siempre y cuando conozca dichas reglas. Pero aun conociendo las reglas y el formato, no siempre somos capaces de entender al emisor del mensaje.

Esto se debe a que la propia naturaleza de ser del emisor, junto con su manera de ver la realidad que está transmitiendo, es diferente a la de los posibles receptores del mensaje. Además, el emisor podría utilizar ciertas palabras que, si bien a él le resultan más familiares o sencillas, a otros posibles receptores les resulten desconocidas o difícil de comprender. Esto se debe a que el lenguaje natural está en continua evolución, generando nuevas palabras que en ocasiones son sinónimas o antónimas a otras ya creadas o nuevos conceptos para nuevas realidades que van surgiendo.

Por si este problema no fuera suficiente, las personas pueden mantener conversaciones durante largos periodos de tiempo, con interrupciones, teniendo como referencia un tema principal y a su vez, subtemas recurrentes dentro o fuera del mismo tema principal. A este fenómeno se le llama contexto. El contexto es el conjunto de circunstancias (materiales o abstractas) que se producen alrededor de un hecho o evento dado. Esto implica que, si ya pueden existir problemas por la mera utilización de las propias palabras de un lenguaje natural, la suma de no conocer el contexto en el cual se desarrolla la comunicación puede llegar a resultar una tediosa y desor-

denada unión de palabras, con significados y con referencias a elementos anteriores completamente incomprensible.

El Ser Humano ha desarrollado, además del lenguaje, herramientas naturales que le ayudan en su utilización y que sirven para minimizar los problemas anteriormente mencionados. Estas herramientas comprenden la capacidad de asociar de manera lógica o intuitiva frases o palabras a hechos. Esta táctica ayuda enormemente a la hora de comprender una conversación sin el conocimiento del contexto previo o de las nuevas palabras desconocidas.

El proceso es simple. Mediante la escucha de nuevos mensajes dentro de la misma conversación, estas herramientas sirven para relacionar los nuevos mensajes que se van procesando con mensajes anteriores y con ideas, hechos, conceptos, realidades, personalidades o referencias que el propio individuo conoce previamente. De esta manera, puede darle el contexto a la conversación sin necesidad de haber participado en ella desde su comienzo.

Por la parte que atañe a las palabras nuevas o desconocidas, el proceso es semejante. Se obtiene el mensaje en el cual está la palabra desconocida y, mediante el uso y significado que el emisor da a las palabras que lo rodean junto con los mensajes anteriores y posteriores, es posible que el individuo logre crearse una idea de cuál es el posible o posibles significados de esa nueva palabra.

Para el desarrollo de estas herramientas naturales, la evolución ha sido de millones de años. Los seres humanos hemos logrado tener un sistema de comunicación, junto con unas herramientas naturales, lo suficientemente robusto para poder lograr comunicarnos con facilidad, pudiendo entender y comprender nuevos términos gracias a la vinculación lógica o el contexto.

Pero hoy en día, la mayoría de las comunicaciones no siguen los canales que antaño utilizaban. Actualmente, gran cantidad de las comunicaciones las hacemos por medios digitales, principalmente Internet. Esto hace de Internet un lugar repleto de información sobre casi todo lo que acaece en la sociedad. Aunque esta situación parezca idílica, también tiene un pero, y es que nuestra información queda almacenada en lugares que desconocemos y que terceras entidades pueden utilizar para analizarnos, estudiarnos, predecirnos o, incluso, alterar nuestras conductas.

En la última década, junto con el aumento del uso de dispositivos móviles, la cantidad de información que se almacena en "*la nube*", siendo este concepto servidores de empresas o entidades que nos ofrecen servicios deslocalizados, ha aumentado de manera vertiginosa. Para hacernos una idea, desde la primera interconexión de dos computadoras, el 21 de noviembre de 1969 entre la universidad de Stanford y UCLA, hasta el año 2003, año de nacimiento de las primeras redes sociales digitales como MySpace, Delicious, Linkedin y Facebook, el volumen que las Webs tenían era de una docena de exabytes de información. Hoy en día, esa misma cantidad de

información se genera semanalmente. Esto se debe al auge de las redes sociales y el afán por compartir todo (ideas, opiniones, imágenes, criticas, etc.) de los usuarios que las pueblan.

Teniendo esto en cuenta, la célebre cita "*la información es poder*" toma un nuevo sentido. "*El procesamiento de la información localizada en Internet, es poder*".

La actualización de esta frase es debido a que, a partir de esta información almacenada en diferentes fuentes de Internet, es posible generar diferentes perfiles de usuarios concretos, como han demostrado Yu et al. (2006), Golemati et al. (2007), Amato y Straccia (1999), Sugiyama et al. (2004) o Khemmarat et al. (2014) en sus respectivos trabajos.

Estos perfiles abarcan desde la capacidad monetaria de un individuo como muestra Ellison et al. (2014), hasta la detección de acosadores, como nos enseña Galán-García et al. (2014). En las redes sociales y demás lugares de la Web, los usuarios hacen nuevos amigos, publican sus estados de ánimo, sus gustos, qué famosos les gustan, qué tipo de visión tienen acerca de un evento concreto, ideas políticas, opiniones sobre productos que han comprado y sobre las marcas que comercializan dichos productos. Además, en la misma línea, y en base a la misma tecnología, Davis et al. (2013) y Kumar y Singh (2013) nos muestran que es posible identificar conductas relacionadas con algún tipo de cibercrimen, como apología de comportamientos radicales, grupos terroristas, etc.

Pero para procesar toda esta nueva información, es necesario utilizar técnicas que detecten, analicen y comprendan qué es exactamente lo que el individuo ha publicado. El análisis manual por parte de un Ser Humano es prácticamente imposible debido a la gran cantidad de información generada, por lo que es necesario el uso de sistemas informáticos para dicha tarea.

El principal inconveniente con el que se encuentran estos sistemas es que buscan el significado literal de las palabras en un determinado idioma y, como antes se ha comentado, se da la circunstancia de que, el lenguaje humano y sus palabras o expresiones, tienen en unas ocasiones un significado distinto al que se le da en otras, dependiendo del *contexto*. A este fenomeno se le llama desambiguación y existen diferentes tecnicas para solucionarlo, tal y como nos muestran Sussna (1993), Walmsley et al. (2014) y Moro et al. (2014).

En las siguientes líneas, se mostrará una breve historia de las diferentes técnicas que se han utilizado desde el nacimiento del procesamiento del lenguaje natural, problemas que atañen a esta área de la computación y diferentes investigaciones relacionadas con el tema de esta investigación.

2.1.2 Historia del PLN

El PLN como área de investigación o tecnología, puede considerarse que comienza en 1950, aunque existían trabajos previos. En 1950, Turing (1950) publicó el artículo "*Computing Machinery and Intelligence*". En dicho ensayo, Turing imaginó una máquina capaz de comunicarse a través del lenguaje natural por medio de un texto, y propuso lo que hoy conocemos como Test de Turing como criterio de inteligencia. En 1954, Dostert (1955), que estaba dentro del grupo de Georgetown, junto con la empresa IBM realizo la demostración con la que se tradujeron del ruso al inglés más de sesenta frases. Este experimento llevó a decir a los autores que el problema de la traducción automática estaría resuelto en menos de 10 años, lo cual no se cumplió tal y como lo demostró el informe de ALPAC, realizado por Pierce y Carroll (1966). Después de esto, solo se llevaron a cabo unas pocas investigaciones menores en traducción automática hasta finales de 1980. Algunos desarrollos en la década de 1960 fueron, el sistema "SHRDLU" realizado por Winograd (1980) que permitía dar ordenes mediante el lenguaje para que una maquina pudiera mover bloques de colores y formas. La famosa ELIZA, un bot conversacional desarrollado por Weizenbaum (1966), profesor del *Massachusetts Institute of Technology* (MIT). Dicho programa fue el primer sistema de Inteligencia Artificial en aplicar el concepto de reconocimiento de patrones de estimulo-respuesta a la comprensión del lenguaje natural. ELIZA también fue el primer bot que almacenaba las conversaciones para mejorar el sistema y la base para numerosos bot conversacionales que vendrían después. Durante la década de 1970 se empezaron a escribir "ontologías conceptuales" para intentar generar recursos que pudieran ayudar a convertir información del mundo real en información estructurada que los sistemas informáticos pudieran comprender y procesar. Algunas ontologías son MARGIE desarrollada por Schank et al. (1973), SAM desarrollada por Cullingford (1978), PAM desarrollada por Wilensky (1978) o QUALM desarrollada por Lehnert (1977). Hasta la década de los 80, la mayoría de los sistemas de PLN estaban basados en recursos generados de manera manual, normalmente reglas. En esta década comenzaron a utilizar *machine learning* o aprendizaje automático lo que llevo a la mejora de esta área. Algunos de los algoritmos más utilizados eran los árboles de decisión porque generaban reglas parecidas a las creadas de manera manual. El etiquetado gramatical o "*part-of-spech*" introdujo los modelos de cadenas ocultas de Markov, publicado por Baum et al. (1970). También se comenzaron a utilizar sistemas y modelos estadísticos para calcular las probabilidades y pesos de las posibles respuestas correspondientes al texto que se estaba procesando, como muestran los trabajos de Rabiner (1989) y Huang et al. (1990). La ventaja de este tipo de sistemas es que, teniendo como entrada un conjunto desconocido (algo muy común en el mundo real) producen resultados más fiables cuando se integran en sistemas más amplios y que comprenden múltiples subtareas.

El PLN, como área definida, comenzó a aparecer a finales de los 90. Más con-

cretamente en 1997 con los trabajos de Kessler et al. (1997), Spertus (1997) y en 1998 con Argamon et al. (1998). Pero es a partir de comienzos de la década del 2000 cuando se convierte mayormente en una sub-área de la disciplina de la gestión de la información como muestran los trabajos de Kobayashi et al. (2001), Rauber y Müller-Kögler (2001), Subasic y Huettner (2001), Dimitrova et al. (2002), Durbin et al. (2003), Liu et al. (2003), Riloff et al. (2003) y Hillard et al. (2003).

Las investigaciones recientes como las de Sidorov et al. (2014), Collobert y Weston (2008), Nandhini y Sheeba (2015b) y Shams (2014) se centran, cada vez más, en los algoritmos de aprendizaje no supervisados y semi-supervisados. Tales algoritmos son capaces de aprender de conjuntos de datos que no han sido etiquetados previamente o etiquetados parcialmente. La complejidad de esta tarea es mayor que la del aprendizaje supervisado y, dependiendo del tamaño de los datos de entrenamiento, normalmente suele ser menos precisa. Sin embargo, existe gran cantidad de datos sin anotar, como el contenido de las Webs de Internet y de las Webs Semánticas, que suelen ayudar a mejorar los resultados. Por ejemplo, hasta comienzos de la década del 2000, los dos enfoques más importantes para la detección de sentimientos estaban basados en técnicas de *aprendizaje automático* y *análisis semántico*. Desde entonces, sin embargo, se ha extendido el uso de las técnicas de PLN, que se verán en la sección 2.1.4. También es necesario destacar que para la detección de sentimientos existen numerosos problemas. Los enfoques que se suelen utilizar para detectarlos y tratarlos son: la clasificación de la subjetividad, la clasificación binaria sentimental (positivo-negativo) y la clasificación porcentual sentimental (porcentaje de positivo o negativo) tal y como se muestra en los trabajos Wiebe (1994a), Koppel y Schler (2006a) y Koppel y Schler (2005a).

2.1.3 Etapas del PLN

El PLN posee unas etapas numeradas y detalladas para llevar a cabo su tarea. Estas tareas son *(i)* análisis morfológico, *(ii)* análisis sintáctico, *(iii)* análisis semántico y *(iv)* análisis pragmático. A continuación entraremos en más detalle sobre cada una de ellas.

El *análisis morfológico* para Anglin y Miller (2000) consiste en determinar la forma, clase o categoría gramatical de cada palabra de una oración y detectar la relación que se establece entre las unidades mínimas que la forman.

Como dice el trabajo de Ballmer y Brennenstuhl (1980), este nivel de análisis está estrechamente relacionado con el análisis léxico. Las palabras que forman parte del diccionario están representadas por una entrada léxica y en caso de que ésta tenga más de un significado, o diferentes categorías gramaticales, tendrá asignadas diferentes entradas. El análisis léxico incluye la información morfológica, la categoría

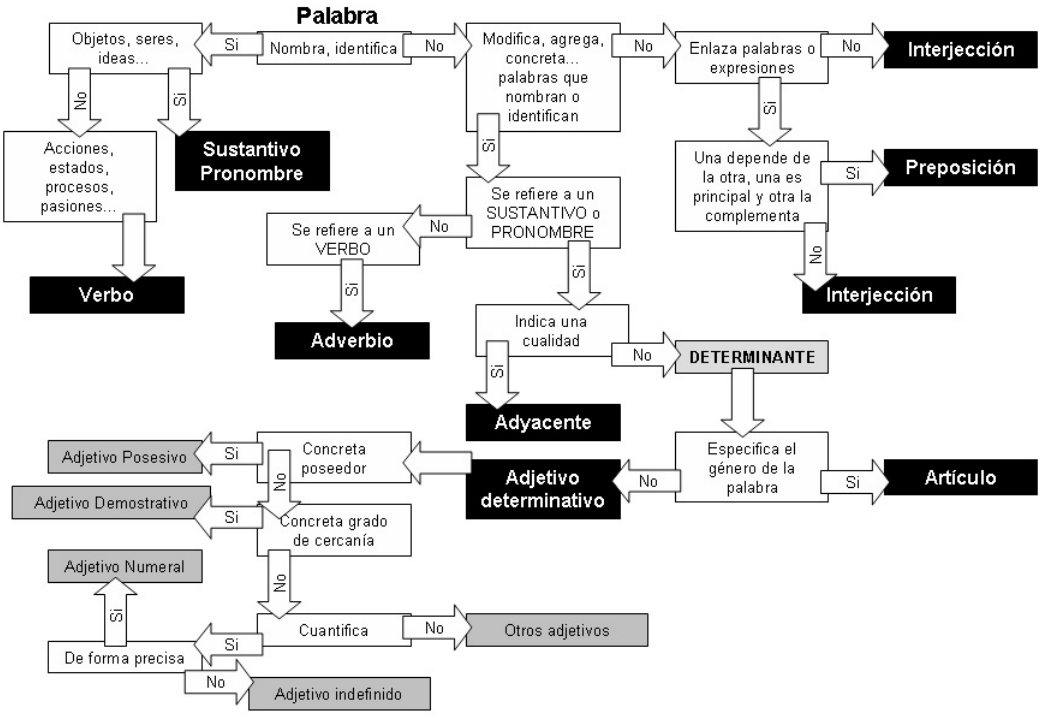


Figura 2.1: Diagrama de flujo del criterio Semántico para categorizar palabras.

gramatical, irregularidades sintácticas y representación del significado. Normalmente, el análisis léxico solo contiene la raíz de las palabras con formas regulares, siendo el analizador morfológico el encargado de determinar si el género, número o flexión que componen el resto de la palabra son adecuados.

Para Salton y Smith (1989), el *análisis sintáctico* consiste en etiquetar cada uno de los componentes sintácticos que aparecen en la oración y analizar cómo se combinan las palabras para formar construcciones gramaticales correctas. El diagrama de flujo para generar construcciones gramaticales correctas puede verse en la figura 2.1.

El resultado de esta etiquetación es generar una estructura que corresponda a las categorías sintácticas formadas por cada una de las unidades léxicas que aparecen en la oración. Las gramáticas están formadas por conjuntos de reglas, como por ejemplo:

- SN = Sintagma nominal
- SV = Sintagma verbal
- SP = Sintagma preposicional o Preposición
- Det = Determinante

Para Goddard (1998), el *análisis semántico* consiste en el análisis de las frases y su codificación para descubrir el significado que esconden.

En los últimos años las técnicas de procesamiento sintáctico han experimentado avances significativos, resolviendo algunos de los problemas fundamentales. Sin embargo, las técnicas de representación del significado no han obtenido los resultados deseados y numerosas cuestiones continúan sin encontrar soluciones satisfactorias. La definición del significado de una frase es una ardua tarea, ya que puede dar lugar a diversas interpretaciones o tener diferentes significados. A efectos prácticos, una buena segmentación del texto y una buena obtención de las partes independientes y dependientes del contexto general ayuda a comprender mejor el significado general.

El significado independiente del contexto, tratado por la semántica, hace referencia al propio significado que las palabras tienen, sin considerar el significado adquirido o el que las dé el resto del texto que las rodea. Este enfoque hace referencia al significado real de las palabras. Tanto la influencia del resto del texto como las que el hablante quiere darle, son ignoradas.

Por otra parte, el significado dependiente del contexto, estudiado desde la pragmática, que es el campo en el que se estudia la influencia del contexto en cada palabra, se refiere al componente significativo de una frase asociado a las circunstancias en que ésta se utiliza.

Atendiendo al desarrollo en el proceso de interpretación semántica, es posible optar entre múltiples pautas para su organización. Por ejemplo, en referencia a la estructura semántica que se va a generar, puede interesarnos que exista una simetría respecto a la estructura sintáctica, de tal manera que se genere una estructura arbórea para el análisis semántico, que tendrá las mismas características que el árbol sintáctico o, por el contrario, que no se dé tal correspondencia entre ellas, caso en el que se realizarán varias transformaciones sobre la estructura utilizada en la sintaxis generándose la representación semántica sobre dichas transformaciones.

Cada una de las opciones anteriores, puede implementarse de forma secuencial, comenzando por realizar el análisis sintáctico y continuando con el análisis semántico. También se puede hacer de forma paralela, iniciando el análisis semántico de cada constituyente cuando éste ha sido tratado por el analizador sintáctico. Finalmente, en combinación con cada una de las opciones anteriores, podemos escoger un modelo en el que exista una correspondencia entre reglas sintácticas y semánticas o, contrariamente, podemos optar por un modelo que no cumpla tal requisito. En caso afirmativo, para cada regla sintáctica existirá una regla semántica correspondiente. El significado es representado por formalismos conocidos por el nombre de *representación del conocimiento*, en inglés *knowledge representation*.

El léxico proporciona el componente semántico de cada palabra en un formalismo concreto, y el analizador semántico lo procesa para obtener una representación

del significado de la frase.

Por último, para Wilson (1990), el *análisis pragmático* añade información, extra o adicional, al análisis del significado de la frase en función del contexto donde aparece.

Se trata de uno de los niveles de análisis más complejos, cuya finalidad es incorporar al análisis semántico la aportación significativa que pueden hacer los participantes, la evolución del discurso o la información presupuesta. Éste incorpora así mismo información sobre las relaciones que se dan entre los hechos que forman el contexto y entre diferentes entidades.

2.1.4 Aplicaciones y técnicas

Clasificación de la subjetividad. La subjetividad, según Wiebe (1994b), referida al lenguaje natural se refiere a los aspectos del lenguaje, y valga la redundancia, a aquéllos utilizados para expresar opiniones y evaluaciones. La subjetividad se representa de la siguiente forma: dado un conjunto de frases $S = \{ S1, \ldots, Sn \}$ en un documento D, el problema de la clasificación de la subjetividad radica en distinguir las frases que son utilizadas para representar opiniones Ss de las que se utilizan para mostrar información de forma objetiva $S0$, cuando $Ss \cup S0 = S$. Esta tarea es especialmente relevante para la transmisión de noticias y para los foros de Internet, en los se expresa la opinión de varios agentes.

Clasificación de sentimientos. Dentro de la clasificación de sentimientos se han definido dos tipologías: por una parte la clasificación binaria y, por otra, la clasificación múltiple o multi-clase. Estas clasificaciones consisten en que, dado un conjunto de documentos $D = d1, ..., d2$ y una serie de categorías predefinidas $C = positivo, negativo$, se asigna a cada documento d_i una etiqueta definida en C. Si el conjunto C tiene sólo dos elementos, se trata de una clasificación binaria, si fuera mayor, pasaría a ser una clasificación multi-clase.

Hasta ahora, todo el trabajo realizado en relación con la identificación de opiniones se ha enfocado en la discriminación de opiniones positivas y negativas *(Polaridad)*. Sin embargo, podría resultar interesante tener más información que la que otorga este tipo de distinción binaria, especialmente cuando se está haciendo una categoría o clasificación de objetos en base a recomendaciones u opiniones. Trabajos como los de Koppel y Schler (2005b) y Koppel y Schler (2006b) demuestran lo importante que resulta utilizar ejemplos neutros a la hora de discernir entre opiniones polarizadas. Si únicamente se intenta aprender a partir de ejemplos positivos y negativos, no se obtendrán resultados satisfactorios cuando haya que clasificar ejemplos neutros. De hecho, el entrenar los sistemas de detección con ejemplos neutros,

permite una mejor distinción entre opiniones positivas y negativas.

Aplicaciones de la detección de sentimientos. Existen multitud de aplicaciones potenciales de la detección de sentimientos, de las cuales se presentan a continuación algunos de los ejemplos más relevantes.

- **Subcomponente tecnológico.** El análisis de sentimientos y el sondeo de opiniones según Tatemura (2000), tienen la posibilidad de adaptarse como facilitadores tecnológicos dentro de otros sistemas, como sistemas de recomendación aumentada según Terveen et al. (1997), para la detección de provocaciones, en inglés *flames*, o disputas en diversos canales de comunicación según Spertus (1997), o para evitar poner anuncios en páginas web que perjudiquen a la empresa anunciadora debido al contenido inadecuado de dicha página como muestra Jin et al. (2007).

- **Inteligencia empresarial.** Las aplicaciones inteligentes (e.g. las soluciones de inteligencia empresarial, en inglés *business intelligence* (BI)) conforman uno de los campos a los que mejor se adapta el análisis de opiniones. Actualmente resulta complicado entender por qué determinados productos se están vendiendo mal, por ejemplo. Así, un sistema que busque y analice críticas y opiniones vertidas en diferentes puntos de la Web, y desarrolle un resumen consolidado de aquellos puntos en los que existe un consenso de ideas, constituye automáticamente una asistencia extremadamente valiosa en la toma de decisiones de negocio como muestra Lee (2003). Además, según Mishne y Glance (2006), se pueden realizar análisis de mercado y análisis de marca en base al seguimiento de los juicios vertidos por usuarios y clientes.

- **Dominios sociales.** Como es evidente, la minería de opiniones e ideas pasa a ser una herramienta de gran interés en diferentes dominios sociales. Por ejemplo, en la política ya existen trabajos como los de Efron (2004), Hopkins y King (2007), Laver et al. (2003), Cardie et al. (2006), Shulman et al. (2005) y Conrad y Schilder (2007) orientados a tratar de entender lo que los votantes están pensando, a mejorar la calidad de la información a la que tienen acceso dichos votantes, a analizar la opinión del electorado respecto a ciertas reformas legislativas, etc. Además de en relación a la percepción que se consigue en relación a ciertas materias legales. En otros campos, como la sociología, también está alcanzando un nivel importante a la hora de estudiar cómo se difunden innovaciones o ideas, definiéndose diferentes corrientes de influencia según van apareciendo y desapareciendo *líderes de opinión* y modificando la *receptividad* del público a lo largo del tiempo como se muestra en el trabajo de Rogers (1995).

- **Enfoque de similitud.** Yu y Hatzivassiloglou (2003) exploraban la hipótesis de que, dado un tema concreto, las frases de opinión tendrán mayor similitud con frases que expresen opinión que con frases que indiquen hechos. La similitud se mide en función de palabras compartidas, expresiones o sintagmas descritos en los trabajos de Dagan et al. (1993), Miller y Charles (1991), Resnik (1995) y Zhang et al. (2002). El método consiste en tres pasos: (1) en primer lugar se obtienen documentos del mismo tema que la frase en cuestión; (2) después se calcula la similitud respecto a cada frase de dichos documentos y se obtiene el valor promedio; (3) finalmente se asigna la frase estudiada a la categoría (opinión o hecho) dependiendo del valor promedio más alto.

- **Clasificador Bayesiano.** En este enfoque, se presuponen todas las frases de un artículo de opinión o información, como opiniones o hechos respectivamente. Las frases en documentos de opinión o informativos se usan como ejemplos de las dos categorías. Las características incluyen palabras, *bigramas* y *trigramas*, así como la parte del discurso de cada frase. Además, la presencia en la frase de palabras orientadas semánticamente (positivas o negativas) es un indicador de que la frase es subjetiva. Por tanto, se pueden incluir los contadores de palabras positivas o negativas en la frase, así como contadores de polaridad de secuencias de palabras orientadas semánticamente. También se incluyen contadores de partes del discurso combinadas con información de polaridad así como características que codifican la polaridad del verbo principal, el sujeto y sus modificadores inmediatos.

- **Multi-clasificador Bayesiano.** La hipótesis del clasificador Bayesiano del punto anterior es una aproximación. Para hacer frente al problema general se aplica un algoritmo que hace uso de múltiples clasificadores, cada cual enfocado a un conjunto diferente de características. El objetivo es reducir el conjunto de entrenamiento a las frases que *probablemente* estén mejor etiquetadas, mejorando de esta manera la precisión en la clasificación. Existen diferentes conjuntos de características que entrenan diferentes clasificadores bayesianos respectivamente. Si se asume como verdad que la información provista por las etiquetas del documento y que todas las frases heredan el estado de su documento como opiniones o hechos, en primer lugar se entrena el primer clasificador con todo el conjunto de entrenamiento y se utiliza el clasificador correspondiente para predecir las etiquetas de dicho conjunto. Las frases que obtienen una etiqueta diferente a la asumida en un principio se eliminan, y se pasan al siguiente clasificador las frases restantes. Este proceso se sigue repitiendo hasta que se dejen de eliminar frases. Yu y Hatzivassiloglou (2003) muestran los resultados utilizando cinco conjuntos de características, empezando por palabras, y añadiendo *bigramas*, *trigramas*, parte del discurso

y polaridad.

- **Clasificador en base a partes.** El enfoque, según Pang y Lee (2004), propone la hipótesis de que fragmentos de texto (objetos) cercanos entre sí (dentro de los límites del discurso) pueden compartir el mismo grado de subjetividad. De esta manera, Pang incorpora a su algoritmo la interacción de pares de información, esto es, especificando que dos frases concretas deberían recibir idealmente la misma designación de subjetividad. Dicho algoritmo utiliza una eficiente e intuitiva formulación basada en grafos confiando en buscar los cortes mínimos.

2.1.5 Clasificación de opinión mediante palabras

Generalmente, este tipo de clasificación requiere la construcción manual o semimanual de diccionarios de palabras sobre los que construir técnicas de clasificación de opiniones como bien dicen Hatzivassiloglou y McKeown (1997), Lin (1998), Pereira et al. (1993), Riloff et al. (2003) y Turney y Littman (2010). Algunos estudios como los de Turney y Littman (2010) y Andreevskaia y Bergler (2006), han demostrado que restringir características a adjetivos para la clasificación de opiniones mediante palabras mejoraría los resultados. Sin embargo, otras investigaciones como las de Tatemura (2000), Andreevskaia y Bergler (2006), Esuli y Sebastiani (2006), Takamura et al. (2005) y Turney (2002) han incidido en que la mayoría de adjetivos y adverbios, así como un pequeño conjunto de nombres y verbos, tiene orientación semántica. Los métodos automáticos de anotación de opiniones a nivel de palabra se agrupan en dos categorías principales: (1) el enfoque basado en *corpus* como los de Yu y Hatzivassiloglou (2003), Hatzivassiloglou y McKeown (1997) y Turney y Littman (2010), que incluye métodos que confían en patrones de palabras sintácticos o co-ocurrentes dentro de grandes textos para determinar su opinión; y (2) enfoques basados en diccionario, que usan redes de palabras, especialmente sintagmas y jerarquías como el trabajo de Kim y Hovy (2004), con el fin de obtener palabras que reflejen opinión, o para medir la similitud entre palabras candidatas y palabras que implican opinión (e.g. *bueno* o *malo*) como el trabajo de Esuli y Sebastiani (2006).

Análisis de conjunciones entre adjetivos. Este método se basa en predecir la orientación subjetiva analizando pares de adjetivos (unidos por una conjunción) que se hayan extraído de un gran conjunto de documentos sin etiquetar. La idea subyacente es que los adjetivos tendrán la misma orientación si están unidos por una conjunción copulativa *(y)*, mientras que un *pero* unirá adjetivos de sentido contrario. Para inferir la orientación de los adjetivos se lleva a cabo un algoritmo de aprendizaje supervisado que incluye los siguiente pasos: primero se obtienen todas

las conjunciones de adjetivos de un conjunto de documentos; segundo, se entrena un clasificador regresivo lineal y se clasifican los pares de adjetivos, bien determinando que tienen la misma orientación, bien diferente (de esta manera, las relaciones entre adjetivos conforman un grafo); tercero, un algoritmo de agrupación, en inglés *clustering*, divide el grafo que se ha generado en dos grupos. Así, basándose en la presunción de que los adjetivos positivos se utilizan con mayor frecuencia que los negativos, el grupo que contenga la frecuencia media más alta de términos en el conjunto de documentos, será el que tenga los términos positivos.

Análisis de relaciones léxicas. En este caso se presenta una estrategia para inferir orientación semántica a partir de las asociaciones semánticas entre palabras y frases. Se parte de la suposición de que dos palabras tienden a tener la misma orientación semántica si tienen una asociación semántica fuerte. Por tanto, se centra en el uso de relaciones léxicas definidas en redes de palabras para calcular la distancia entre adjetivos. Se puede definir un grafo de los adjetivos que están presentes en la intersección entre un conjunto de términos y una red de palabras, añadiendo un enlace entre dos adjetivos siempre y cuando la red de palabras indique que hay una relación de sinonimia entre ellos, y definiendo una medida de distancia a partir de nociones elementales de teoría de grafos.

Análisis por glosas. El termino glosa se refiere a la explicación o comentario de una palabra, frase o texto difícil de entender. Según Esuli y Sebastiani (2006), la característica del método de análisis por glosas subyace en basarse en las definiciones textuales de un término en los diccionarios o glosarios *online*, para extraer conclusiones. Así, se asume que si la palabra está orientada semánticamente en una dirección, las palabras que aparezcan en su definición tenderán a estar orientadas en esa misma dirección. El proceso de clasificación se compone de los siguientes pasos: (1) se toma como entrada un par de conjuntos semilla con las categorías positivas y negativas; (2) dichos conjuntos se enriquecen con nuevos términos haciendo uso de relaciones léxicas (como la sinonimia); (3) por cada término en cada uno de los dos conjuntos, o en el conjunto de términos a clasificar, se obtiene una representación textual cotejando todas las definiciones de dicho término que se obtengan de diccionarios (legibles por máquinas), transformando cada representación en un vector por técnicas estándar de indexado de texto; (4) por último, se entrena un conjunto binario con los términos de los conjuntos positivo y negativo, para después aplicarse sobre los términos del conjunto de prueba.

Análisis conjunto de relaciones léxicas y glosas. Este método determina la *sensibilidad* de palabras y frases tanto por relaciones léxicas (sinonimia, antonimia

e hiponimia) como por definiciones textuales provistas por redes de palabras. Andreevskaia y Bergler (2006) proponen un algoritmo denominado STEP (*Semantic Tag Extraction Program*). Dicho algoritmo comienza con un reducido conjunto de palabras semilla cuyo valor sensible (positivo o negativo) ya es conocido. A continuación, se incrementa dicho conjunto añadiendo sinónimos, antónimos o hipónimos. A través de las definiciones textuales de la red de palabras, se identifican las entradas que contienen en su definición palabras que conllevan sentimientos y se añaden a su categoría correspondiente (positivo, negativo o neutro). Por último se eliminan los errores que se hayan introducido en el primer paso y se filtran aquellas palabras que se hayan definido como contradictorias. En dicho algoritmo, para cada palabra hace falta computar una métrica de superposición de red (*Net Overlap Score*), restando el número total de veces que se le ha asignado a una palabra una sensibilidad negativa de aquellas veces en las que se le ha atribuido un sentido positivo. Estos valores se normalizan en un intervalo estándar $[0, 1]$, siendo 0 la ausencia de pertenencia a la categoría de sentimiento (en este caso se trata de una palabra neutra) y 1 que refleja el grado máximo de pertenencia a dicha categoría.

Análisis de información mutua punto a punto. Según Turney (2002), la estrategia general de este método es inferir orientación semántica a partir de asociaciones semánticas. La asunción subyacente es que una frase tiene una orientación semántica positiva cuando tiene buenas asociaciones y viceversa. Dicha orientación semántica se calcula a partir de la fuerza de su asociación con un conjunto de palabras positivas menos la fuerza de la asociación a un conjunto de palabras negativas; o lo que es lo mismo, calculando el valor de la información mutua punto a punto, en inglés *pointwise mutual information* (*PMI*).

2.2 Recuperación de información

La recuperación de la información (RI), en ingles "*Information Retrieval*" (IR), es una sub-área de las Ciencias Computacionales para la búsqueda de información relevante en recursos digitales tales como documentos, bibliotecas o fuentes de Internet. Estas búsquedas pueden centrarse en diferentes elementos de los documentos como pueden ser el propio texto, los metadatos o las uniones con elementos de bases de datos relacionales.

Los primeros vestigios para el almacenamiento de información escrita datan en torno al año 3000 A.C.. Los responsables fueron los sumerios y sus sistemas para almacenar se basaban en tablas de arcilla con inscripciones cuneiformes. Su sistema de búsqueda era mediante clasificaciones especiales para identificar cada tabla con su contenido.

El almacenamiento y recuperación de información escrita a lo largo de la historia, ha sido y es una importante tarea, especialmente después de la invención del papel y la imprenta. Algunos de los primeros sistemas de IR que se crearon antes del Siglo XX fueron los de Joseph Marie Jacquard y Herman Hollerith. El del primero fue una máquina de tejer que utilizaba tarjetas perforadas para diseñar las costuras. El del segundo, fue el sistema para procesar los votos del censo Americano, tal y como explica Nieto-Nieto (2012) en su trabajo.

En la década de 1920 y 1930, Emanuel Goldberg construyó su máquina estadística. Esta máquina servía como motor de búsqueda de documentos y utilizaba células fotoeléctricas y reconocimiento de patrones para buscar los metadatos en rollos de documentos microfilmados, tal y como explica Buckland (2002) en su trabajo.

En 1945 Bush (1945) escribió el articulo *"As We May Think"* dando la idea de acceso automático a grandes colecciones de conocimiento almacenado. En la década de 1950, la idea se fue macerando mediante descripciones más concretas de cómo buscar automáticamente archivos de texto.

En 1950 se realizaron muchos trabajos que promovían la idea de realizar búsquedas básicas de texto mediante sistemas computacionales o, como los conocemos hoy en día, ordenadores. Esta idea fue en aumento y, al final de la Segunda Guerra Mundial, el ejercito de los Estados Unidos de América necesitaba encontrar una solución eficaz para indexar y recuperar documentos científicos capturados a los Alemanes durante la Segunda Guerra Mundial. Se plantearon varias soluciones como la de Luhn (1957). Él había empezado a trabajar en un sistema mecanizado de tarjetas perforadas para la búsqueda de componentes químicos, pero lo que proponía era un método que utilizaba palabras como unidad de indexación para los documentos y medir el solapamiento de las palabras como criterio para la recuperación. Con la Guerra Fría, el ejército americano tenía miedo a la huida de científicos hacia la URSS. Para minimizar la perdida de información valiosa, el gobierno americano promovió iniciativas para mejorar los sistemas mecanizados de búsqueda de contenidos, como el de Kent (1966), y la invención de la indexación de citas como el trabajo de Garfield y Merton (1979). La primera vez que apareció el termino *"Information Retrieval"* fue en 1950 y sus autores fueron Fairthorne y Mooers (1968). Un año más tarde, Philip Bagley hizo uno de los primeros experimentos de RI mediante un sistema computacional, descrito en el trabajo de Smith (1976). En 1955, Kent et al. (2007) publicó la descripción de las comúnmente empleadas medidas de "*precision*" y "*recall*" en *American Documentation*. En este documento, Allen también proponía un entorno para evaluar los sistemas de IR usando métodos estadísticos para determinar el número de documentos relevantes no recuperados.

En la década de 1960 se publicaron varios documentos claves para esta área. En 1960 Maron y Kuhns (1960) publicaron un nuevo enfoque que usaba la indexación

probabilística y el número de documentos relevantes. Este método, dada una cadena de búsqueda, mostraba una lista de documentos afines con sus probabilidades y ordenados por relevancia. Pero los trabajos más notables fueron el sistema *SMART* desarrollado por Salton (1971) y sus estudiantes de las universidad de Harvard y de Cornell, y las evaluaciones realizadas por Cleverdon (1967) y su grupo del "*College of Aeronautics*" en Cranfield. Este último sistema, fue desarrollado como una metodología para la evaluación de los sistemas RI y, todavía hoy en día, se sigue utilizando esta metodología. Por otra parte, el sistema *SMART* permitía a los investigadores desarrollar y experimentar con nuevas ideas para mejorar los sistemas de búsqueda y su calidad. Con estos dos trabajos, un entorno para experimentar y una metodología para evaluar, los investigadores pudieron ir mejorando, de manera rápida y metódica, el campo de RI permitiendo nuevas investigaciones críticas.

Muchos de los trabajos que surgieron en las décadas de 1970 y 1980, estaban basados en los avances de la década de 1960. El problema de los nuevos trabajos era que estaban planteados y probados con colecciones, o corpus, muy pequeñas de texto. El motivo se debía a que, por aquel entonces, los investigadores no disponían de acceso a grandes colecciones de texto como las que disfrutamos hoy en día. El resultado fue la imposibilidad de verificar si esas aproximaciones podrían funcionar de igual manera con grandes corpus de texto.

El tamaño de los corpus de texto era un gran problema y, para resolverlo, en 1992 varias agencias gubernamentales de Estados Unidos, bajo la tutela del "*National Institute of Standards and Technology (NIST)*" crearon la "*Text Retrieval Conference (TREC-1)*, descrita en el trabajo de Harman (1993)". TREC es una serie de conferencias de evaluación para fomentar la investigación de RI en grandes corpus de texto. Dentro de estas conferencias, se ponen a disposición de la comunidad científica grandes corpus de texto. Este nuevo escenario brinda la oportunidad de revisar, corregir y perfeccionar metodologías antiguas, además de promover nuevos enfoques. Las conferencias TREC han dividido en diferentes ramas el campo de RI. Algunas de esas ramas son; RI en fragmentos hablados, RI en diferentes idiomas, filtrado de información, interacciones de los usuarios con sistemas de RI y muchos más.

Desde 1996, los enfoques más importantes y relevantes de sistemas RI han sido los buscadores Web como Google, Bing o Yahoo!. Estos sistemas utilizan RI para obtener referencias, documentos o enlaces Web sobre lo que un usuario quiere buscar en Internet.

2.2.1 **Modificación de las consultas**

Cuando nacieron las metodologías y sistemas de RI, las personas no sabían realizar búsquedas efectivas porque eran bastante complejas. Uno de los primeros enfoques

en utilizarse para simplificar y mejorar su uso, fue la utilización de sinónimos en las cadenas de búsqueda o *query*. Este enfoque en inglés se denomina como *query modification*.

En las primeras investigaciones en RI, como por ejemplo la de Sparck Jones (1971), se utilizaron tesauros para encontrar sinónimos de las palabras de las *querys*. Sin embargo, lograr tener un buen tesauro, de propósito general, era demasiado difícil y complicado. La solución que se les ocurrió a los investigadores fue desarrollar técnicas automáticas que generaran tesauros para modificar las cadenas de búsqueda. La mayoría de estos métodos automáticos están basados en el análisis de las concurrencias de las palabras en los documentos. La solución parecía buena, pero el aumento de palabras basadas en los tesauros automáticos en las cadenas de búsqueda, limita el éxito y la eficacia en la búsqueda. La principal razón era la falta de contexto cuando se insertaban sinónimos en las *querys*. No todos los sinónimos que se insertan tienen el mismo significado dentro del contexto de la búsqueda. Por ejemplo, el término asiento es una buena alternativa para banco pero, si la *query* es "Los políticos y los bancos son unos ladrones", se modifica considerablemente la semántica original y, aunque se obtendría un mayor número de resultados a partir de la búsqueda, una parte importante de los mismos no serían válidos.

En 1965, Rocchio (1971) propuso la utilización de *feedback* relevante para la modificación de las *querys*. Su idea se basaba en que, para un usuario, es sencillo valorar la relevancia de los resultados de una búsqueda. El usuario seleccionaba los resultados óptimos y el sistema generaba automáticamente una búsqueda mejor utilizando el razonamiento que el usuario le había dado. Además, este raciocinio se almacenaba para futuras búsquedas. Este tipo de metodología ha demostrado ser bastante útil en colecciones de prueba. En la década de 1990, aparecieron nuevas técnicas que mejoraban la expansión de las *querys* sin la ayuda del *feedback* del usuario. Una de las más importantes, desarrollada por Buckley et al. (1995), es la de *pseudo-feedback*. Esta técnica es una variante de la "*relevant feedback*" y propone que, los primeros puestos de documentos recuperados por un sistema RI, sean del tema general de la búsqueda inicial y que la selección de términos relacionados con estos documentos, debería de generar nuevos términos útiles, independientemente de la relevancia del documento. *Pseudo-feedback* utiliza esta nueva información para generar una nueva cadena de búsqueda y ordenar o "*rankear*" los documentos para presentárselos al usuario. *Pseudo-feedback* ha demostrado ser una técnica muy efectiva, especialmente con *querys* cortas.

2.2.2 Otras técnicas y aplicaciones

A lo largo de los años, en el campo de RI, se han desarrollado muchas técnicas con diferentes resultados. Por ejemplo, las técnicas de agrupación. Estas técnicas se han

convertido en un área muy activa de la investigación en RI. Su hipótesis dice que los documentos que se agrupan unos cerca de otros (que son muy similares entre ellos), probablemente tengan la misma relevancia para una misma *query* inicial, tal y como se describe en el trabajo de Griffiths et al. (1997). El problema de este tipo de técnicas es que, la agrupación de documentos para mejorar la eficacia o eficiencia de las búsquedas ha sido muy limitada. Por otro lado, estas técnicas han permitido avances en RI como las interfaces de búsqueda y navegación. Una herramienta que se propuso para mejorar la eficacia de estos sistemas de RI fue el PLN. Por desgracia, los resultados tuvieron una éxito limitado, como se puede apreciar en el trabajo de Strzalkowski et al. (1996). Actualmente han mejorado los resultados, como demuestra el trabajo en ciberacoso, en inglés *cyberbullying*, de Nandhini y Sheeba (2015a). Aunque la clasificación de documentos es un enfoque muy importante de RI, no es el único. Hay otras áreas de investigación, como en el trabajo de Allan et al. (1998) que utilizan sistemas de RI para detectar y seguir temáticas, en inglés *topic detection and tracking* (TDT), el trabajo de Jones et al. (1996) para la recuperación del fragmentos hablados, o el de Pasca y Harabagiu (2001) de sistemas de pregunta-respuesta. También esta el de Belkin y Croft (1992) en filtrado de información, o el de Grefenstette (1999) para recuperar información de diferentes idiomas y lenguajes.

2.2.3 **Modelos de RI**

Los primeros sistemas RI utilizaban elementos booleanos para realizar las búsquedas. Estos elementos formaban una compleja combinación de *ANDs*, *ORs* y *NOTs*, pero no lograban obtener resultados aceptables. El motivo era que, con esos operandos, los motores de búsqueda de RI no eran capaces de recuperar la relevancia de los documentos en base a la cadena de búsqueda. Los resultados solo contenían coincidencias con características específicas tales como fechas o nombres de ficheros. Esta información era importante para recuperar ciertos tipos de documentos, pero para realizar búsquedas dentro del contenido de éstos, era inservible. Los sistemas actuales de RI utilizan la ordenación de los documentos, en base a la estimación de relevancia, de acuerdo a la *query* inicial. Casi todos los sistemas de RI otorgan un valor numérico, como ponderación o *score*, para todos los documentos resultantes de una búsqueda. Y como resultado, devuelven una lista de documentos, ordenados por ese *score*.

Para obtener este tipo de resultados, existen varios modelos que utilizan la información de los documentos para organizarlos. Los 3 modelos más utilizados son, primero los Modelos de Soporte Vectorial (MSV), "*Vector Space Models*" (VSM), segundo los modelos probabilísticos y tercero los modelos de redes de inferencia "*Inference Network Models*".

Modelo soporte vectorial

Este algoritmo representa los documentos del lenguaje natural en un modelo algebraico de vectores multidimensionales tal y como describió Salton et al. (1975). Este espacio está formado únicamente por los ejes positivos de coordenadas. Además, los documentos están representados en una matriz de términos por cada documento donde el elemento (i, j) muestra la asociación entre el termino i y el documento j. $R(q_i, e_j)$ es la función de *rankeo* que asocia un número real con una *query*.

Esta asociación refleja la aparición del termino i en el documento j. Los términos pueden representar diferentes unidades de texto como palabras o frases. Además pueden ser ponderados individualmente, permitiendo aumentar o disminuir el peso de un término dentro de un documento o de todo el corpus de documentos. Para la asignación de un *score* numérico a un documento en base a una *query*, el modelo compara la similitud entre el vector generado con dicha *query* y los vectores de cada documento.

Según Baeza-Yates et al. (1999), normalmente la representación de un documento en un sistema de RI está representada en una tupla de 4 elementos; $[\mathscr{E}, \mathscr{Q}, \mathscr{F}, R, (q_i, e_j)]$ donde:

- $\mathscr{E}$ es la representación de los documentos.
- $\mathscr{Q}$ es un conjunto que representa las *querys* del usuario.
- $\mathscr{F}$ es un entorno para el modelado de documentos, *querys* y sus relaciones.
- $R(q_i, e_j)$ es una función de *rankeo* que asocia un número real con una *query* q_i, $(q_i \in \mathscr{Q})$ y la representación de un documento e_j, $(e_j \in \mathscr{E})$. Esta función se llama función de similitud.

Si $\mathscr{E}$ es el conjunto de textos de un documento e, $e : \{t_1, t_2, ...t_n\}$, cada elemento comprende n número de t términos. Podemos considerar que $w_{i,j}$ es la ponderación del termino t_i en un documento e_j, mientras que si $w_{i,j}$ no está presente en e, entonces $w_{i,j} = 0$. Por lo tanto, un documento puede ser representado como un vector de términos indexados $\vec{e_j} = (w_{1,j}, w_{2,j}, ...w_{n,j})$. Esta formalización se aplica para el uso de MSV.

Con este modelo, es muy común el uso de "frecuencia del término – inversa de la frecuencia en el documento", en inglés "*term frequency - inverse document frequency*" $(tf - idf)$, definido por Salton y McGill (1983), para obtener la ponderación de relevancia de cada palabra. Mientras que el peso de la palabra i en el documento j determinado por $weight(i, j)$, está definido por $weight(i, j) = tf_{i,j} \cdot idf_i$. La frecuencia del termino $tf_{i,j}$ está definido por $tf_{i,j} = \frac{m_{i,j}}{\sum_k m_{k,j}}$ donde $m_{i,j}$ es el número de

veces $t_{i,j}$ que la palabra aparece en el documento e y $\sum_k m_{k,j}$ es el número total de palabras en el documento e.

En cambio, la inversa de la frecuencia en el documento idf_i está definido por $idf_i = \frac{|\mathscr{E}|}{|\mathscr{E}:t_i \in e|}$ donde $|\mathscr{E}|$ es el numero total de documentos y $|\mathscr{E} : t_i \in e|$ es el número de documentos que contienen la palabra $t_{i,j}$.

Modelos probabilísticos

Los modelos probabilísticos en RI se basan en que todos los documentos, dentro de una colección, deben estar clasificados o *rankeados* por probabilidad de relevancia del documento, de acuerdo con la *query* de consulta. Si el resultado se ha clasificado por la disminución de probabilidad, se suele llamar ordenación del principio de clasificación probabilística, o en ingles "*the probabilistic ranking principle*" (PRP), definido en el trabajo de Robertson (1977). Los sistemas no probabilísticos devuelven una probabilidad de verdades, es decir, si un documento coincide exactamente igual con la *query* de búsqueda, el documento es el resultado, en caso contrario no. Con estos tipos de sistemas RI, muchos documentos, posiblemente relevantes, son obviados. Maron y Kuhns (1960) propusieron una de las primeras ideas de RI probabilístico en 1960. Los modelos probabilísticos estiman la probabilidad de los documentos relevantes para una *query*. Esta estimación es una parte importante del modelo y es donde difieren unos modelos de otros. Teniendo esto en cuenta, se han propuesto y desarrollados muchos modelos probabilísticos basados en diferentes técnicas de estimación probabilística. La probabilidad de relevancia P de un documento D viene dada por $P(R|D)$, donde R es la relevancia del documento D. Dado que este criterio de clasificación es del tipo Razón de momios[1] o, en ingles *loggodds*, concretamente del tipo monótona, podemos clasificar los documentos mediante $\log \frac{P(R|D)}{P(\bar{R}|D)}$. Donde $P(\bar{R}|D)$ es la probabilidad de los documentos no relevantes. Esta ecuación, utilizada con una transformación bayesiana simple, se convierte en $\log \frac{P(D|R)P(R)}{P(D|\bar{R})P(\bar{R})}$.

Suponiendo que, a priori, la probabilidad de relevancia $P(R)$ es independiente del documento que se está analizando y es constante en los documentos $P(R)$ y $P(\bar{R})$, que estos factores son de ámbito local y que, para las puntuaciones finales, pueden ser eliminados para la clasificación de la estimación. La anterior formula se simplifica quedándose como $\log \frac{P(D|R)}{P(D|\bar{R}))}$. Las suposiciones de estimación de $P(D|R)$ son una de las principales diferencias entre los modelos probabilísticos de RI. La forma más simple de este modelo asume que los términos, normalmente palabras, son mutuamente independientes, en ingles "*independence assumption*", y $P(D|R)$ es redefinido como un producto individual de probabilidades de términos. Por ejemplo, la probabilidad de presencia/ausencia de un término en un documento rele-

[1] https://es.wikipedia.org/wiki/Razón_de_momios

vante/irrelevante está definido como $P(D|R) = \prod_{t_i \in Q,D} P(t_i|R) \prod_{t_j \in Q,\bar{D}} (1 - P(t_j|R))$. Esta fórmula utiliza la probabilidad de la presencia de un término t_i en documentos relevantes, para todos los términos que tienen en común con la *query* y el documento. También utiliza la probabilidad de ausencia de un término t_j de un documento relevante, de todos los términos que están presentes en la *query* y ausentes en el documento. Si p_i define $P(t_i|R)$ y q_i define $P(t_i|\bar{R})$, la fórmula de clasificación $\log \frac{P(D|R)}{P(D|\bar{R}))}$ se reduce a $\log \frac{\prod_{t_i \in Q,D} p_i \prod_{t_j \in Q,\bar{D}} (1-p_j)}{\prod_{t_i \in Q,D} q_i \prod_{t_j \in Q,\bar{D}} (1-q_j)}$. Para una *query* concreta, podemos añadir una constante $\log(\prod_{t_i \in Q} \frac{1-q_i}{1-p_1})$ para transformar la fórmula de clasificación y usarla solamente con terminos presentes en un documento. El resultado es la fórmula $\log \prod_{t_i \in Q,D} \frac{p_i(1-q_i)}{q_i(1-p_i)} o \sum_{t_i \in Q,D} \log \frac{p_i(1-q_i)}{q_i(1-p_i)}$. Diferentes suposiciones para las estimaciones p_i y q_i generan diferentes funciones de clasificación de documentos. Por ejemplo, Croft y Harper (1979) asumen que p_i es igual para todos los terminos de una *query* y que $\frac{p_i}{(1-p_i)}$ es una constante. Esta constante puede ser ignorada según el propósito de la clasificación. También asumen que casi todos los documentos de la colección son irrelevantes para *querys* concretas debido a q_i para la estimación $\frac{n_j}{N}$. Donde N es el número de los documentos dentro de la colección y n_j es el número de documentos que contienen $termino_i$. Este planteamiento, hoy en día ha sido ratificado debido a que en una colección existen muchos documentos sin ninguna conexión entre ellos. Esto genera una función de puntuación $\sum_{t_i \in Q,D} \log \frac{N-n_i}{n_i}$. Esta función es similar a la de IDF, vista en los modelos VSM de la Sección 2.2.3. Teniendo esto en cuenta, si pensamos que $\log \frac{p_i(1-q_i)}{q_i(1-p_i)}$ es la ponderación de $termino_i$ en el documento D, esta fórmula se parece a la Funcion de Similitud, vista en los modelos VSM de la Sección 2.2.3, con la asignación de puntuaciones a los términos de una *query*.

Modelo de inferencia en redes

Este enfoque, que en inglés se denomina como *Inference Network Model*, genera modelos de RI que utilizan los documentos y los modela como un proceso de inferencia en una red de inferencias, tal y como describen Turtle y Croft (1989). La mayoría de las técnicas y enfoques de sistemas de RI pueden ser implementados con estos modelos. La implementación más básica de este modelo utiliza una *query* para construir instancias de términos de los documentos con una cierta puntuación y relevancia. Estas instancias cuentan la puntuación de los términos, incluidos en la colección de documentos, para cuantificar la importancia de cada documento dentro de la *query*. Desde un punto de vista operacional, la importancia de las instancias de un término dentro de un documento puede ser considerada como la puntuación de dicho termino dentro del documento. Además, la clasificación de documentos en esta la forma más sencilla de este modelo, se convierte en una clasificación similar a la de VSM y modelos probabilísticos descritos anteriormente. La importancia de las instancias de

los términos para un documento no están definidas por el modelo y puede utilizarse cualquier formulación para definirlas.

2.2.4 Evaluación de los modelos RI

El cálculo y la evaluación de la eficacia de los resultados obtenidos de una *query* ha sido, y sigue siendo actualmente, uno de los aspectos fundamentales empleados por los investigadores para validar las distintas técnicas y metodologías de la RI. Teniendo esto en cuenta, Cleverdon (1967), utilizando los tests de Cranfield, realizó varias pruebas en 1960, estableciendo una serie de características para la evaluación de los sistemas RI. Aunque la comunidad científica dispone de estas medidas, principalmente suele utilizar 2 métricas para evaluarlos, "*precision*" y "*recall*".

"*Precision*" es el número de documentos relevantes para el usuario recuperados en base a la *query* que ha realizado. Su definición es la Fórmula 2.1.

$$Precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|} \tag{2.1}$$

"*Recall*" es el número de documentos que son relevantes en base a la *query* que han sido recuperados. Su definición es la Fórmula 2.2

$$Recall = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{relevant\ documents\}|} \tag{2.2}$$

Un buen sistema RI es el que recupera gran cantidad de documentos relevantes (alto *recall*) y pocos irrelevantes (alta *precision*). Por otra parte, estas dos métricas han demostrado ser un poco contradictorias a lo largo de los años. Si tendemos a mejorar el *recall*, empeoraremos la *precision* y al revés. Ambas medidas están orientadas hacia un propósito concreto y no tienen noción de la clasificación de documentos recuperados. Todo depende del objetivo que se quiere alcanzar, si obtener muchos documentos relevantes o pocos irrelevantes. Los investigadores han utilizado diversas variaciones de estas métricas para evaluar la clasificación de documentos recuperados. Una muy importante y muy usada por parte de los investigadores, es el cálculo de la media de la *precision*. Esta medida se obtiene calculando la *precision* en diferentes puntos de la recuperación de documentos (al 10 %, 20 %, etc.) para finalmente hacer la media de todos los valores, tal y como describen Salton y McGill (1983).

2.2.5 Frases cortas

La detección del contexto en frases cortas es un área relativamente nueva e importante dentro del campo del PLN y, concretamente, dentro de RI. Para recuperar el

contexto, existen diferentes enfoques. Uno de ellos es el utilizado por Wang et al. (2015) que utiliza un mapa de confidencias e información contextual para detectar, de manera robusta, textos en escenarios naturales. Otro enfoque, más visual, es la herramienta *conTEXT*, desarrollada por Khalili et al. (2014). Esta herramienta muestra multitud de conexiones entre diferentes palabras que se han extraído de recursos como Twitter, además de otros datos interesantes, y muestran la definición que DBPedia propone para la palabra. Liu et al. (2014b) utiliza redes heterogéneas de contexto enriquecido, o en inglés "*context-rich heterogeneous network*", para recomendar referencias en publicaciones científicas utilizando el propio texto del artículo a referir.

En base a lo expuesto hasta ahora, hemos considerado que la identificación de *topics* es un enfoque muy atractivo y, por ese motivo, hemos decidido centrar nuestra investigación en este área. Según Hong et al. (2011), la identificación de *topics* en frases cortas difiere con la de los documentos en varios aspectos. Los más significativos son:

1. El problema de la mezcla de idiomas dentro de cada mensaje.

2. Las frases cortas contienen muchos términos que no dan nada de información, solo sirven para mantener la conversación activa y aparecen en gran cantidad.

3. La cantidad de elementos para analizar es mucho menor en las frases cortas, llegando a tener solamente unas pocas palabras por mensaje.

Además, la detección de la temática en documentos grandes, según Sebastiani (2002), se supone un problema ya resuelto. En cambio, la detección de temáticas en mensajes cortos, todavía no tiene solución. Existen varias técnicas que intentan, mediante diferentes métodos, lograr este objetivo.

Kolenda et al. (2001) propone "*Independent Component Analisys*" (ICA) junto con "*Latent Semantic Analisys*" (LSA) para detectar los *topics* dentro de los chats.

El proceso que estos autores proponen comienza con la eliminación de los stopwords[2] (Wilbur y Sirotkin, 1992) presentes en los mensajes, continúa dividiendo éstos en sesiones y aplicando LSA a dichas sesiones para, después, utilizar ICA para analizar cada sesión. El problema está en que es muy dependiente de las particiones de los mensajes. El trabajo de Bingham et al. (2003) es bastante semejante al de Kolenda et al. (2001), pero cambiando ICA por "*Complexity Pursuit*" (CP) y las particiones se hacen por periodos de tiempo, no por cada mensaje. Este trabajo presenta problemas semejantes al anterior añadiendo la dependencia del tiempo. Adams y Martell (2008) proponen un método basado en $TD-IDF$, que convierte las frases

[2]Palabras sin significado como por ejemplo, los artículos, pronombres o preposiciones.

en vectores y calcula su diferencia en base a la distancia del coseno. Además le suma una penalización en base al tiempo que hay entre cada frase y hace una ampliación de los hipónimos, en base a *WordNet*, y de los "*nicknames*". El problema es que este enfoque no contempla la semántica que puede contener cada frase. Gainaru et al. (2010) han desarrollado una herramienta que utiliza $TF - IDF$ junto con *n-grams* y un nuevo enfoque conocido como "*Cue Phrase Analysis*" (CPA) para la detección y análisis de textos cortos.

Pons-Porrata et al. (2007) proponen un sistema que utiliza el resumen de los *topics* en un periodo de tiempo junto con un método jerárquico y agrupado como base para clasificar nuevos *topics*. Dong et al. (2006) extraen características de la sesión como las URLs o los iconos. Además, utilizan clasificación con vectores junto con otros algoritmos como *Naïve Bayes* o Clasificación Asociada (CA). En el trabajo de Bengel et al. (2004), generan vectores de *topics* y, mediante la distancia del coseno, clasifican nuevos *topics* para detectar cierto tipo de perfiles. Hui et al. (2008) desarrolló un sistema llamado "*IMAnalysis*" que obtiene información de diferentes redes sociales y extrae los *topics* mediante algoritmos de clasificación como *Naïve Bayes*, K-nn, SVM y CA. Estos clasificadores también son utilizados por Dong et al. (2006) y Özyurt y Köse (2010) junto con J-48 y clasificadores de regresión en el trabajo de RahmanMiah et al. (2011).

Zhang et al. (2014) proponen un método que utiliza un pre-procesado como el de los anteriores trabajos, sumándole una traducción de todas las palabras a un idioma concreto. Después, utiliza "*Probabilistic Latent Semantic Analisys*" (PLSA) con las palabras para generar una matriz de probabilidades con el objetivo de, finalmente extraer los *topics*. Chen et al. (2012) propone otro tipo de enfoque para solucionar la ausencia de semántica en la extracción de *topics* mediante "*Semantic Dependency Distance*" (SDD) y PLSA en Chino.

El gran problema de la mayoría de estos estudios, suele ser que las pruebas están hechas en idiomas concretos, principalmente inglés, y la utilización del mismo sistema en otros idiomas, no está probada. También es un problema la escasez de datos que lleva la frase. Esto hace más difícil detectar los *topics*. Por otra parte, es necesario tener en cuenta la semántica y el contexto para una mejor detección de los *topics*.

2.3 Clasificación documental de temáticas

El Ser Humano, como ser racional, se encuentra continuamente buscando el orden de las cosas. Como consecuencia de esta búsqueda, ha desarrollado un profundo interés por la clasificación. La finalidad es poder conocer y utilizar todos los elementos que nos rodean. El texto, como estructura, no ha permanecido ajeno a este interés

por la categorización inherente al Ser Humano. Se catalogan todas las frases, palabras y elementos lingüísticos para poder conocerlos y utilizarlos. En este apartado revisamos la relación de la categorización de texto con otras tareas con el fin de aclarar conceptos, presentando las distintas aplicaciones y técnicas, y enfatizando en el uso del Aprendizaje Automático para la construcción de clasificadores. Además, se mostrará el procedimiento de evaluación que se utiliza en la bibliográfica, se detallarán las métricas con las que se valoran los resultados y se ofrecerá una descripción de algunas de las colecciones de evaluación disponibles.

2.3.1 Tareas de clasificación de texto

La categorización de texto, tal y como dice determina Hearst (1999), es una labor de clasificación de entidades textuales dentro de un amplio abanico de tareas enmarcadas en el campo de la minería de texto (text mining). Se trata de un campo que vive una expansión sin precedentes, motivada por la enorme cantidad de texto en formato electrónico disponible en múltiples entornos, muy especialmente en Internet y la World Wide Web. Las tareas de clasificación de textos se han abordado mayoritariamente utilizando técnicas estadísticas, por lo que son el objetivo del PLN estadístico. Por contraposición, el PLN basado en conocimiento utiliza modelos que pretenden capturar de manera precisa el significado del texto. La amplia gama de tareas de clasificación de texto que dan soporte al descubrimiento de nuevo conocimiento a partir del texto es muy amplio. Lewis (1992) lo organiza en base a dos criterios. Por una parte, en base al tipo de aprendizaje utilizado, y por otro lado, en base a la granularidad de los textos a clasificar, siendo los tipos de aprendizaje utilizados:

- **Aprendizaje supervisado o categorización**, en el que las clases se conocen de antemano y se disponen de ejemplos de textos clasificados en cada categoría.
- **Aprendizaje no supervisado o agrupamiento (clustering)**, en el que las clases no son conocidas a priori y el objetivo es agrupar las entidades textuales de acuerdo a su contenido, de modo que entidades similares pertenezcan al mismo grupo.

Por otro lado la granularidad de los textos se puede organizar en varios niveles:

- **Términos**, que incluye a palabras aisladas o expresiones cortas, o incluso raíces de palabras.
- **Frases**, en las que se incluyen desde cláusulas a oraciones completas.

- **Documentos**, que pueden ser tan breves como correos electrónicos, o tan largos como libros.

De acuerdo con esta organización, la tabla 2.1 muestra algunas de las tareas de clasificación más representativas. Por ejemplo, según Sanderson (1994), la desambiguación de términos (Word Sense Disambiguation) o resolución de la ambigüedad léxica, trata de identificar el significado de una palabra según el contexto de aparición. Se trata de una labor supervisada porque los posibles significados son conocidos de antemano, y afecta a términos compuestos por una o pocas palabras. Igualmente, como explica Maña López et al. (2010), la generación automática de resúmenes informativos se puede abordar como un trabajo de clasificación de oraciones, en las que se decide cuales deben aparecer en el resumen y cuales no, en función de una serie de ejemplos de otros resúmenes y documentos.

Tabla 2.1: Organización de algunas tareas de clasificación de texto.

	Aprendizaje supervisado	Aprendizaje no supervisado
Términos	Desambiguación del significado Etiquetado sintáctico (POS Tagging) Reconocimiento de entidades nombradas Resolución de la anáfora pronominal Análisis parcial (Partial Parking/Chunking)	Descubrimiento de nuevos significados
Frases	Generación automática de resúmenes	Segmentación de texto
Documentos	Recuperación de documentos Categorización de textos	Agrupamiento de documentos

En general, estas tareas pueden tener sentido de manera autónoma (como ocurre frecuentemente en la categorización de textos), o pueden estar al servicio de otras funciones o de otras aplicaciones más generales. Por ejemplo, el etiquetado sintáctico es un paso casi imprescindible antes de realizar el análisis sintáctico de una oración. Igualmente, la desambiguación del significado se utiliza frecuentemente en métodos de recuperación de textos para reducir los errores producidos por la variabilidad del lenguaje, y es imprescindible para lograr una traducción automática de cierta calidad.

En resumen, la categorización de texto es una tarea de clasificación supervisada de documentos, que emplea habitualmente técnicas de PLN estadístico, y que se sitúa en el marco de la minería de texto.

2.3.2 Categorización automática basada en aprendizaje

La categorización se puede realizar de manera manual o automática. En caso de realizarse automáticamente, el clasificador puede ser construido manualmente a partir del conocimiento de expertos catalogadores, o bien automáticamente utilizando técnicas de Recuperación de Información y de Aprendizaje Automático.

Este último enfoque es el denominado como "categorización automática de texto basada en aprendizaje", y es el que se emplea con éxito en múltiples tareas de clasificación de texto con adversario[3], como puede ser el filtrado de spam. En los ejemplos nos basaremos en el spam ya que, en numerosas ocasiones, el spam esta constituido por frases cortas. Así mismo, se puede añadir que esta aproximación ha alcanzado una efectividad comparable a la de los catalogadores profesionales como bien se muestra en el trabajo de Sebastiani (2002).

2.3.2.1 Inducción automática de clasificadores

El objeto de la categorización automática basada en aprendizaje es la construcción automática de un clasificador de textos a partir de una serie de ejemplos. Minimizando el factor humano en la construcción del clasificador, podemos resolver el problema del "cuello de botella de la adquisición de conocimiento", tan común en todas las aplicaciones basadas en conocimiento, como los sistemas expertos.

La categorización de texto consiste en asignar un valor booleano a cada par (d_j, c_k) de $D \times C$, siendo D un conjunto de documentos sin clasificar, y C un conjunto predefinido de categorías. La asignación del valor cierto al par (d_j, c_k) se interpreta como que el documento d_j se clasifica dentro de la categoría c_k. Más formalmente, la tarea consiste en aproximar una función desconocida $\Phi : D \times C \rightarrow T, F$, que describe cómo deberían ser clasificados los documentos, por medio de una función $\Phi' : D \times C \rightarrow T, F$, llamada clasificador, hipótesis o modelo, de modo que coincidan en la medida de lo posible.

Dado un documento, el número de categorías a las que se asigna es variable y dependiente de la aplicación y de las técnicas empleadas. En algunas situaciones, a cada documento se le asigna una sola categoría. En otras ocasiones, se le puede asignar un número fijo $k > 0$, o hasta k categorías, o al menos k, siendo $k \preceq |C|$. Por ejemplo, a cada nueva adquisición de una biblioteca se le asignan una o más

[3]Texto que va siendo modificado para evadir sistemas de reconocimiento.

categorías, pero cada noticia de un periódico se presenta en una sola sección. Cuando un documento se puede asignar a varias categorías, se dice que las categorías se solapan. Para resolver el problema de la asignación de múltiples categorías a cada documento se puede construir un clasificador binario por cada categoría (es decir, para cada c_k, se construye un clasificador $\Phi_k : D \rightarrow T, F$ que, dado un documento, sólo puede asignar el mismo a c_k o a su complementaria, para cada $1 \preceq k \preceq |C|$). Este enfoque exige que las $|C|$ categorías sean estocásticamente independientes entre si, es decir, que dadas c_m y c_n, $m \neq n$, el valor de $\phi(d_j, c_m)$ no dependa de $\phi(d_j, cn)$ y viceversa, situación que no siempre se da en la realidad. En general, los problemas de clasificación de texto con adversario involucran sólo dos categorías (p. ej., correo basura o legítimo), aunque en el caso del filtrado de contenidos, se suele construir un clasificador binario para cada dominio: pornografía, violencia, juegos de casino, etc.

La construcción de clasificadores ϕ' es el objetivo del desarrollo de sistemas de categorización automática de texto. Para ello, se han aplicado diversas técnicas provenientes generalmente del campo de la Recuperación de Información y del Aprendizaje Automático. En concreto, se pretende construir ϕ' de modo que, a partir de las propiedades del documento a categorizar, pueda tomar la decisión de clasificación para cada categoría. Las propiedades de los documentos que se utilizan en los categorizadores automáticos son usualmente las palabras que aparecen en ellos. Las palabras suelen dar una idea básica de la temática del documento, lo que permite construir clasificadores temáticos efectivos. En otras palabras, una representación de los documentos basada en palabras captura las propiedades semánticas básicas del documento.

La representación basada en palabras es la utilizada con más frecuencia en la Recuperación de Información y es la base a partir de la cual se pueden aplicar numerosos algoritmos de Aprendizaje Automático. Existen múltiples algoritmos que producen una gran diversidad de clasificadores, que pueden ser sistemas de reglas, árboles de decisión, redes neuronales, redes bayesianas, funciones lineales, tablas probabilísticas, etc. La idoneidad de un algoritmo u otro para un problema concreto depende de las condiciones del mismo, en términos de eficacia, eficiencia en la fase de aprendizaje y en la de clasificación, comprensibilidad del clasificador, etc. Esta idoneidad se suele evaluar sobre una colección de datos de prueba que representen, del modo más fiel posible, la realidad que se está estudiando.

2.3.2.2 **Estructura de un categorizador basado en aprendizaje**

El funcionamiento de un clasificador implica tomar decisiones sobre la representación y el aprendizaje, y contiene dos flujos de proceso, uno no interactivo y otro interactivo. Estos dos flujos se muestran en la figura 2.2.

El proceso no interactivo incluye en primer lugar la obtención de una representación de los documentos que, por un lado, capture en lo posible su significado, y por el otro, permita el aprendizaje a partir de ella. Usualmente, este paso incluye la definición y cálculo de los parámetros de un modelo inicial de representación y la aplicación de alguna técnica de selección y/o extracción de términos. El segundo paso incluye la aplicación de un algoritmo de aprendizaje (entrenamiento) sobre los documentos de los que se conoce su clase, representados anteriormente, lo que da lugar a un modelo de clasificación o clasificador.

El modelo obtenido puede ser aplicado sobre otros documentos representados de manera similar, con el objetivo de obtener las categorías asociadas a los mismos. El proceso interactivo consiste precisamente en representar los documentos a clasificar en términos similares a los de la colección de entrenamiento, y aplicarles el clasificador obtenido en el entrenamiento.

El proceso no interactivo está concebido para ser ejecutado periódicamente, dependiendo del problema que se aborde. La existencia de algoritmos incrementales que pueden refinar iterativamente su modelo sobre nuevos documentos permite que el entrenamiento se pueda realizar incluso cuando se reciba cada nuevo documento clasificado. Esto es especialmente útil en la clasificación con adversario, ya que las técnicas de los adversarios cambian con rapidez con el fin de comprometer al clasificador.

Distintos regímenes de entrenamiento, como los descritos en el trabajo de Puertas Sanz et al. (2008), incluyen el entrenamiento sobre errores y el entrenamiento completo. Este último se basa en que si una clasificación no ha sido corregida por un usuario, es que es correcta, y se incorpora el documento a la colección de entrenamiento. El primero sólo se entrena al incorporar documentos que corresponden a correcciones del usuario (por ejemplo, un correo basura clasificado como legítimo y, por tanto, mostrado al usuario).

2.3.2.3 **Representación de los documentos**

Los documentos de texto, tal y como se presentan en diferentes dominios de estudio, están concebidos para el consumo humano y no para su proceso automático. El análisis del contenido textual de los documentos preclasificados tiene como fin la definición y obtención de una representación de los mismos a la que se puedan aplicar técnicas de aprendizaje para obtener un modelo de clasificación. Este proceso de análisis es equivalente al realizado en los sistemas de recuperación de documentos, y con frecuencia se denomina de igual modo, esto es, “indexación” de los documentos. Como en la recuperación de documentos, el objetivo básico, tal y como describen Salton y McGill (1983) y Lewis (1992), es diseñar y obtener una representación que capture, en la medida de lo posible, el significado o semántica del texto.

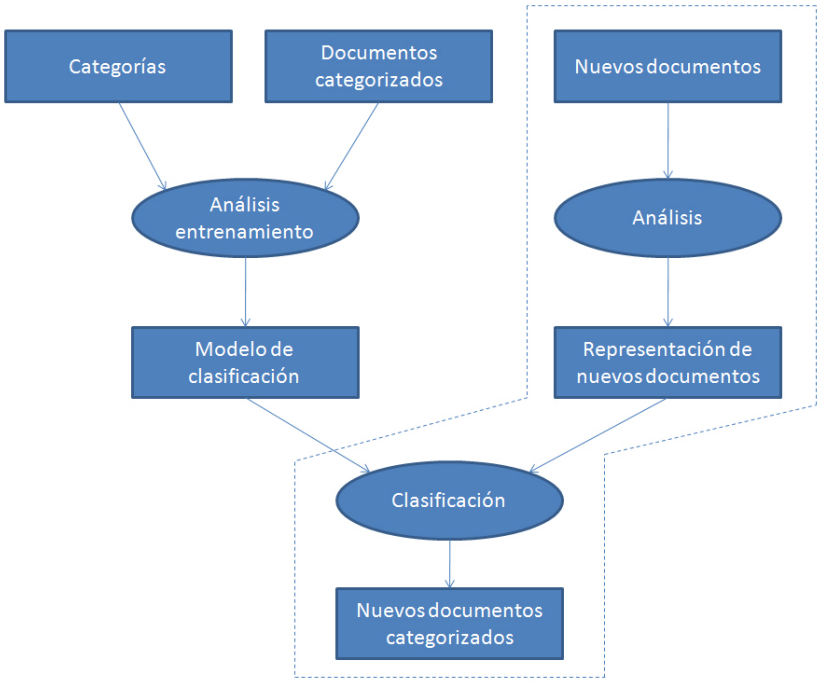


Figura 2.2: Estructura de procesamiento de un categorizador de texto basado en aprendizaje.

El modelo de representación más frecuentemente utilizado es el Modelo del Espacio Vectorial descrito en el trabajo de Salton y McGill (1983). En este modelo, los documentos se representan por medio de vectores de pesos de los términos. Dicho peso representa la importancia del término en el documento, aproximando de alguna manera la contribución que el término hace al significado del documento.

En la literatura existen diversas variaciones sobre este modelo, dependiendo fundamentalmente del modo que se definen los términos, y en cómo se calculan sus respectivos pesos en los documentos. El enfoque más habitual en la bibliografía es identificar los términos (p.ej.: palabras individuales) que aparecen en los documentos de entrenamiento. Este enfoque es referido típicamente como el modelo de "bolsa de palabras" (bag of words). En varios trabajos se ha demostrado que representaciones más sofisticadas que ésta no son necesariamente más efectivas (Apté et al., 1994; Dumais et al., 1998; Lewis, 1992). El proceso de separar un texto en palabras se suele denominar *tokenización* (tokenization), y es un punto clave en la clasificación con adversario, ya que ha sido el objeto de la mayoría de ataques por parte de los *spammers*.

Con mucha frecuencia, cuando los pesos son palabras, se hace uso de una lista de parada para eliminar aquellas palabras más frecuentes en la colección de documen-

tos. Generalmente estas palabras son preposiciones, artículos, pronombres, etc. que, en general, contribuyen poco al significado de los documentos de cara a su clasificación temática, con excepciones en algunos dominios como la identificación de autor. Igualmente, el uso de métodos de extracción de raíces como el algoritmo de Porter, explicado en el trabajo de Willett (2006), que es casi invariable en la Recuperación de Información, también se utiliza frecuentemente en el proceso de categorización. Ello se debe a que, tal y como se describe en el trabajo de Sebastiani (2002), reduce notablemente el número de términos de indexación y simplifica las dependencias estocásticas entre los términos, lo que facilita la fase de aprendizaje.

2.3.2.4 **Técnicas de selección y extracción de términos**

La mayoría de los algoritmos de aprendizaje tienen dificultades de eficiencia, si no problemas mayores, en presencia de grandes cantidades de atributos. Por ello, se suele realizar un proceso de reducción de la dimensionalidad del espacio de representación, esto es, disminuir notablemente el número de atributos usados para modelar los documentos (alrededor de un 1-10 % es una cifra típica tal y como se describe en los trabajos de Yang y Pedersen (1997) y en el de Sebastiani (2002).

Las técnicas de reducción de la dimensionalidad se suelen agrupar en dos conjuntos, dependiendo de si el conjunto reducido de términos es un subconjunto del original (selección de términos) o de si es un nuevo conjunto de términos sustancialmente distinto en la forma de los mismos (extracción de términos).

Las técnicas de selección de términos, tal y como se describe en el trabajo de Yang y Pedersen (1997), tienen como fin obtener un subconjunto de los originales (después de la eliminación de las palabras de la lista de parada y de la extracción de raíces), de modo que el conjunto resultante posea un poder predictivo con respecto a las categorías igual o incluso mayor que el original. La manera más habitual de seleccionar términos es elegir aquellos que aisladamente obtienen un valor predictivo mayor, de acuerdo a alguna métrica de calidad que lo caracteriza, como la Ganancia de Información utilizada en el trabajo de Lewis (1992); Larkey (1998), la Información Mutua utilizada por Dumais et al. (1998), o χ^2 que utiliza Yang y Pedersen (1997).

Las técnicas de extracción de términos pretenden generar, partiendo del conjunto original, un conjunto con un cardinal menor formado por términos artificialmente creados. Una razón para usar términos artificiales en vez de las palabras se debe a los problemas de polisemia y sinonimia. Los términos originales pueden no ser los representantes óptimos de los documentos. Las dos técnicas de extracción de términos que se han utilizado en la categorización automática son: i) agrupamiento de términos descrito por Lewis (1992) e ii) indexación semántica latente descrito por Li y Jain (1998).

2.3.2.5 **Algoritmos de aprendizaje**

El elemento al que más atención se le presta en general en los métodos de inducción de categorizadores automáticos es el algoritmo de aprendizaje. Los algoritmos de aprendizaje tienen como objetivo la construcción de la función $\phi' : D \times C \rightarrow T, F$ reseñada anteriormente, que no trabaja realmente sobre los documentos de aprendizaje sino sobre la representación de los documentos obtenida utilizando una serie de métodos de representación, selección y extracción de términos.

Dado el creciente interés de los expertos en el Aprendizaje Automático en la clasificación de documentos, son numerosos los algoritmos de aprendizaje que se han aplicado a la inducción de categorizadores automáticos, incluyendo: modelos probabilísticos como Bayes ingenuo (Naive Bayes), inducción de árboles de decisión como C4.5, métodos de aprendizaje basados en reglas como Ripper, aprendizaje basado en ejemplares como el de los k-vecinos más próximos (k Nearest Neighbours, kNN), inducción de clasificadores lineales como el algoritmo de Rocchio o métodos de regresión como LLSF (Linear Least Squares Fit), redes neuronales, Support Vector Machines (SVM), etc.

Estos algoritmos difieren en gran medida entre sí en términos del tipo de modelo que generan, de su capacidad para procesar grandes cantidades de atributos, de su eficiencia y de su eficacia. A fecha de hoy, las SVMs se han convertido prácticamente en un estándar en el ámbito de la clasificación de texto, ya que son capaces de trabajar con grandes cantidades de atributos y suelen obtener los modelos más eficaces en múltiples dominios.

Sin embargo, en la clasificación de texto con adversario, y más concretamente en el filtrado de spam, el algoritmo más empleado en aplicaciones industriales es el clasificador bayesiano ingenuo. Ello se debe a que es fácil de implementar, es incremental (como se puede actualizar el modelo con cada nuevo documento agregado, resulta muy eficiente en la fase de entrenamiento), y que en general es capaz de ofrecer resultados excelentes en lo que se refiere a efectividad, como en el caso del correo basura (superiores al 99 % de éxito). En cualquier caso, no es el más efectivo, de acuerdo con los resultados obtenidos en distintas evaluaciones competitivas, como el TREC Spam Track descrito en el trabajo de Cormack y Lynam (2005).

2.3.3 **Algoritmos y modelos de PLN**

Para llevar a cabo la tarea de ordenar, buscar, analizar y procesar los documentos o corpus que se generan cada día con el objetivo de obtener información útil para cualquier actividad, actualmente existen varias técnicas y modelos. En el ámbito del PLN los datos se trabajan con algoritmos y modelos tales como, por ejemplo, N-gramas (Brown et al., 1992), análisis semántico latente (LSA) (Landauer et al., 1998), aná-

lisis probabilístico de la semántica latente (PLSA) (Hofmann, 2001), Latent Dirichlet allocation (LDA) (Blei et al., 2003), distribución de Pólya (Elkan, 2006), algoritmo de "*expectation maximization*" (EM) (Dempster et al., 1977), Indexación Semántica Latente LSI (Dumais et al., 1988), term frecuency-inverse document frecuency (tf-idf) (Joachims, 1997) y SVM (Dumais et al., 1998). Algunas de estas técnicas se basan en modelos probabilísticos.

El modelo probabilístico de N-gramas sirve para predecir el siguiente elemento en una secuencia de modelo de Markov (Abney y Light, 1999) de la forma (n-1). Un ngrama es una secuencia contigua de n elementos de una determinada secuencia de texto o voz y pueden tener cualquier combinación. Normalmente, los productos que se generan suelen ser fonemas, silabas, letras, palabras, etc. Los ngramas comúnmente suelen obtenerse de un texto o un corpus.

La técnica LSA es muy utilizada en procesamiento de lenguaje natural, en concreto en la semántica vectorial (modelo algebraico para representar documentos como vectores) y sirve para analizar las relaciones entre un conjunto de documentos y los términos que contienen. Esto se lleva a cabo mediante la producción de una serie de conceptos relacionados con los términos y los documentos. LSA se basa en que las palabras que tienen mismo sentido, estarán en trozos similares de texto. La técnica genera una matriz que contiene el recuento de palabras por párrafo (las filas representan las palabras únicas y las columnas representan cada párrafo) a partir de un texto. Después utiliza el método matemático de descomposición de valor singular (SVD), descrito en el trabajo de Trefethen y Bau III (1997), para reducir el número de columnas manteniendo la relación de similitud con las filas. Una vez obtenida la matriz reducida, las palabras se comparan tomando el ángulo del coseno que forman los vectores de dos filas. Los valores cercanos a 1 representan palabras muy similares, mientras que los cercanos a 0 representan palabras muy diferentes tal y como se describió en el trabajo de Dumais et al. (1988).

La técnica estadística PLSA o indexación probabilística de la semántica latente (PLSI) esta basada en LSA. Se basa en una descomposición mixta obtenida de LSA. La mejora que genera con respecto a LSA es utilizar métodos estadísticos para reducir las variables observadas y su afinidad respecto a ciertas variables ocultas. Por ejemplo, sirve para realizar el análisis de coocurrencia.

El modelo estadístico LDA es un modelo generativo (modelo para la generación aleatoria de datos observables) que permite mediante grupos observables explicar los grupos no observables, explicando porque algunas partes de los datos son similares. Por ejemplo, si el grupo observable está formado por palabras recogidas de documentos, el objetivo es que cada documento sea una mezcla de un número pequeño de temas y que la creación de esos temas venga dada por ciertas palabras. En LDA, cada documento puede ser visto como una mezcla de varios temas. Esto es

similar a PLSA, excepto que en LDA se asume que la distribución de categorías tiene, a priori, una distribución de Dirichlet.

La distribución estadística multinominal Dirichlet es una distribución de probabilidad para variables discretas aleatorias en un ámbito multivariante. Normalmente, esta distribución se suele llamar distribución multinomial del compuesto Dirichlet (DCM) o distribución multivariable Pólya. Es una distribución de probabilidad compuesta que utiliza vectores de probabilidad p con parámetros alpha extraídos de una distribución Dirichlet y un conjunto de muestras discretas extraídas de la distribución categórica con las probabilidades del vector p. En el ámbito de clasificación de documentos se utiliza para representar la distribución del recuento de palabras para los diferentes tipos de documentos.

El algoritmo EM es un método iterativo para encontrar el elemento más semejante (máxima probabilidad estimada) o el maximum a posteriori (MAP) (elemento de estadística bayesiana para determinar la probabilidad posterior sobre un elemento aleatorio con probabilidad condicional a la evidencia obtenida a partir de un experimento o encuesta previa). El modelo depende de variables latentes no observadas. El uso más frecuente para el que se usa este modelo es para agrupar datos mediante el aprendizaje automático. En procesamiento del lenguaje natural, dos usos destacados de este algoritmo son el algoritmo Baum-Welch o algoritmo adelante-atrás, descrito en el trabajo de Baum et al. (1970), que se utiliza para encontrar los parámetros desconocidos de un modelo oculto de Markov, y el algoritmo dentro-fuera (inside-outside), descrito en el trabajo de Baker (1979), que se utiliza para modelos probabilísticos no supervisados de gramáticas libres de contexto.

El método LSI se usa para indexar y recuperar datos mediante la técnica matemática de SVD (es una factorización de una matriz real y compleja) que identifica patrones en las relaciones entre los términos y conceptos de una colección estructurada de texto o corpus. LSI se basa en el principio de que las palabras que se utilizan en el mismo contexto tienden a tener un significado similar. Una característica importante de LSI es la capacidad que tiene para extraer el contenido conceptual de un texto mediante el establecimiento de asociaciones entre los términos que aparecen en contextos similares. También sirve para el análisis de correspondencias, tal y como lo describe en su trabajo Deerwester (1988).

El modelo de tf-idf es una estadística numérica que refleja la importancia de una palabra dentro de un documento o copus. Se suele utilizar como elemento ponderador en la recuperación de información y data mining. El valor tf-idf aumenta proporcionalmente según el número de veces que una palabra aparece dentro de un documento, pero para compensar el exceso de apariciones de dicha palabra, se analiza también la frecuencia de esa palabra en el corpus. De esta manera se controla la existencia de palabras más comunes que otras.

El modelo SVM es se encuentra dentro de los algoritmos de *machine learning* supervisados. Este modelo se utiliza para analizar y reconocer patrones con el objetivo de clasificarlos en base al análisis de regresión. A fecha de hoy, las SVM se han convertido prácticamente en un estándar en el ámbito de la clasificación de texto, ya que son capaces de trabajar con grandes cantidades de atributos y suelen obtener los modelos más eficaces en múltiples dominios.

2.3.4 **Evaluación**

La evaluación de un sistema de categorización es un aspecto crítico, en tanto que sin ella no es posible tomar decisiones sobre su calidad o sobre su implantación. En general, la evaluación de los sistemas de clasificación está centrada en la eficacia o efectividad, ya que se trata de sistemas que toman decisiones que pueden estar equivocadas (a diferencia de, por ejemplo, los Sistemas de Gestión de Bases de Datos: si toman una decisión equivocada ante una consulta, no son poco efectivos: simplemente, no funcionan).

Aunque la evaluación está generalmente centrada en la efectividad, en ocasiones se considera la eficiencia, y con menos frecuencia la comprensibilidad de los modelos, la portabilidad y escalabilidad de las técnicas, etc. En general, las características a tener en cuenta serán las mismas que las utilizadas para un sistema de Procesamiento de Lenguaje Natural, estandarizadas de manera parcial en Europa por el grupo EAGLES, descrito en el trabajo de EAGLES (1995).

Sin embargo, nosotros centraremos la evaluación en la eficacia y en la eficiencia, dado que son dos de los aspectos más críticos, y que son los cubiertos en general por la bibliografía. En la evaluación de la efectividad se pretende estimar la calidad de la función de aproximación o clasificador, en términos de aciertos o fallos sobre una colección de documentos cuyas clasificaciones ya son conocidas, denominada “colección de evaluación” o “de prueba” (“test collection” o “test set”).

En general, resulta esencial que los elementos más relevantes utilizados en la evaluación (procedimientos, métricas y colecciones) sean estándar, con el fin de facilitar la comparación de resultados con trabajos previos o de otros autores. Una dificultad añadida en el caso de la clasificación con adversario es que los documentos problema (i.e. en spam los correos basura) evolucionan con rapidez para adaptarse a los clasificadores más efectivos, por lo que la validez de las colecciones y de los propios clasificadores está limitada en el tiempo.

2.3.4.1 **Procedimiento de evaluación**

Denominamos procedimiento de evaluación al modo en que se procesa la colección de datos con vistas a extraer las medidas de efectividad.

En general, se parte de una colección de documentos categorizados manualmente, sobre la que entrenar y evaluar el clasificador. Obviamente, no es correcto entrenar y evaluar sobre los mismos documentos, por lo que la colección se fracciona en una subcolección de entrenamiento y otra de evaluación, tal y como hace Sebastiani (2002). Este procedimiento se ha aplicado tradicionalmente con la colección Reuters-21578, descrita en el trabajo de Lewis (1997), y que ha llegado a ser la más utilizada en el ámbito de la categorización temática. Sin embargo, en la literatura se han consolidado varias particiones distintas con diferentes grados de dificultad para un clasificador, lo que no ha facilitado la comparación, tal y como describen Yang y Liu (1999) en su trabajo.

El procedimiento anterior priva de datos tanto al entrenamiento (se puede asumir que cuanto mayor sea la colección de entrenamiento, mejor será el clasificador) como a la evaluación (cuanto mayor sea la colección de evaluación, más fiable será la medida). Actualmente el enfoque más seguido (especialmente en los trabajos recientes de clasificación con adversario y en las evaluaciones competitivas asociadas) es la validación cruzada. Esta técnica consiste en dividir la colección global en K grupos de igual tamaño (frecuentemente K=10), de manera que se mantenga la proporción de clases en cada grupo. A continuación, se lanzan K pruebas, usando en cada una de ellas un grupo para la evaluación y el resto para entrenamiento, tomándose una medida que finalmente se promedia sobre el total de las pruebas.

2.3.4.2 **Métricas de efectividad**

Las métricas de eficacia usadas en la evaluación de los sistemas de categorización de texto provienen, como es lógico, de los campos de la Recuperación de Información y del Aprendizaje Automático. Usualmente, estas métricas están basadas en contar el número de éxitos en las decisiones de clasificación en relación con el número total de decisiones a tomar, sobre la colección de evaluación.

Tabla 2.2: Matriz de confusión para dos clases.

	Real $\rightarrow C^+$	Real $\rightarrow C^-$
Clasificador $\rightarrow C^+$	TP	FP
Clasificador $\rightarrow C^-$	FN	TN

Asumamos el caso binario: dos categorías, una C^+ y su complementaria C^-. Dada una colección de prueba con N documentos, se ejecuta el clasificador y se construu-

ye una tabla o matriz de confusión como la de la tabla 2.2. En las filas se muestran las decisiones tomadas por el clasificador, y en las columnas los valores reales. Por ejemplo, la entrada TP (*True Positives*, positivos correctos) representa el número de documentos que el categorizador ha clasificado correctamente en C^+, mientras que FP (*False Positives*, positivos incorrectos) es el número de documentos clasificados incorrectamente por el clasificador en C^+. Los valores FN y TN son obviamente complementarios. En la tabla se puede ver que la diagonal principal representa los aciertos, y la complementaria representa los errores. Obviamente, la suma de los cuatro valores debe ser N. Las medidas más habituales, tal y como se describen en los trabajos de Salton y McGill (1983) y Sebastiani (2002) son las siguientes :

- **Cobertura (recall, R)** – representa la proporción de documentos correctamente clasificados en C+ sobre los que debían estar en ella:

$$R = \frac{TP}{TP + FN}$$

- **Precisión (precision, P)** – representa la proporción de documentos clasificados correctamente en C+ sobre los que han sido clasificados en ella:

$$P = \frac{TP}{TP + FP}$$

- **Exactitud (accuracy, A)** – representa la proporción de aciertos total sobre el número de intentos:

$$A = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Error (E)** – representa la proporción de errores total sobre el número de intentos:

$$E = \frac{FP + FN}{TP + TN + FP + FN}$$

Las medidas de exactitud y error son opuestas, y frecuentemente se descartan en categorización, porque en casos de categorías poco pobladas (o en otras palabras, de eventos poco probables) es muy sencillo conseguir un clasificador muy efectivo de manera trivial. Por ejemplo si hay un elemento en C^+ por cada 99 en C^-, el rechazador trivial (que asigna C^- a cualquier documento) tiene una exactitud del 99 % (0,99), pero es absolutamente inútil tal y como se describe en el trabajo de Sebastiani (2002).

2.3.4.3 **Colecciones de evaluación**

La existencia de colecciones de evaluación estándar es esencial para la eventual comparación de distintos enfoques, especialmente entre trabajos científicos de distintos investigadores.

La que se ha utilizado con más frecuencia para la categorización automática de texto es Reuters-21578, descrita en el trabajo de Lewis (1997). Es una colección de 21.578 noticias periodísticas breves (2-3 párrafos) sobre economía, clasificadas manualmente de acuerdo con un conjunto de categorías temáticas de índole económica. Esta colección ha sido usada, por ejemplo, en los trabajos de Yang y Liu (1999) y Sebastiani (2002). Desgraciadamente, ha sido particionada de varias formas distintas, y se han realizado usos no estándar de la misma. Actualmente, esta colección se ha visto superada por la llamada Reuters Corpus Volumen 1, descrito en el trabajo de Lewis et al. (2004). Esta colección contiene más de 800.000 documentos y es de dimensiones más realistas.

Se han utilizado otras colecciones de evaluación en múltiples trabajos de investigación, con distintas dimensiones y temáticas. Por ejemplo, la colección Ohsumed incluye aproximadamente 300.000 resúmenes de noticias médicas clasificadas con respecto a descriptores médicos (los Medical Subject Headings), y ha sido utilizada en los trabajos de Joachims (1998) y Lewis et al. (1996). O la colección 20-Newsgroups, que incluye 20.000 mensajes publicados en 20 grupos de noticias de temas variados, y ha sido usada en los trabajos de McCallumzy y Nigamy (1998) y Joachims (1997).

Dado que el esfuerzo necesario para construir una colección de evaluación de un gran tamaño es alto, la tendencia actual es construirlas en el marco de evaluaciones competitivas financiadas por organismos gubernamentales u oficiales. Por ejemplo, en el marco de la biomedicina, se han construido distintas colecciones en las competiciones TREC Genomics Track como I2B2, generado por Uzuner et al. (2006), Biocreative I y II creado por Hirschman et al. (2005) y CMC Medical NLP Challenge.

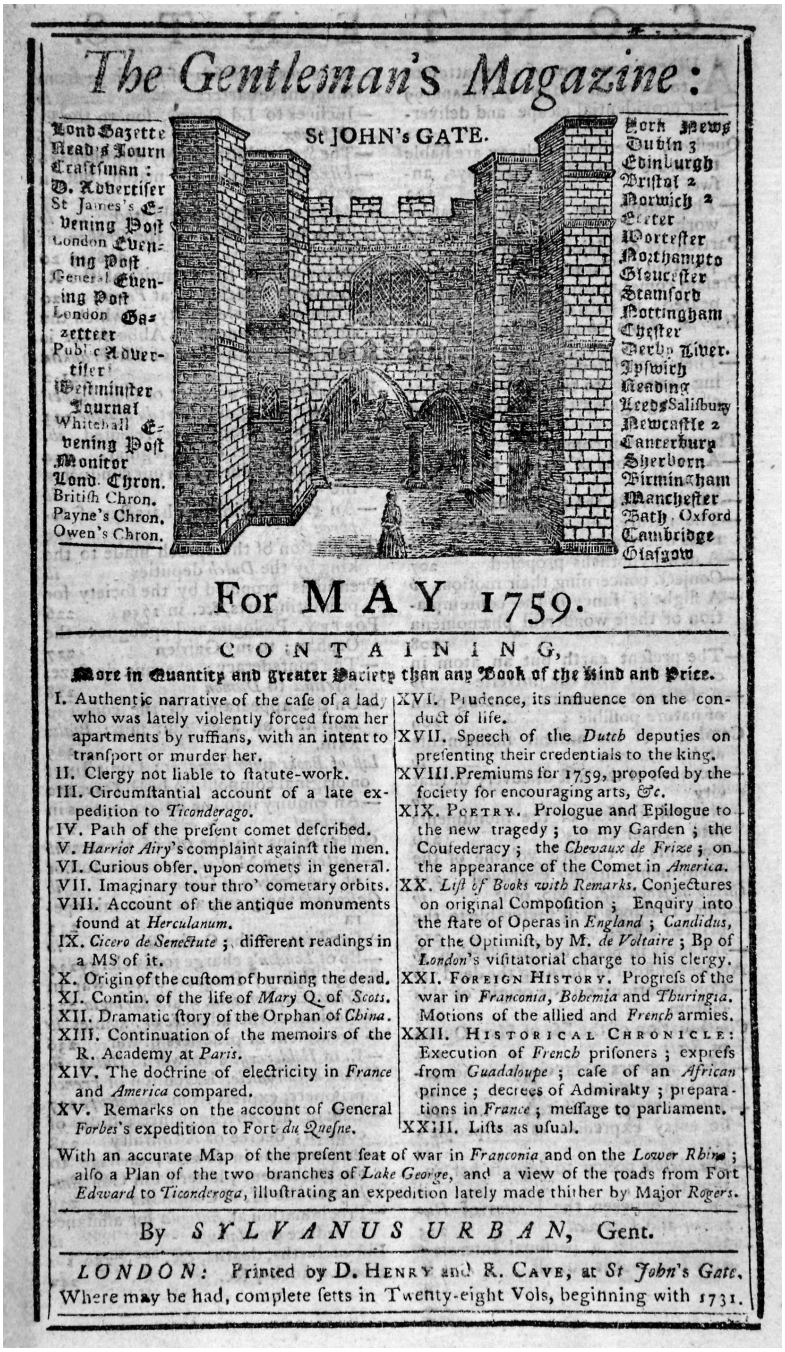
En el ámbito de la clasificación de texto con adversario, existe claramente un déficit de colecciones estándar en algunas tareas, mientras que proliferan en otras. La aplicación más abandonada en este sentido es el filtrado de contenidos Web, que no cuenta con ninguna colección estándar. Por el contrario, el filtrado de spam Web y el de correo basura cuentan con numerosas colecciones. No obstante, no existe actualmente ninguna colección que enlace las categorías de Amazon con las temáticas de DBPedia.

2.4 Marketing

El marketing que todos conocemos no es una técnica que se haya inventado recientemente. Sí es verdad que ha ido evolucionando desde la antigua Babilonia, donde se han encontrado tablillas de arcilla con inscripciones de un comerciante para la venta de sus productos, hasta convertirse en el paradigma que actualmente conocemos. Pero la historia del marketing se remonta hasta el siglo XIII, concretamente en torno al año 1450 cuando Johannes Gutenberg inventó la imprenta y distribuyó la biblia de 42 líneas de forma masiva. Ésta era la primera vez que se podían generar manuscritos e información de manera no manual y en poco tiempo, dando lugar a la publicidad impresa. Pocos años después, en torno a 1730 aparecieron las primeras revistas impresas como un nuevo medio de comunicación. La considerada primera revista de carácter general de la historia se publicó en Londres en 1931. En la figura 2.3 se muestra una de las portadas de 1759 de esta misma revista. El alcance que lograban estas publicaciones era notable y sobre 1870 aparecieron los primeros pósteres modernos con técnicas de litografía y generación en masa. Los pósteres dieron a los políticos de la época una nueva forma de llegar al pueblo de manera más llamativa, constante y barata. Cada partido político generaba sus anuncios y pósteres para después pegarlos por toda la ciudad y que el pueblo pudiera ver su mensaje y pudieran seguirlos. Esta práctica logró hacer llegar las ideas partidistas a mucha más gente que mediante el antiguo método de charlas en plazas. Los comercios rápidamente se dieron cuenta de que ellos también podían utilizar, para su propio beneficio, los mismos recursos que los políticos. Por ello, comenzaron a anunciarse por este nuevo canal de comunicación logrando mayores ingresos. Las vallas publicitarias no tardaron en llegar para anunciar productos y servicios. La saturación y popularidad de los anuncios impresos y pósteres se vio reflejada en la ley que, en 1839, tuvo que imponer el gobierno londinense para prohibirlos debido a que toda la ciudad se encontraba empapelada con ellos.

El siguiente hecho histórico que impulsó notablemente el marketing sucedió en 1922 con la emisión del primer anuncio radiofónico. El anuncio se emitió en Nueva York, costó sobre $50 y era sobre un complejo residencial construido por *Queensboro Corporation* en Jackson Heights. Hasta ese momento, la radio todavía no tenía un modelo viable y rentable para poder justificar la inversión necesaria en la creación de nuevos contenidos pero, con la emisión de este anuncio, se comenzó a vender "tiempo" en antena a los potenciales anunciantes. El crecimiento de la venta de aparatos radiofónicos creció de manera muy notable, llegando al punto de que, en 1933, más de la mitad de la población de los Estados Unidos de América tenía un aparato receptor de radio cuando 12 años antes la tasa era de 0.

Unos pocos años más tarde, en 1941 y con la aparición de la televisión, se emitió el primer anuncio publicitario televisado. En este caso el anuncio trataba sobre



The Gentleman's Magazine:

Lond Gazette
Read's Journ
Craftſman:
D. Advertiſer
St James's E-
vening Poſt
London Even-
ing Poſt
General Even-
ing Poſt
London Ga-
zetteer
Public Adver-
tiſer
Weſtminſter
Journal
Whitehall E-
vening Poſt
Monitor
Lond. Chron.
Britiſh Chron.
Payne's Chron.
Owen's Chron.

St JOHN's GATE.

York News
Dublin 3
Edinburgh
Briſtol 2
Norwich 2
Exeter
Worceſter
Northampto
Glouceſter
Stamford
Nottingham
Cheſter
Derby Liver.
Ipſwich
Reading
Leeds Saliſbury
Newcaſtle 2
Canterbury
Sherborn
Birmingham
Manchester
Bath Oxford
Cambridge
Glaſgow

For MAY 1759.

CONTAINING,

More in Quantity and greater Variety than any Book of the Kind and Price.

I. Authentic narrative of the caſe of a lady who was lately violently forced from her apartments by ruffians, with an intent to tranſport or murder her.
II. Clergy not liable to ſtatute-work.
III. Circumſtantial account of a late expedition to *Ticonderago*.
IV. Path of the preſent comet deſcribed.
V. *Harriot Airy*'s complaint againſt the men.
VI. Curious obſer. upon comets in general.
VII. Imaginary tour thro' cometary orbits.
VIII. Account of the antique monuments found at *Herculanum*.
IX. *Cicero de Senectute*; different readings in a MS of it.
X. Origin of the cuſtom of burning the dead.
XI. Contin. of the life of *Mary* Q. of *Scots*.
XII. Dramatic ſtory of the Orphan of *China*.
XIII. Continuation of the memoirs of the R. Academy at *Paris*.
XIV. The doctrine of electricity in *France* and *America* compared.
XV. Remarks on the account of General *Forbes*'s expedition to Fort *du Queſne*.
XVI. Prudence, its influence on the conduct of life.
XVII. Speech of the *Dutch* deputies on preſenting their credentials to the king.
XVIII. Premiums for 1759, propoſed by the ſociety for encouraging arts, &c.
XIX. POETRY. Prologue and Epilogue to the new tragedy; to my Garden; the Confederacy; the *Chevaux de Frize*; on the appearance of the Comet in *America*.
XX. *Liſt of Books with Remarks*. Conjectures on original Compoſition; Enquiry into the ſtate of Operas in *England*; *Candidus*, or the Optimiſt, by M. *de Voltaire*; Bp of *London*'s viſitatorial charge to his clergy.
XXI. FOREIGN HISTORY. Progreſs of the war in *Franconia*, *Bohemia* and *Thuringia*. Motions of the allied and *French* armies.
XXII. HISTORICAL CHRONICLE: Execution of *French* priſoners; expreſs from *Guadaloupe*; caſe of an *African* prince; decrees of Admiralty; preparations in *France*; meſſage to parliament.
XXIII. Liſts as uſual.

With an accurate Map of the preſent ſeat of war in *Franconia* and on the *Lower Rhine*; alſo a Plan of the two branches of *Lake George*, and a view of the roads from Fort *Edward* to *Ticonderoga*, illuſtrating an expedition lately made thither by Major *Rogers*.

By *SYLVANUS URBAN*, Gent.

LONDON: Printed by D. HENRY and R. CAVE, at *St John's Gate*.
Where may be had, complete ſetts in Twenty-eight Vols, beginning with 1731.

Figura 2.3: Portada de "*The Gentleman's Magazine*" de la publicación de Mayo de 1759. Primera revista de carácter general que contenía los primeros anuncios impresos.

relojes de la empresa Bulova. El anuncio costó cerca de los $9 y se emitió antes de un partido de béisbol, llegando a verse en unas 4000 televisiones. El impacto de la televisión en la sociedad fue muy alto y esto propició que en 1954 los ingresos obtenidos por publicidad en este medio superaran por primera vez a los ingresos por anuncios de radio y revistas juntos. Los siguientes años, la inversión en anuncios radiofónicos siguió cayendo mientras que la de televisión no hacía más que aumentar. El ecosistema estaba ya planteado y el líder como soporte publicitario principal era la prensa, seguido de cerca por la televisión. Pero en 1946 empezó a gestarse un nuevo jugador, el teléfono. El motivo fue que en ese año, el teléfono fijo ya se encontraba en más de la mitad de los hogares norteamericanos. Los primeros anunciantes de telemarketing comenzaron en la década de 1950. Durante 30 años, esta práctica se extendió y explotó hasta límites insospechados. El resultado fue que, a partir de la década de los 70, esta técnica empieza a perder eficacia. El marketing llamado como "*outbound marketing*" o marketing que interrumpía al consumidor insistiéndole en un producto sin escuchar las necesidades del usuario llegó a su punto de saturación, dejando de ser un medio tan rentable. El usuario ya no toleraba ese tipo de marketing tan bien como antes debido a que era molesto. Aun así, el telemarketing se sigue utilizando hoy en día debido su bajo coste.

En 1973 el Doctor Martin Cooper, que por aquel entonces era investigador de Motorola, realizó la primera llamada mediante un teléfono móvil. Este hecho, junto con la aparición de los primeros ordenadores personales por parte de IBM o Apple y el uso de Internet, comenzó a sentar las bases de lo que iba a ser el futuro del marketing más competitivo y beneficioso. Los nuevos avances en edición de imágenes ayudaron a que el marketing impreso resurgiera con más fuerza, haciendo que revistas o periódicos llegaran a tener ingresos por valor de 25.000 millones de dólares en 1985. Pero a pesar de este incremento, la televisión sustituye por primera vez a los periódicos como soporte publicitario principal incrementando notablemente sus ingresos. Por otra parte, los ordenadores e Internet se empezaron a explotar, no solo para la edición de imágenes, sino también para el envío de marketing mediante correos electrónicos. Había nacido el SPAM. El término SPAM fue acuñado para este tipo de correos electrónicos o emails no deseados y principalmente con contenido publicitario, por un *sketch* de 1970 de los Monty Python. En 1994 el bufete de abogados *Canter and Siegel* envío a miles de grupos de noticias un email publicando sus servicios. Este suceso fue uno de los primeros actos de SPAM. Hoy en día este medio se sigue utilizando, pero también ha sido regulado en muchos países debido a la cantidad de SPAM que se envía. En la figura 2.4 se muestran los 10 países que más SPAM envían del mundo[4].

Pero todavía ni se había empezado a arañar la superficie de las posibilidades que las nuevas tecnologías podían ofrecer al marketing. En 1995 el uso de Internet

[4]https://www.spamhaus.org/statistics/countries/

Figura 2.4: Ranking de paises enviadores de SPAM a 10 de Julio de 2015.

a nivel empresarial se estaba extendiendo en varios países, pero existía un grave problema, buscar información relevante o páginas Web que tuvieran información resultaba bastante difícil. Viendo esta necesidad, empresas como Yahoo!, Altavista o Ask.com lanzaron motores de búsqueda de contenidos web. Estos motores buscaban en miles de páginas web palabras clave o elementos descriptivos que tuvieran algún tipo de relación con los términos que el usuario estaba buscando. Rápidamente, el uso de estos sistemas se popularizó y surgieron conceptos como la optimización de motores de búsqueda, en inglés *Search Engine Optimization* (SEO) o marketing en motores de búsqueda, *Search Engine Marketing* (SEM) dando comienzo al marketing en buscadores web. En 1998 Brad Fitzpatrick, Evan Williams y Meg Hourihan crean el concepto de *blogging*. La base era crear blogs sobre cualquier temática y que cualquier usuario pudiera comentar lo que quisiera y compartirlo con el resto del mundo, llegando a existir más de 50 millones de blogs en el año 2006. La irrupción de esta tecnología supuso un incremento radical en el volumen de información existente hasta entonces en Internet. Gracias a que es una tecnología muy simple de entender, se ha hecho muy popular tanto entre usuarios normales, que los utilizan como bitácoras personales; como entre empresas, siendo una manera muy efectiva para promocionarse y fortalecer la imagen de sus marcas. También en 1998 aparecen los buscadores de Google y MSN. Google, tras el éxito inicial de su buscador, decide apostar por el marketing y los anuncios como área de negocio y, en el año 2000, crea la plataforma Google AdWords para publicar anuncios en los resultados de las búsquedas. Esta herramienta se convierte rápidamente en una gran fuente de ingresos, convirtiendo los anuncios en buscadores y el posicionamiento de los resultados de las búsquedas en elementos muy importantes para las empresas que querían obtener tráfico hacia sus sitios web. Cinco años más tarde, Google crea Google Analytics para monitorizar el tráfico hacia los sitios web y poder evaluar qué tipo de usuarios visitan los sitios web, durante cuánto tiempo, desde dónde y mucha más información que pudiera servir al dueño del sitio web a mejorar.

Además, entre 2003 y 2006 surgieron las redes sociales LinkedIn, MySpace, Facebook y Twitter. Estas nuevas plataformas orientadas principalmente a la comunicación entre usuarios, generaron y generan mucha información valiosa acerca de los hábitos, preferencias y gustos de los usuarios. Estos nuevos recursos para obtener más información, que el usuario pueda dar su opinión y que se tenga en cuenta, empezó a plantear un nuevo modelo de marketing, el "*inbound marketing*". Este nuevo modelo, a diferencia del "*outbound marketing*", se centra en no perseguir al cliente, sino que sea el cliente el que venga mediante la generación de recursos o contenidos que le aporten un valor especifico y, de esta manera, se conviertan en clientes. Un ejemplo de este modelo es Amazon, que en 2006 obtuvo 10.000 millones de dólares de beneficio subiendo a 25.000 millones en 2009. En esta plataforma se venden diferentes tipos de productos y cada producto tiene una opinión de los usuarios que

lo han comprado. Esta opinión sirve para que futuros posibles clientes tengan más información acerca de cómo es el producto con sus ventajas y desventajas, dándole un valor añadido al producto. Amazon es un buen ejemplo de lo que se conoce como comercio electrónico o *e-commerce*.

La opinión de los usuarios empieza a tener cada vez más importancia a la hora de ofrecer productos. Un claro ejemplo es que Google comienza a personalizar las búsquedas de los usuarios en base al historial de sus búsquedas. Además, con la explosión de la telefonía móvil y los *smartphones*, cada usuario compartía, de manera más sencilla y rápida, más opiniones sobre productos o servicios, aumentando también las necesidades de nuevas plataformas que le permitieran hacerlo. El usuario se convierte en un consumidor de información feroz, que cada vez demanda más información actualizada y en 2009 Google lanzo las búsquedas en tiempo real para cubrir esa necesidad.

La telefonía móvil también ha supuesto un antes y un después en la historia de Internet y del marketing. En 2007, existían más de 295 millones de personas que utilizaban las redes 3G y en el 2010 casi la totalidad de los hogares norteamericanos tenían telefonía móvil. Además, la media de uso de los jóvenes de Internet y televisión casi se asemejaban, siendo de 13,7 horas para Internet y de 13,6 para la televisión. Esta cantidad de usuarios conectados a Internet y que utilizan buscadores para encontrar información, ha generado que existan sitios web que intentan manipular los resultados de las búsquedas de manera fraudulenta. El motivo es obtener tráfico hacia sitios web específicos, aunque no tengan la información que el usuario esperaba. Para evitar estas tácticas, en el 2011 Google sacó Panda, una plataforma para mejorar la calidad de los resultados, penalizando sitios que no contenían información relevante. Su funcionamiento utiliza información de redes sociales y de Internet en general.

El modelo de "*inbound marketing*" ha demostrado ser una táctica muy eficiente y barata, más incluso que el "*outbound marketing*". Además, con los nuevos canales como el social media, en el cual, la interacción con el cliente es más personal y, teniendo en cuenta que la gran mayoría de ciudadanos de los países desarrollados tienen un perfil en alguna red social, el alcance y los beneficios que se obtienen son desorbitados. Por último, el portal de vídeos Youtube, que comparte beneficios con los usuarios que suben sus contenidos a esta plataforma, ha creado una nueva fórmula que genera ingresos para sí mismo y para los clientes a la vez que aporta un valor al resto de usuarios de Internet.

2.4.1 Tipos de publicidad

Estos avances no han hecho más que afianzar el modelo de "*inbound marketing*". Este modelo, se da especialmente en Internet. Se llama propaganda on-line, en inglés "*on-line advertisement*", y se dividen en 10 tipos según Nosrati et al. (2013).

2.4.1.1 Publicidad gráfica

Este tipo de publicidad es casi el más común de todos. Normalmente suele contener texto, logos, fotografías o cualquier tipo de imágenes. También suele contener mapas o elementos similares para que el usuario sepa exactamente qué se está ofertando. Estos anuncios aparecen en páginas de periódicos, revistas o en cualquier contenido editado (Anagnostopoulos et al., 2011). La publicidad gráfica, aunque su nombre dé a entender que es algo que debe contener imágenes, no tiene por qué. Los anuncios clasificados, también son de este tipo y suelen estar en otras secciones debido a que son algo más específicos, de ámbito más concreto y normalmente solo suelen contener texto. La publicidad de comienzos del siglo XX no poseía audio ni vídeo y no tenía por qué tener imágenes. Aun así, esta etiquetada como publicidad gráfica. Otro tipo de publicidad que también está dentro de esta categoría es la utilizada en los mensajes comerciales que se envían a los dispositivos móviles en formato SMS o email. La manera más normal de ver este tipo de anuncios es mediante vallas, pósteres o tarjetas.

2.4.1.2 Search Engine Marketing

Este tipo de publicidad (SEM), es un sistema de marketing online desarrollado específicamente para Internet. Uno de sus principales objetivos es la promoción de sitios web para aumentar su visibilidad en resultados de buscadores web, en inglés "*Search Engine Results Pages*" (SERPs), mediante la optimización y el marketing (Sherman, 2007), (Lin y Hung, 2009), (Jansen y Schuster, 2011). SEM utiliza optimizadores para motores de búsqueda, en inglés " *Search Engine Optimization*" (SEO) para mejorar, ajustar o reescribir el contenido de los sitios web y, de esta manera, mejorar el posicionamiento o ranking de la página en los resultados de los buscadores. Por otra parte, también utiliza sistemas que cuentan el número de clicks para lograr medir las ganancias de los sitios web (Sullivan, 2010). Algunas de las métricas que se utilizan para medir la optimización y éxito de los sitios web son; investigaciones y análisis referente a sitios web de palabras clave, en inglés "*keywords*", la saturación y popularidad de los sitios web, herramientas de análisis web tipo backend como "*web analytic*" o herramientas "*Whois*" para obtener información sobre el dominio.

2.4.1.3 **Optimización de los motores de búsqueda**

Este tipo de publicidad, en inglés *Search Engine Optimization* (SEO), es el proceso que afecta a la visibilidad de un sitio web en los resultados "naturales" de un buscador, sin necesidad de tener que pagar para mejorar el posicionamiento. El motivo por el cual este proceso es importante es porque los resultados que aparecen en primer lugar, o los que tienen mejor ranking o los que más veces aparecen en los resultados de un buscador, por regla general, suelen ser los que más visitas, usuarios y tráfico reciben. Además, la mayoría de los usuarios que visitan un sitio web, suelen venir redirigidos desde un buscador web. El SEO puede centrarse en diferentes tipos de búsquedas, según las necesidades u objetivos que se tengan. Por ejemplo, las búsquedas de imágenes o videos, de lugares cercanos a donde se encuentra el usuario, de noticias, de temáticas industriales o de documentos académicos (Beel et al., 2010).

2.4.1.4 **Social Media Marketing**

Este tipo de publicidad (SMM), está relacionada con la interacción de una empresa, mediante comunicados o mediante sistemas para mantener el contacto con sus clientes y poder ofrecerles soluciones a las cuestiones que planteen en plataformas de redes sociales. También se refiere al proceso de conseguir más tráfico o atención de los usuarios hacia un sitio web concreto mediante la interacción en plataformas de redes sociales (Trattner y Kappe, 2013). Una peculiaridad de este tipo de publicidad es que no se obtienen beneficios tangibles y directos, sino que se asemeja más al estilo del "boca-a-boca" y los beneficios que se obtienen son captaciones de clientes potenciales o mejora de la imagen de marca del sitio web.

2.4.1.5 **Email Marketing**

Este tipo de publicidad está basada en el envío directo de emails o correos electrónicos con contenido comercial a un grupo determinado de gente. Todo email enviado a un posible cliente potencial puede ser considerado como un email de marketing. Este tipo de anuncios enviados por email pueden ser de varios tipos como peticiones para crear negocios, solicitud de donaciones o ventas, o intentar mejorar la afinidad, confianza o imagen para una marca.

Las ventajas de este tipo de publicidad son diversas. Una de las más importantes es el gasto que conlleva realizar una campaña de marketing mediante email. Hoy en día, el coste por envío de email es casi cero y su alcance puede ser gigantesco. Además, existen sistemas para comprobar cuántos usuarios han visto el email y poder valorar el alcance e impacto de la campaña. En lo referente a gestionar las respuestas, los filtros que ofrecen los clientes de correo ayudan a enviar al destinatario correcto preguntas o notificaciones de anuncios que han tenido éxito. Otro factor positivo es

la gran aceptación que tiene el email dentro de los usuarios y el número de veces que se revisa para ver si existen nuevos emails (Lenhart et al., 2001). Este factor amplía el impacto sobre este tipo de publicidad.

Pero no todos son ventajas. El problema con este tipo de publicidad ocurre con el envío masivo y desmesurado de anuncios a usuarios. Este tipo de actividad está catalogada como SPAM y en varios países está catalogado como delito, por ejemplo Estados Unidos[5]. Esto ha llevado a que muchos de los correos que se envían sean catalogados como SPAM y a que más del 90 % del tráfico mundial de emails sea SPAM.

2.4.1.6 **Marketing de referencia**

Este tipo de publicidad está muy ligada a la técnica del boca-a-boca. Su base es la confianza depositada en el emisor del comunicado por parte del receptor. El mensaje que se envía puede contener aspectos positivos o negativos sobre un producto o marca, lo cual influirá en la decisión del receptor. Un ejemplo de este tipo de publicidad y de su utilidad y relevancia fue el estudio llevado a cabo por la Universidad Johann Wolfgang Goethe y la Universidad de Pennsylvania Schmitt et al. (2011) en 2010. En él, un banco pagaba $25 a sus clientes por traer a nuevos clientes. El estudio mostró que este tipo de nuevos clientes, llamados clientes con referencia, eran más rentables que los clientes logrados por métodos tradicionales. La retención era más larga, mejorando la fidelidad a la marca y la contribución monetaria era mayor, tanto a corto como a largo plazo. Por otra parte, según el estudio, el coste de traer a clientes con referencia a corto plazo no conllevaba un incremento de los gastos, pero a largo plazo aumentaba los beneficios de la empresa.

2.4.1.7 **Affiliate marketing**

Este tipo de publicidad está basada en recompensas por usuario captado. También se denomina como marketing contextual y está formado por 4 actores, *anunciador*, *editor*, *plataforma de anuncios* y *usuario*.

1. **Anunciador**. Este actor es, normalmente, una compañía que quiere anunciar sus productos o servicios y paga por ello.

2. **Editor**. Este actor es el responsable o propietario de un espacio web con contenidos en sintonía con una temática determinada o generalista que quiere tener beneficios mediante la publicación de anuncios.

[5]https://www.ftc.gov/

3. **Plataforma de Anuncios**. Este actor es el sistema que selecciona los anuncios de los anunciadores para ser publicados en los sitios Web de los editores en base a ciertos criterios y filtros. Además es el que cuantifica y decide el modelo de pagos por el cual se abonan los beneficios a los editores.

4. **Usuario**. Este actor es la persona que visita el sitio web del editor y que interactúa con los anuncios generando tráfico hacia los productos y relevancia del sitio web.

Además de los actores, existen varias formas de rentabilizar este tipo de publicidad. Las más importantes maneras de monetizar este entorno son 3 (Mahdian y Tomak, 2007), (Nazerzadeh et al., 2008).

1. Pagar por impresión, en inglés "*pay-per-impression*" (PPI), el anunciador paga por anuncios mostrados en una web.

2. Pagar por click, en inglés "*pay-per-click*" (PPC), el anunciador paga por cada click que los usuarios hayan hecho sobre su anuncio en un sitio web.

3. Pagar por acción "*pay-per-action*" (PPA), el anunciante paga por cada redirección a su sitio web (tienda, plataforma, etc.) realizado por un anuncio en un sitio web de un editor. Las acciones se negocian previamente, porque pueden ser suscripciones a sus servicios o cualquier otro tipo de acción.

Estos modelos de monetización no son los únicos y cada plataforma de anuncios las lleva a cabo de diferente manera, según sus capacidades, anunciantes y publicistas. Aunque muchas veces, este tipo de marketing es ignorado por los anunciantes Prussakov (2007), los motores de búsqueda, páginas web y demás plataformas de Internet suelen utilizarlos. El motivo es que suelen conllevar un bajo coste y gran beneficio.

2.4.1.8 **Marketing de contenido**

Este tipo de publicidad, en inglés *Content marketing*, presenta la información no como contenido promocional, sino como un contenido útil y relevante que aporte un valor añadido a los usuarios y para mejorar la visión por parte de los clientes de la empresa. Esta información puede estar en diferentes formatos, incluyendo noticias, videos, fotos, post en redes sociales, publicaciones científicas o corporativas, *whitepapers*, *e-books*, infografías, casos de estudio, pruebas de conceptos, manuales, *podcasts*, etc. (Rowley, 2008).

2.4.1.9 **Inbound marketing**

Este tipo de publicidad es la que actualmente más se está empezando a utilizar debido a su gran aceptación e impacto. Su objetivo es mejorar la imagen de marca mediante los sistemas digitales, tales como redes sociales, blogs, *podcast*, *e-books*, *whitepapers*, post, SEO, SMM, etc. Para lograrlo, intenta llamar la atención de los usuarios con campañas que mejoren la localización de los sitios web de la empresa y generando contenido que de un valor añadido a las publicaciones para que sean los usuarios los que quieran obtener más sin necesidad de que la empresa tenga que ofrecérselo.

La metodología que sigue tiene 3 estados: buscar, convertir y analizar. Además, esta metodología se ilustra en 5 estados o conceptos: atraer trafico web, convertir a los visitantes en clientes potenciales, convertir a los clientes potenciales en ventas, convertir a los clientes en mejores clientes y que repitan la experiencia de compra y analizar continuamente para mejorar el proceso. Con este proceso, y mediante otras técnicas de marketing y segmentación, se consigue dividir a los clientes en segmentos con diferentes niveles y que son asociados a grupos específicos de productos y/o marcas a las que son afines y que no pierden interés independientemente del tiempo que pase.

2.4.1.10 **Comunicaciones de marketing**

Este tipo de publicidad está basada en la comunicación entre la marca y el usuario o cliente. La comunicación son todos aquellos mensajes y contenidos multimedia que se usan para comunicarse con el mercado de clientes. La comunicación que se genera es la parte que proporciona la promoción de los productos, dando lugar a las 4 pes (4P) del marketing mix: lugar, precio, promoción y producto respectivamente. En inglés se denominan *Place*, *Price*, *Promotion* y *Product*.

Tradicionalmente, los profesionales encargados de la comunicación de marketing se centraban únicamente en la creación y difusión del material impreso de marketing. Sin embargo, las tendencias han demostrado que el uso de elementos estratégicos de "*branding*" y marketing, con el fin de garantizar la coherencia de la entrega de mensajes en toda la organización, un "*look & feel*" consistente da mejores resultados.

Muchas tendencias que se dan en las empresas pueden atribuirse a las comunicaciones de marketing. Por ejemplo la transición que ha existido en la atención al cliente. La comunicación de marketing es un punto crucial de las relaciones que se establecen con los clientes y con la visión, impacto y afinidad que se tiene de la empresa.

2.4.2 Técnicas contextuales para anuncios web

El objetivo del marketing es generar, de alguna manera, una diferencia visible de un producto y que esté asociado a unas ganancias. De los tipos presentados en el punto anterior, el que más se utiliza es el de "*affiliated marketing*" o marketing contextual. Normalmente, y como previamente se ha comentado, una de las formas más utilizadas para generar dinero con los anuncios en sitios web de Internet es mediante los clicks de los usuarios. Esta manera de monetizar el espacio que se deja a los anuncios en las páginas web no siempre satisface a los editores, los cuales siempre están buscando nuevas fórmulas o técnicas para mejorar y aumentar sus beneficios.

Uno de los principales modelos que se usan es ofrecer al usuario que visita su web productos afines al contenido que se le está mostrando. El problema surge cuando existen millones de anuncios y hay que elegir entre ellos para seleccionar el que más impacto o relevancia tenga con el contenido. Para llevar a cabo la tarea de unir anuncios con páginas web, existen dos técnicas principales. La primera está basada en la búsqueda de palabras claves en el contenido web y en los anuncios, y la segunda en la generación de vectores por cada anuncio y contenido web y su posterior comparación. Esta última técnica está basada en técnicas que utilizan MSV.

2.4.2.1 Modelo de palabras clave

El modelo de palabras claves, en inglés *keywords*, se basa en la extracción de palabras o frases del contenido de las páginas web. Con estos elementos obtenidos, se buscan anuncios o grupos de anuncios que estén en consonancia. El proceso es buscar el mayor número de coincidencias en los grupos de anuncios que previamente han sido etiquetados. La etiquetación suele basarse en temáticas especificas según el objetivo que se quiere conseguir con los anuncios, el producto que ofertan, el nombre de la empresa, el nombre del producto, el público objetivo, etc. En esta área, existen varios trabajos de investigación. Langheinrich et al. (1999) propuso un sistema llamado *ADWIZ* que era capaz de adaptar los anuncios on-line, según términos cortos de intereses de un usuario de manera no intrusiva. *ADWIZ* no utilizaba directamente el contenido del sitio web visitado por el usuario, se basaba en las palabras clave utilizadas por el usuario en el motor de búsqueda y en la dirección web que había seleccionado el usuario. Turney (2000) propuso el sistema *GenEx* para la extracción de *keywords* y utilizarlos, principalmente, en la etiquetación de documentos científicos o cualquier otro tipo de recurso textual. Carrasco et al. (2003) propuso un método basado en agrupaciones que aglutinaba las *keywords* de manera gráfica para ofrecer palabras clave e identificar grupos de anuncios para el usuario. Yih et al. (2006) propuso un sistema para extraer *keywords* de páginas no visitadas ni etiquetadas en base a características previamente definidas junto con la frecuencia de términos, en inglés *Term Frecuency* (TF) de las palabras clave. Además el sistema podía entrenar

con estas características para mejorar. Jang et al. (2007) estableció una relación entre diferentes *keywords* en ontologías y anuncios. El proceso era utilizar un módulo para la extracción de *keywords*, tanto de páginas web como de anuncios, y después utilizar el algoritmo Apriori[6] para determinar las relaciones entre las diversas palabras claves. Liu et al. (2014a) usó un método para la extracción y sugerencia de *keywords* utilizando el análisis del discurso, en inglés *Part-of-Speech* (POS), y el reconocimiento de entidades nombradas, en inglés *named-entities-recognition* (NER). Khan et al. (2009) recuperaba las *keywords* óptimas y más relevantes de las páginas web mediante la técnica de frecuencia de términos - frecuencia inversa de términos, en inglés *Term Frecuency – Inverse Term Frecuency* (TF-IDF). Además, también utilizaba palabras clave obtenidas de diferentes partes del contenido de los anuncios para comprobar su afinidad mediante una función de búsqueda. El resultado que se obtenía era el anuncio más relevante en base al contenido de la página web seleccionada. Liu et al. (2010) crearon una red textual para una única página web y, mediante el uso del algoritmo *PageRank*, podían determinar las *keywords* más relevantes de esa web. Dave y Varma (2010) propusieron un sistema que usaba métodos de fragmentación de texto y, para el entrenamiento, clasificadores de Naïve Bayes con páginas web etiquetadas con *keywords* de anuncios. El sistema era capaz de determinar *keywords* para nuevas páginas web que no habían sido procesadas. Ribeiro-Neto et al. (2005) estudió el problema de la asociación entre los anuncios y las páginas web. Este tipo de modelo, en ciencias de la computación, se llama publicidad orientada al contenido y asume que, para el análisis, se tiene acceso a todo el contenido textual de un sitio web, a las *keywords* declaradas por el anunciante y al texto que asocia el sitio web con el ámbito de negocio del anunciante. Como resultado se obtuvo que los métodos o estrategias de búsqueda que utilizaban o tenían en cuenta los problemas semánticos, eran más efectivos. Estos resultados se basaban en las métricas de media de *precision* y, en comparación con las estrategias basadas en vectores triviales, eran un 60 % mejores.

2.4.2.2 **Modelo de comparación de vectores**

El modelo de comparación de vectores está basado en modelos de MSV. Este tipo de modelo algebraico se utiliza para representar el texto de los documentos, o los propios documentos, como vectores en base a un índice de términos extraídos de los datos de los propios documentos.

Uno de los primeros trabajos de investigación que utilizaban el enfoque de MSV con el contenido de páginas web y anuncios, fue presentado por Ribeiro-Neto et al. (2005). En él, se utilizaba MSV para representar las páginas web y los anuncios. También se remarca el problema de que el vocabulario o las palabras que se utilizan

[6]https://en.wikipedia.org/wiki/Apriori_algorithm

para generar el contenido de las web, tiene una correlación muy débil con el vocabulario o las palabras que forman el contenido de los anuncios. A este problema, los autores lo nombraron como la impedancia del vocabulario, en inglés *the vocabulary impedance*. Para intentar solucionarlo, propusieron un enfoque basado en la expansión del contenido de la web y de las *keywords* con el contenido de páginas similares y sus *keywords*. A este enfoque lo llamaron *impedance coupling strategy*. Broder et al. (2007) propusieron un enfoque semántico para los anuncios contextuales. Los autores se centraron en las *keywords* ambiguas (homonimia y polisemia) y en el contenido ambiguo de los sitios web (discrepancia en el contexto). La solución que propusieron fue un sistema automático de clasificación para sitios web y anuncios que, utilizando una taxonomía determinada, calculaba la similitud entre el contenido de las páginas web y el de los anuncios. Los valores de similitud de la taxonomía estaban basados en el coseno de los vectores. La mejora que introdujeron fue de un 25 % de incremento en la media de la métrica *recall* para los modelos anteriores basados en la distancia del coseno. La información obtenida del proceso de clasificación ayudó a filtrar anuncios irrelevantes para la web seleccionada, mejorando el rendimiento de selección. Los autores utilizaron una ontología de ámbito comercial, creada específicamente para anuncios, para clasificar los sitios web y los anuncios. Esta ontología representaba una estructura jerárquica de peticiones para anuncios (en base a qué contenido, se ofrecían diferentes anuncios) y estaba constituida por unos 6000 nodos. Lee et al. (2013) establecieron una taxonomía jerárquica de topics para clasificar páginas web y anuncios. Además, clasificaron los anuncios en base a la relevancia de los topics que contenían. Anagnostopoulos et al. (2007) propusieron un enfoque, tipo "*Just-in-time*", para anuncios contextuales en base a las *keywords* obtenidas y a la clasificación de características, tanto de los sitios web como de los anuncios. Ellos utilizaron estas propiedades para clasificar las propiedades y generar vectores que representasen las palabras para, finalmente, calcular el valor de similitud entre el contenido de las páginas web y el de los anuncios. En un trabajo posterior, Anagnostopoulos et al. (2011) propusieron otro método para sugerir anuncios de manera más rápida. El método consistía en utilizar una búsqueda semántica y sintáctica y en resumir el contenido de la página web. Con este enfoque de utilizar un resumen en vez del contenido completo, el tráfico de red entre la plataforma de anuncios y la web se optimizó, perdiendo únicamente entre un 1 % y un 3 % de anuncios relevantes. Fan y Chang (2011) utilizaron MSV para evaluar la similitud entre páginas web y anuncios. Su método consistía en preprocesar los términos del contenido del sitio web y los de los anuncios con un sistema de detección de sentimientos. Una vez preprocesados, utilizando funciones de similitud del coseno, realizar búsquedas con los términos. Pak y Chung (2010) propusieron un enfoque que utilizaba Wikipedia para mejorar la métrica de *precision* en los anuncios contextuales. El proceso era utilizar los artículos de Wikipedia, el contenido de los sitios web y el de los anuncios para

generar vectores. Después de esto, se calculaba la distancia del coseno para evaluar la similitud entre cada artículo de Wikipedia, cada página web y cada anuncio. Chen et al. (2015) propusieron un método de anotaciones para anuncios contextuales para unir anuncios multimedia y páginas web. El método utilizaba la representación de las páginas web como vectores y los anuncios eran representados como un conjunto pequeño de etiquetas de *keywords*, previamente etiquetadas por expertos y extraídas de los textos de los propios anuncios.

“Todo el mundo hace locuras en los anuncios, como comer en un burguer.”

Marjorie Bouvier Simpson
(1957 –)

3

Base de conocimiento

Que una máquina o sistema creado por el ser humano pueda poseer toda la información sobre todos los elementos conocidos es posible pero, actualmente, no dispone de las técnicas necesarias para unir los diferentes conocimientos de manera útil e inequívoca.

Un sistema que si lo logra de forma natural y, normalmente, con gran acierto, es el cerebro humano. Éste, consigue asociar diferentes comunicaciones, como frases o conversaciones, con diferentes aspectos del entorno. Esto lo consigue basándose en la experiencia y la transmisión del conocimiento.

Para lograr una empresa tan difícil como generar una metodología para la unión entre la información y el conocimiento a un sistema y que, además, pueda generar unos resultados con un índice de acierto deseable, es necesario obtener antes una base de conocimiento con información útil, etiquetada y relacionada entre sí.

Desde antaño, se dice que la información es poder. Es por este motivo que en el presente trabajo, la información juega un gran papel. Esto se debe a que es necesaria una referencia sobre algún aspecto remarcable de la conversación para poder detectar el contexto, el objetivo y el significado de ésta.

Pero, realmente, no toda la información es importante. Es necesario saber separar la “paja del grano” y lograr una información que nos sea útil. La utilidad de la información la medimos en base a la facilidad con la que podemos obtenerla, y a la manejabilidad con la que podamos procesarla y gestionarla. Además, también entra en escena la definición de un tipo de metodología para poder actualizar, de manera sencilla, esta ingente cantidad de información que es necesaria para llevar a cabo

el objetivo de dar contexto a las frases cortas. El motivo por el cual es necesario un sistema para su actualización, se debe a que, el contexto de las frases, cambia según modas, épocas, tecnologías, hechos históricos o, simplemente por evolución del lenguaje.

La razón por la que utilizamos la extracción de contextos en frases en vez de únicamente las palabras de las frases, se debe a querer utilizar un enfoque distinto al más utilizado actualmente. Además, este enfoque abarca más elementos que únicamente las palabras. Un contexto puede contener más relaciones que una única palabra, posibilitando que surjan más relaciones para comprender lo que los interlocutores o usuarios están tratando.

Por otra parte, una vez que podamos gestionar el contexto de una frase, necesitamos poder ofrecer algo al usuario. Por este motivo, nuestra base de conocimiento tendrá también productos etiquetados, siguiendo la misma metodología que para la base de conocimiento de contexto.

Lo que hemos querido plantear en este trabajo es la utilización de contextos para ofrecer productos. Estos contextos deben estar definidos previamente para poder realizar una unión entre los diferentes contextos detectados y los productos asociados a dicho contexto. Aunque nuestro trabajo se centra en determinar los métodos idóneos para ofrecer el producto que más posibilidades tiene de venderse y el contexto que más coincide con la frase, es necesaria una categorización de los diferentes productos y contextos para poder determinar esta asociación. Por lo que también puede verse como un sistema y metodologias de un sistema de categorización.

Con todas estas ideas en mente, se ha creado una base de conocimiento basada en clases y contenidos ligados a dichas clases que será descrita a continuación. Por otra parte, cabe mencionar que este recurso esta publicado y accesible para cualquier persona que quiera hacer uso de ella en la web del autor[1].

3.1 Obtención de datos

La acción humana de la comunicación parte del instante en el cual dos o más individuos interactúan mediante algún tipo de canal con ciertas reglas. Normalmente la comunicación se suele hacer sobre temas que ambos conocen en mayor o menor medida y que ambos entienden.

Cada individuo almacena en su memoria información útil para poder determinar el contexto de la conversación, asociarlo a otros temas o contextos y, de esta manera, poder mantener una conversación, convirtiendo dicha información en conocimiento.

La adquisición de este conocimiento en máquinas requiere de relaciones muy

[1] paginaspersonales.deusto.es/patxigg/

claras y detalladas entre los temas y los elementos que contienen dichos temas.

Para lograr una base de conocimiento como la que hemos generado, ha sido necesario pasar antes por varias iteraciones, los cuales mostraron el camino a seguir para lograr el objetivo.

3.2 Creación de la base de conocimiento

La creación de una base de conocimiento útil y significativa ha sido y es uno de los principales objetivos de este trabajo. Debido principalmente a la gran cantidad de información que se genera al día en Internet. La información se modifica, el lenguaje evoluciona y, por tanto, este conocimiento no puede quedarse atrás. Se tiene que adaptar a las nuevas modas, costumbres, tecnologías y uso del lenguaje. Para lograr este objetivo, se ha diseñado una metodología específica para que una vez obtenidas las fuentes, el proceso de crearlas fuese automático. Esto se debe a la gran cantidad de datos que se manejan.

En este apartado, se explicará qué herramientas se han utilizado y de qué manera, así como los métodos seguidos para limpiar los datos y almacenar la información generada.

También se detallarán los tipos de pasos realizados para la construcción de nuestra base de conocimiento en cada uno de los experimentos y, por último, se describirá la metodología final que se ha elegido y que se utiliza para actualizar el conjunto de bases de conocimiento que tenemos.

3.2.1 Análisis de las herramientas de categorización de textos

En este apartado analizaremos algunas de las herramientas más interesantes para solucionar el problema de la categorización de textos.

Mostraremos las bondades y las debilidades de cada una de ellas con el fin de justificar la elección de la herramienta que hemos determinado más adecuada.

Actualmente, existen varias herramientas para la categorización de textos. Algunas de ellas son Lucene y Lucene Solr (ambas de Apache Fundation y descritas en el trabajo de Smiley y Pugh (2009)), Sphinx de Sphinx Technologies Inc[2], PostgreSQL de Postgresql.org y detallado en el trabajo de Momjian (2001) y MySQl de Oracle que se detalla en el trabajo de Widenius y Axmark (2002). Las dos últimas herramientas son bases de datos bien conocidas que, para este aspecto de categorización de textos, deben ser configuradas para la búsqueda de texto completo, es decir, la

[2] sphinxsearch.com/about/company/

obtención de textos completos en base a ciertas variables de búsqueda.

Los criterios que buscábamos para determinar la elección de una herramienta u otra fueron:

- Relevancia del resultado ofrecido y del ranking obtenido.
- Velocidad de indexación y de búsqueda.
- Facilidad de uso e integración con diferentes herramientas.
- Consumos de CPU y memoria RAM para funcionar.
- Escalabilidad.

3.2.1.1 Herramientas de bases de datos

Las bases de datos han sido y son una herramienta muy utilizada para almacenar y recuperar información sobre ciertos criterios establecidos. Su desarrollo para almacenar datos ha crecido con el avance de la tecnología y, hoy en día, son capaces de almacenar grandes cantidades de datos y obtener información de ellas de manera muy rápida. Aun con todos estos avances, la indexación de texto completo y su recuperación es un apartado que no contemplaban en sus inicios, por lo que han tenido que desarrollar nuevas opciones para satisfacer este tipo de búsquedas. Este tipo de configuración se llama *Full Text Search*.

PostgreSQL

La herramienta PostgreSQL cuenta con una configuración específica para la indexación de textos completos. Según su documentación, permite indexar textos en índices y tablas, parsear documentos completos y consultas, o como más se conocen, querys de búsqueda, devolver resultados según su importancia o relevancia con los parámetros de búsqueda, eliminar stopwords de los documentos y de las querys, utilizar diccionarios propios, de sinónimos, de *snowball* y morfológicos entre otros. *Snowball*[3] es un tipo de lenguaje de procesamiento para cadenas de texto pequeñas diseñado para crear algoritmos de *stemming*[4] y se utilizan en la recuperación de información. Con el uso de diccionarios de sinónimos, también permite la utilización de tesauros para mejorar los resultados. Un tesauro es la lista de palabras o términos empleados para representar conceptos. Además, contiene las características específicas de una base de datos tradicional como *triggers* (disparadores según eventos que

[3] snowballstem.org

[4] Método para reducir una palabra a su raíz o a un lema.

suceden), para las actualizaciones, obtención de estadísticas de los documentos y la manipulación de documentos y de querys.

Las restricciones que tiene esta herramienta son:

- La longitud de cada lexema debe ser inferior a 2KB.
- La longitud de un *tsvector* (lexemas + posiciones) debe ser menor que 1 MB.
- El número de lexemas debe ser menor que 2^{64}.
- Los valores de posición en *tsvector* debe ser mayor que 0 y no más de 16.383.
- No se admiten más de 256 posiciones por lexema.
- El número de nodos (lexemas + operadores) en un *tsquery* debe ser inferior a 32.768.

MySQL

La otra herramienta de base de datos que se ha comentado, MySQL, soporta indexación y búsqueda de texto completo mediante la generación de un índice especial de tipo FULLTEXT. Los índices FULLTEXT pueden usarse sólo con tablas MyISAM, pueden ser creados desde columnas CHAR, VARCHAR, o TEXT como parte de un comando CREATE TABLE o añadidos posteriormente usando ALTER TABLE o CREATE INDEX. Para conjuntos de datos grandes, es mucho más rápido cargar los datos en una tabla que no tenga índice FULLTEXT y crear el índice posteriormente.MySQL soporta expansión de consultas (en particular, su variante “expansión de consultas ciega”). Generalmente, esto es útil cuando una frase buscada es demasiado corta, lo que a menudo significa que el usuario se fía de conocimiento implícito que normalmente no tiene el motor de búsqueda *full-text*. Por ejemplo, un usuario buscando “database” puede referirse a “MySQL”, “Oracle”, “DB22”, y “RDBMS” todas son frases que deberían coincidir con “databases” y deberían retornarse también. Este es conocimiento implícito. El uso de operadores de proximidad está soportado, al igual que el de “always-index words” que pueden ser cualquier cadena de caracteres que quiera el usuario para tratarlas como palabras, tales como “C++”, “AS/400”, o “TCP/IP”. También soporta la búsqueda *full-text* en tablas MERGE, el *stemming* y el uso de stopwords, entre otras características.

Las restricciones que tiene esta herramienta, como ya se ha comentado, son:

- Las búsquedas *full-text* sólo las soportan las tablas MyISAM.
- Las búsquedas *full-text* pueden usarse con la mayoría de conjuntos de caracteres *multi-byte* con la excepción de Unicode.

- El conjunto de caracteres utf8 puede usarse, pero no el conjunto ucs2.
- Los idiomas ideográficos, como Chino y Japonés, plantean un problema al no tener delimitadores de palabras. El parser FULLTEXT no puede determinar dónde empiezan y dónde acaban las palabras, aunque para este problema existen algunas soluciones.
- Mientras el uso de múltiples conjuntos de caracteres en una misma tabla se soporta, todas las columnas en un índice FULLTEXT deben usar el mismo conjunto de caracteres y colación.
- La lista de columnas MATCH debe coincidir exactamente con la lista de columnas en algún índice FULLTEXT definido en la tabla a no ser que MATCH estén IN BOOLEAN MODE.
- El argumento de AGAINST debe ser una cadena constante.

3.2.1.2 **Herramientas específicas de recuperación de información**

Tras analizar las herramientas basadas en bases de datos, se observó que el rendimiento y los servicios necesarios para poder utilizar dichas herramientas eran elevados, utilizando gran cantidad de la memoria RAM y la CPU. Por este motivo, abandonamos este tipo de tecnología para emplear herramientas específicamente desarrolladas para la indexación, el análisis, la categorización y el procesamiento de textos. A continuación revisaremos las más importantes y los análisis realizados.

Sphinx

La herramienta Sphinx es una solución desarrollada desde cero en el lenguaje C++. Especialmente pensada para mejorar el rendimiento, la calidad de las búsquedas y la integración con otros sistemas. Es una herramienta que permite hacer búsquedas sobre elementos SQL y no SQL, ya sean índices propios de la herramienta o no, de manera rápida y sencilla. También puede estar instalado en un servidor como un servicio de búsquedas, tipo base de datos. Una peculiaridad de Sphinx es que permite configuraciones acordes a nuestro hardware disponible. Las búsquedas en este motor son simples y solo son necesarias unas líneas de código para poder realizarlas. Otras bondades de esta herramienta son su escalabilidad, el uso de elementos comunes de tratamiento de textos ya mencionados en la herramienta PostgreSQL y MySQL (estas características no son tablas ni elementos de bases de datos, son solo sobre el tratamiento de textos, tales como los diccionarios, *stemming*, stopwords, etc.), la mejora de algoritmos para la categorización de textos y su configuración para obtener resultados significantes y la posibilidad de ser un sistema distribuido.

Esta herramienta es de las más completas para la categorización de texto que existen. Su rendimiento es muy bueno y la posibilidad de configurarlo para dispositivos con pocos recursos hardware la hacen muy potente. Pero tiene el gran inconveniente de que, sea cual sea el sistema operativo en el que se quiera desplegar, hace falta una instalación.

Lucene Solr

La herramienta Lucene Solr es una solución muy utilizada en diversas empresas para indexar, analizar, categorizar y ofrecer resultados según el criterio de búsqueda deseado. Está escrito en Java y posee todas las características de un motor de indexación, junto con la integración con bases de datos. Una peculiaridad de Solr es que es altamente confiable, escalable y tolerante a fallos, proporcionando indexación distribuida, replicación y carga equilibrada por cada consulta, automatización y recuperación de fallos, configuración centralizada, funciones de navegación por documentos y resultados, soporte para servicios Web, JSON, XML y REST, independización de servidor de aplicaciones, fácilmente configurable y usable desde otros sistemas independientemente del lenguaje de programación mediante plugins. Estas son algunas de las características de esta herramienta tan potente.

Además, junto con Sphinx, es de las más completas del mercado de categorización de textos. Su potencia y eficacia está demostrada en múltiples proyectos, empresas e investigaciones. El problema de Lucene Solr es que es una aplicación con interfaz y muchos elementos que no siempre son necesarios, también tiene el inconveniente de que es necesario un contenedor de aplicaciones para su utilización, aunque sea de manera portable.

Lucene

Por último, encontramos la herramienta Lucene, en la cual está basada Lucene Solr. Esta solución tiene todas las bondades de Lucene Solr y Sphinx. Está escrita en Java, es muy ligera ya que solo son librerías, el consumo de recursos hardware es mínimo y es completamente portable, puede ir en cualquier proyecto. La velocidad de indexación y recuperación de datos es muy alta, las actualizaciones incrementales no llevan mucho tiempo, la escalabilidad es altísima, las características, configuraciones y funcionalidades son muy amplias y, además de todo esto, posee una gran comunidad activa de desarrolladores. En el ámbito de la experimentación e investigación, Lucene ofrece soporte y herramientas muy útiles para perfeccionar los sistemas.

Esta herramienta contiene lo mejor de Lucene Solr debido a que es su motor, es completamente portable y su consumo de recursos puede ser ampliamente configurable. El motor de indexación, análisis y procesamiento de Lucene tiene una rele-

vancia muy importante en el ámbito de las soluciones de categorización de textos debido a su inigualable rendimiento. El problema de este sistema es la integración de Lucene y su comprensión (el aprendizaje es difícil al principio, pero después es sencillo su uso) y, por último, al estar escrito en Java, es necesario utilizar Java para los proyectos o herramientas que permitan la comunicación entre Java y el lenguaje seleccionado.

3.2.1.3 **Conclusiones**

Como se podía intuir, las herramientas específicamente desarrolladas para la indexación, el análisis, la categorización y el procesamiento de textos son mucho más eficaces a la hora de indexar y obtener resultados en base a las consultas realizadas. Además, su nivel de optimización supone una menor utilización de recursos hardware que las soluciones basadas en bases de datos. Es por esto que las soluciones de bases de datos fueron finalmente desechadas.

Por otra parte, debido a la necesidad de una instalación en un sistema operativo concreto y la necesidad de un servidor y servicios activos, se desecharon las opciones de Sphinx y Lucene Solr, quedándonos con Lucene por su portabilidad, sus amplias opciones de configuración y escalabilidad, la calidad y velocidad de los resultados a la hora de indexar y procesar los textos y por la integración que posee con proyectos ya conocidos. Además, si hubiera existido la necesidad de una integración con otras herramientas no hubiera sido un problema remarcable ya que existen herramientas que nos permitirán portar el código que hubiéramos generado a otro lenguaje diferente de Java.

3.2.2 **Enfoque 1, Diccionario de la RAE**

En el enfoque 1, se utilizó como fuente de datos el diccionario de la RAE. La premisa era que el mejor recurso para dar la definición o definiciones más exactas a las palabras con las que cuenta una frase, y de esta manera obtener su temática, era el propio diccionario de dicha lengua, el cual además, se actualizaba con los nuevos términos que se creaban y era el elemento de referencia para las palabras del idioma castellano.

En este punto de la investigación, todavía no se tenía experiencia con este tipo de fuentes ni con el campo en sí, por lo que pareció una idea factible y un punto de inicio correcto para comenzar con el trabajo.

La metodología que se planteo fue la que se muestra en la figura 3.1. Lo primero que se hizo fue analizar los diferentes diccionarios online que existían. De estos diccionarios escogimos el de la RAE, no porque fuese más fácil la extracción de datos

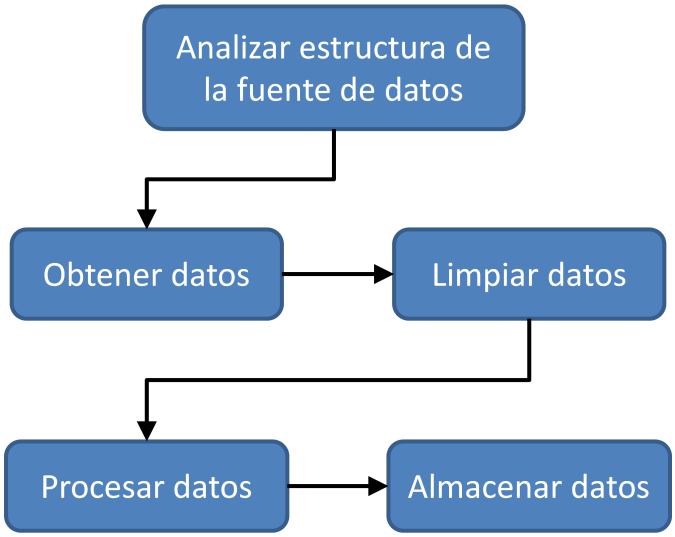


Figura 3.1: Metodología del experimento 1

ni porque fuese la Web más eficiente ni rápida, sino porque es la entidad oficial que da definiciones a las palabras del idioma castellano.

Una vez seleccionada la fuente de datos, se vio que la Web de la RAE no facilitaba un mapa o mostraba todas las palabas que contenía. Solo ofrecía definiciones a palabras que se le solicitaban. Este contratiempo obligó a obtener previamente un listado de palabras para hacer las búsquedas en la plataforma Web. Finalmente, el listado de las palabras se obtuvo de un CD de la RAE, del cual se extrajeron todas las palabras que contenía para después obtener la definición de dichas palabras desde la Web.

excursión.

(Del lat. *excursĭo, -ōnis*).

1. f. correría (‖ hostilidad de guerra contra un país).

2. f. Ida a alguna ciudad, museo o lugar para estudio, recreo o ejercicio físico.

Figura 3.2: Datos de la Web de la RAE sobre la definición de excursión.

Una vez obtenido el listado de palabras, se comenzó a obtener las definiciones de la aplicación Web. La extracción se realizó mediante una aplicación desarrollada

en Java y con las librerías de JSoup[5]. Esta librería permite recorrer el código HTML de manera sencilla, obteniendo el texto con y sin etiquetas del propio código HTML.

Los elementos que se han utilizado son los que se muestran en la figura 3.2. En color azul están las definiciones. Este elemento no era necesario porque es la palabra a definir. En color verde están las raíces de la palabra, tampoco era necesario. Por último, están las de color negro. Esta información es la definición de la palabra que se buscaba y es lo se necesitaba para la base de conocimiento. Esta información se almacenaba por cada palabra de la lista que existía en la Web de la RAE.

Para obtener estos datos, se creó un sistema capaz de descargar, de la aplicación que la RAE pone a disposición de los usuarios en Internet para obtener definiciones, cada una de las palabras de la lista que se había creado. Los elementos se clasificaron por orden alfabético y cada resultado obtenido era un elemento de la base de conocimiento.

En total se obtuvieron más de 79.000 términos con sus definiciones, los cuales fueron almacenados. Esta obtención de datos duró más de 4 meses debido al mal funcionamiento de la aplicación Web de la RAE, la cual obligó a modificar en varias ocasiones la cantidad de términos a buscar en un determinado tiempo.

Una vez obtenidos los datos, surgió el problema del almacenamiento. En el apartado 3.2.1 ya se ha mostrado cual ha sido la elección para este trabajo, así como los motivos, quedando por comentar ahora la implementación.

El almacenamiento fue un aspecto complejo debido a los elementos que eran necesarios indexar. En este experimento, las palabras que se definían, fueron tomadas por elementos en los cuales no se debía buscar. A este tipo de elementos se les llama invariables dentro del índice. Por otra parte, estaban las definiciones, las cuales eran los valores en los cuales efectuábamos las búsquedas.

Para almacenar los diferentes elementos, se implementó otra aplicación, también desarrollada en Java. Esta aplicación obtenía cada elemento con sus definiciones, obtenía su nombre y la definición y almacenaba dichos elementos. La palabra como elemento invariable y la definición como elemento sobre el que buscar. Estas versiones de índices fueron creados con la versión 3.4 de Lucene. Los temas que serían los resultados, se decidió que fueran las propias palabras.

Para realizar las pruebas, se utilizaron conversaciones obtenidas de usuarios reales, eliminando sus nombres y referencias personales para garantizar su privacidad. También se utilizaron pasajes de libros, extrayendo unicamente las conversaciones que se diferenciaban claramente y que no tenían ningún tipo de relato o interrupción entre ellas que hiciese cambiar el contexto de la conversación.

Aunque la valoración de los resultados fue evaluada por una sola persona que

[5] jsoup.org

realizo a mano toda la lectura de las conversaciones y sus etiquetas generadas, los resultados que se obtuvieron no fueron nada buenos. Los resultados no coincidían en casi nada, por no decir en nada, con el objetivo o temática de la conversación. El motivo era que el sistema no tenía en cuenta elementos importantes de la conversación como las palabras significativas debido a que estas, no aparecían en las definiciones que se habían almacenado. La razón es que dentro de las definiciones, la propia palabra a definir no suele estar escrita, si no, no sería una definición.

Teniendo en cuenta este problema, se pensó en introducir el propio elemento a definir dentro de la definición para ir probando si, de esta manera, se mejoraban los resultados.

Los resultados mejoraron levemente y seguían sin tener sentido. Por estos motivos, esta fuente de información fue descartada para nuestra base de conocimiento.

Como conclusión de este experimento a la hora de crear la base de conocimiento, se extrajeron ideas tales como intentar evitar obtener los datos de plataformas Web siempre que fuese posible. Esto se debe a que este tipo de plataformas suelen cambiar el aspecto y el orden de los elementos, teniendo como consecuencia que el sistema para obtener los datos quede inservible. Por otra parte, también se llegó a la conclusión de que es mejor agrupar elementos que tuviesen la misma temática antes de almacenar dicha información para mejorar los resultados. Esto se debe a que, para obtener el contexto de una frase, no solo es necesario saber el significado de cada palabra, si no como se usan entre sí y con el conjunto global de la frase, además de tener la palabra a definir dentro de la propia definición. Finalmente, la decisión de elegir las propias palabras definidas como resultado de la temática, visto ahora desde la experiencia obtenida, no fue una buena opción debido a que no clasificaban bien las temáticas.

3.2.3 **Enfoque 2, Artículos de diferentes fuentes**

Para el segundo enfoque se optó por obtener, de manera manual, artículos cortos sobre ciertos temas de fuentes como periódicos, bibliotecas, etc. La idea era verificar que, si la definición de una palabra no era suficiente, igual la propia palabra, junto con los elementos más importantes con los que se suele utilizar, podría servir para determinar la temática principal de la frase.

La peculiaridad de este experimento era que el tratamiento de los datos para crear los índices fue realizado a mano. Los elementos que se mantenían de los artículos para generar el índice eran las palabras más importantes de dicho artículo y las que le daban sentido. El resto de palabras eran eliminadas. Esta técnica se basaba en la eliminación de stopwords o palabras que no añadían información significativa a la frase.

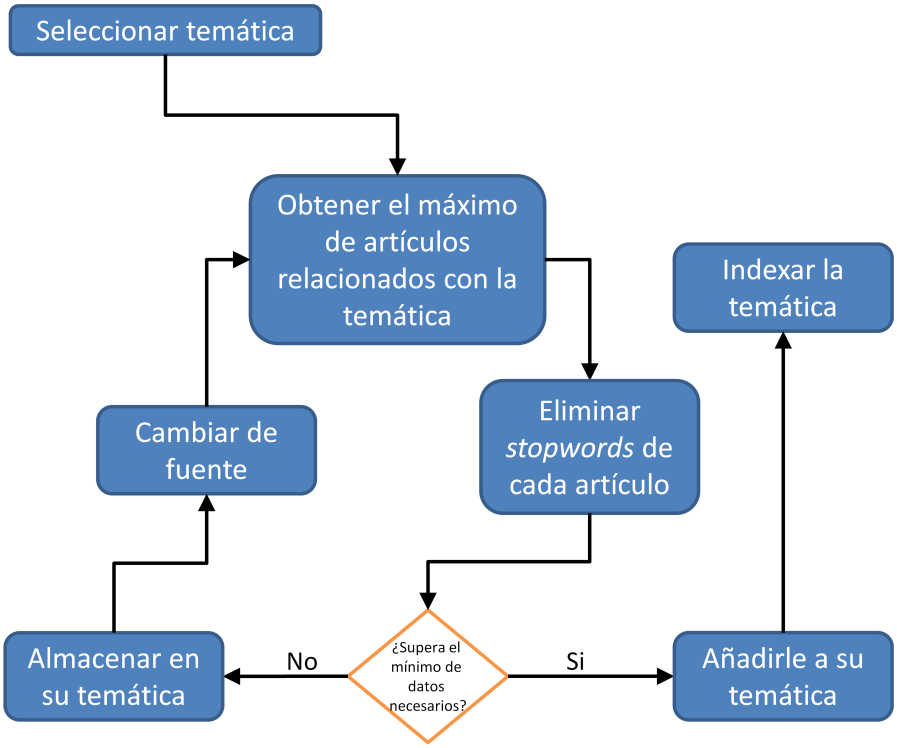


Figura 3.3: Metodología del experimento 2

Debido a la artesanal metodología utilizada, la cual se muestra en la figura 3.3, el tamaño de los artículos listos para indexar y crear la base de conocimiento, no fueron muy numerosos.

En la primera parte del experimento, se seleccionaron un número finito de temáticas para obtener artículos relacionados. Estos artículos comenzaron siendo simplemente 7 temáticas, las cuales, se eligieron aleatoriamente según su utilización en el lenguaje de las noticias de aquel momento.

Las temáticas fueron política, delitos, dinero, deportes, trabajo, tiempo y países. De estos temas, se obtuvieron diversos artículos de diferentes periódicos, artículos de opinión y de comentarios de la gente sobre esas noticias.

Estos artículos fueron obtenidos mediante copia directa de las diferentes fuentes. El texto no fue obtenido de manera automática aunque, a la vez que se iban obteniendo artículos, se iba analizando la fuente y la manera de obtener el texto de manera automática.

El problema que se planteaba, y por el cual no se pudo hacer de manera automática, era que dependiendo de la noticia, el estilo cambiaba. El estilo no era siempre el mismo para el mismo tipo de noticia o artículo y obtener, por tanto la información de dicha fuente de manera automática, no era viable.

A la vez que se iba obteniendo el texto, se utilizaba un sistema propio, este si era automático, para eliminar las stopwords. El objetivo de eliminar en este punto estas palabras, era para obtener el número de palabras mínimas necesarias para que una temática tuviera la suficiente información acerca de sus propios elementos. El número mínimo era una medida determinada empíricamente que consistía en almacenar, por lo menos, más de 500 palabras diferentes.

Una vez terminada la laboriosa tarea de obtener la suficiente documentación sobre las diferentes temáticas, limpiar el texto y eliminar las palabras innecesarias, se creaba una carpeta con el contenido que se había obtenido. Este proceso se hacía con todas las temáticas que se habían seleccionado.

A la hora de almacenar los datos, se utilizo la herramienta ya desarrollada previamente con el motor de indexación de Lucene y que esta descrita en el apartado 3.2.2. La única modificación fue que en vez de utilizar la versión 3.4, se utilizó la versión 3.6 con sus correspondientes necesidades.

Para este índice, los elementos invariables fueron los nombres de las temáticas y los elementos sobre los cuales realizar las búsquedas fueron las palabras almacenadas dentro de las carpetas de cada temática.

Las temáticas, que serían los resultados, fueron las propias temáticas que se habían elegido previamente. Esta decisión era una buena idea para mantener en futuros experimentos.

Aunque al principio se empezó con solo 7 temáticas, pronto se observó que esas temáticas no eran suficientes. Para abarcar una conversación es necesario tener, por lo menos, todos los temas que se tratan en dicha conversación y eso implica la necesidad de definir más temáticas y más elementos.

Las pruebas con conversaciones cortas dieron resultados bastante favorables, pero había demasiados elementos no detectados. Este problema se debía a la poca cantidad de información que ofrecía la fuente y que se había almacenado en la base de conocimiento. Por ese motivo se definieron más temáticas, durante varios experimentos con conversaciones, llegando hasta un total de 15 temáticas.

Además existía otro problema, la fuente desde la cual se habían obtenido los elementos para crear las categorías y los elementos que ellas contenían no eran uniformes, no seguían un patrón específico y no eran replicables, ya que dependían de la persona que los generara debido a que estaban filtrados de manera manual.

Como conclusión sobre los experimentos que se realizaron de esta manera, se obtuvieron las siguientes ideas. A la hora de obtener datos para generar la base de conocimiento, es mejor que la fuente tenga una estructura predefinida y que se tengan que consensuar y publicar los cambios antes de hacerlos. También se observó que los artículos de opinión y de periódicos en general no ofrecían la cantidad suficiente de datos útiles para nuestro fin, debido a que los escritores, muchas veces, mezclaban temas dentro del mismo artículo, diluyendo el tema principal y dejando inservible dicha parte.

3.2.4 **Enfoque 3, Artículos de Wikipedia**

Para el enfoque 3, se pensó en obtener una fuente de datos que uniera los 2 enfoques previos junto con otras características para la unión entre temáticas.

La decisión fue utilizar la enciclopedia on-line Wikipedia. Los motivos de esta decisión fueron: la gran cantidad de definiciones que contiene, los contenidos y descripciones de los artículos, el lenguaje que se utiliza (común y formal), los temas sobre los que trata la idea principal del artículo, el soporte de una gran comunidad y los diversos idiomas en los que trabaja, el soporte de Web Semántica que ofrece y los enlaces a artículos semejantes para obtener más conocimiento sobre la misma materia. En el área de la investigación, también existen motivos para utilizar esta fuente debido a su gran uso en la Web Semantica como lo demuestran los trabajos de Gabrilovich y Markovitch (2007), Strube y Ponzetto (2006) y Völkel et al. (2006).

El objetivo, en este tercer enfoque, era generar una base de conocimiento utilizando lo aprendido previamente. La creación de la base de conocimiento, en este caso, fue bastante más costosa que las anteriores. Aunque se utilizaron los mismos principios, también hubo que adaptar los recursos obtenidos, utilizando unificacio-

nes entre artículos. Se pensó en hacer la parte más tediosa como la obtención de datos, la unión de temáticas y la indexación, todo de manera automatizada. Para ello se reciclaron herramientas que se habían desarrollado para los experimentos anteriores y se desarrollaron nuevas para los objetivos marcados.

Lo primero que se hizo fue analizar la estructura de las páginas Web de Wikipedia. Conocer el formato que utilizaban nos permitió poder automatizar la recuperación de datos vía on-line y, mediante la herramienta Jsoup que ya se utilizó en el enfoque 1, (sección 3.2.2) y la posterior extracción de información. Una vez examinada la estructura y formato de los artículos, se desarrolló la herramienta específica para dicha fuente de recursos.

La creación de la base de conocimiento se comenzó obteniendo ciertos temas específicos, los cuales abarcasen varios aspectos de varios ámbitos generales para lograr obtener toda la información posible relacionada con dicha temática.

El problema que surgió en este punto fue por dónde empezar a obtener datos. ¿Cuál era el mejor artículo o artículos para empezar a recoger la información necesaria para sembrar nuestra fuente de datos y poder después recoger buenos frutos?

Para resolver este problema, la base fue los elementos que se habían generado y obtenido en los enfoques previos. En concreto, en las diferentes temáticas recuperadas. Posteriormente se obtuvieron artículos específicos de la Wikipedia que estaban dentro de la temática principal seleccionada. Estos artículos, se eligieron en base a la representatividad respecto a la temática principal, convirtiéndose además en un buen recurso desde el cual obtener nuevos artículos relacionados.

Así, las temáticas y artículos incluidos en la base de conocimiento fueron:

- **Vehículos** de motor y sistemas de transporte.
- **Tiempo**, centrándose en los diferentes fenómenos relacionados con el tiempo y en los diferentes periodos de tiempo.
- **Sexo**, centrándose en acciones, genitales o tipos de gustos sexuales, además de delitos relacionados con la sexualidad.
- **Deportes**, centrándose en diferentes deportes y entidades relacionadas con el deporte.
- **Trabajo**, centrándose en temas relacionados con el trabajo, la educación, lugares de trabajo, puestos de trabajo y delitos relacionados con el trabajo.
- **Comida**, centrándose en tipos de comidas, verduras y todo tipo de comestibles, además de las bebidas.

- **Dinero**, centrándose en elementos relacionados con el dinero, la moneda, tipos de moneda y bancos.
- **Persona**, centrándose en el ser humano, sus estados de desarrollo y razas.
- **Política**, centrándose en los diferentes elementos que componen la política a grandes rasgos, como son tipos de gobierno y partidos políticos.
- **Países**, centrándose en los 5 continentes y en los países que existen dentro de ellos.
- **Ciudades**, centrándose en las ciudades del mundo, sus nombres, sus gentilicios y sus distritos.
- **Religión**, centrándose en las más significativas y en diversos tipos de sectas, así como en referencias a características o creencias que la mayoría de las religiones poseen.
- **Miedos**, centrándose en los diferentes tipos de miedos o fobias que existen.
- **Enfermedades**, centrándose en las enfermedades humanas, en la medicina, en los medicamentos y en las diferentes formas que puede alcanzar una enfermedad.
- **Delitos**, centrándose en los actos criminales o delitos.

Como puede verse en la tabla 3.1 de temáticas y sus artículos principales, de cada temática se eligieron entre 9 y 11 elementos que fuesen significativos y que, además, no estuviesen en los enlaces de los artículos anteriores para lograr un alcance mayor en la información a obtener. Los artículos principales son artículos específicos de Wikipedia, con el nombre exacto del enlace a su información.

Una vez seleccionados los artículos principales de las temáticas, se diseñó un sistema para obtener la información de dichos artículos y sus "Véase también". Por cada uno de los artículos que se obtuvieron, también se almacenaron sus enlaces a otros artículos relacionados, llegando a seguir este sistema hasta obtener todos los artículos posibles referentes a la categoría principal. Cabe destacar que toda la información obtenida tenía conexión con la temática de la categoría principal y fue obtenida de manera automática, es decir, sin intermediación humana, permitiendo la actualización de forma sencilla si fuese necesario. La profundidad de esta rama de obtención de artículos llegaba tan solo a un nivel determinado, es decir, solo se podían obtener los artículos de la sección "Véase también" si el articulo desde el cual se obtenían era el principal. Esta maniobra se debía a que, de esta manera, se podía aumentar la información relacionada con dicho artículo y mejorar los posibles

PENSAR EN HACERLO EN ITEMIZE

Categoría	Artículos
Vehículos	vehículo, avión, furgoneta, camión, moto, autobús, Metro (sistema de transporte), tranvía, Clasificación de automóviles, Anexo:Modelos de automóviles por tipo
Tiempo	tiempo, lluvia, sol, día, semana, mes, año, Periodización, decada, siglo
Sexo	sexo, puta, pene, coño, violacion, coito, Fetichismo sexual, porno, mamada, sexo anal
Deportes	deporte, futbol, tenis, formula uno, baloncesto, voleybol, Moto GP, fifa, FIBA, natacion
Trabajo	Trabajo (sociología), educacion, Explotación laboral, Fabrica, salario, jubilacion, oficina, jefe, taller, mercado, Desempleo
Comida	comida, alimento, carne, pescado, legumbres, verduras, postres, bebida, Pirámide alimentaria, golosina
Dinero	dinero, moneda, billete, billetero, Euro, dolar, peseta, banco, Cajero automático, Contrato de compraventa
Persona	persona, Ser humano, Color del pelo, adolescencia, juventud, mujer, varon, niño, Adulto, Razas humanas
Política	politica, democracia, dictadura, libertad, Partidos politicos, republica, gobierno, Estado, partido popular, psoe, ONU
Países	pais, nacion, europa, america, asia, oceania, africa, vantartida, Portal:Países, Microestados
Ciudades	ciudad, Anexo:Aglomeraciones urbanas más pobladas del mundo, Anexo:Ciudades de la Unión Europea por población, Anexo:Ciudades por PIB, Anexo:Áreas metropolitanas por población estimada en 2005, Tamaños de ciudades históricas, Anexo:Áreas urbanas más extensas del mundo, Aglomeración urbana, Comunidad local, vecino
Religión	religion, cristianismo, judaismo, budismo, secta, Fe, Fe (cristianismo), Creencia, Cielo (religión), Infierno
Miedos	miedo, Pesadilla, terror, angustia, fobia, Anexo:Fobias, Cultura del miedo, Pánico, Ataque de pánico, Ansiedad
Enfermedades	enfermedad, Portal:Medicina, Anexo:CIE-10 Capítulo XXI: Factores que influyen en el estado de salud y contacto con los servicios de salud, Epidemia, catarro, gripe, sida, virus, Antibiotico, Receta médica
Delitos	delito, robo, Asesinato, Abuso sexual infantil, Pedofilia, Prostitución, Violencia, terrorismo, Persecución, Acoso físico

Tabla 3.1: Categorías y atributos buscados para el enfoque 3, Artículos de Wikipedia

resultados, ya que al final, a ese nivel de profundidad, los artículos relacionados también hablan y tratan del mismo tema principal. Por otra parte, la decisión de centrarse en el apartado "Véase también" se debe a que es una sección en la cual, el autor pone enlaces relacionados con el artículo de forma directa y que existen dentro de la estructura de la Wikipedia, teniendo cierta relación o interés con el artículo principal.

Este enfoque constaba de 2 procesos que dependen entre sí. En el primero se obtienen los datos de la Wikipedia y en el segundo se almacenan según temáticas.

En la figura 3.4 se muestra, de forma gráfica, la metodología seguida para la obtención de los datos de la Wikipedia.

En la figura 3.5 se muestra de forma gráfica, la metodología seguida para la indexación de los datos obtenidos de la Wikipedia.

3.2.4.1 **Explicación de los diagramas**

Como en las anteriores pruebas de generación de la base de conocimiento, se hicieron pruebas básicas para determinar los resultados. El número de bases de conocimiento que se crearon antes de llegar a la lista de artículos y categorías que se ha descrito previamente fueron de más de 50.

En cada una de ellas, se realizaban pruebas para determinar si era necesario introducir nuevos artículos para generar más conocimiento, si algún tipo de artículo se solapaba con otra categoría y si existía la posibilidad de crear nuevas categorías. Estas pruebas eran semejantes a las realizadas previamente con el resto de enfoques, que consistía en evaluar conversaciones de manera manual.

La primera parte del funcionamiento del sistema desarrollado, se describe en la figura 3.4 y es similar a los desarrollados para los anteriores enfoques. Un sistema que, tras un análisis de la estructura de las páginas Web de la plataforma, extrae la información relevante para almacenarla y procesa dicha información. La peculiaridad de este sistema radica en que, además de obtener la información del artículo en cuestión, también obtiene los enlaces a otros artículos relevantes. Estos artículos son almacenados primeramente en memoria para después obtenerlos generando un subnivel de artículos.

El resultado fue una estructura de artículos que comenzó con las 15 temáticas iniciales y terminó con más de 160 elementos sobre los diferentes aspectos que rodeaban a cada una de las temática iniciales.

Una vez obtenidos todos los artículos, el segundo proceso, descrito en la figura 3.5, fue indexar la información obtenida. Se creó la base de conocimiento con las mismas tecnologías que en los anteriores enfoques pero con la versión 4.0 de Lucene. La única peculiaridad de crear esta base de conocimiento fue que se utilizó

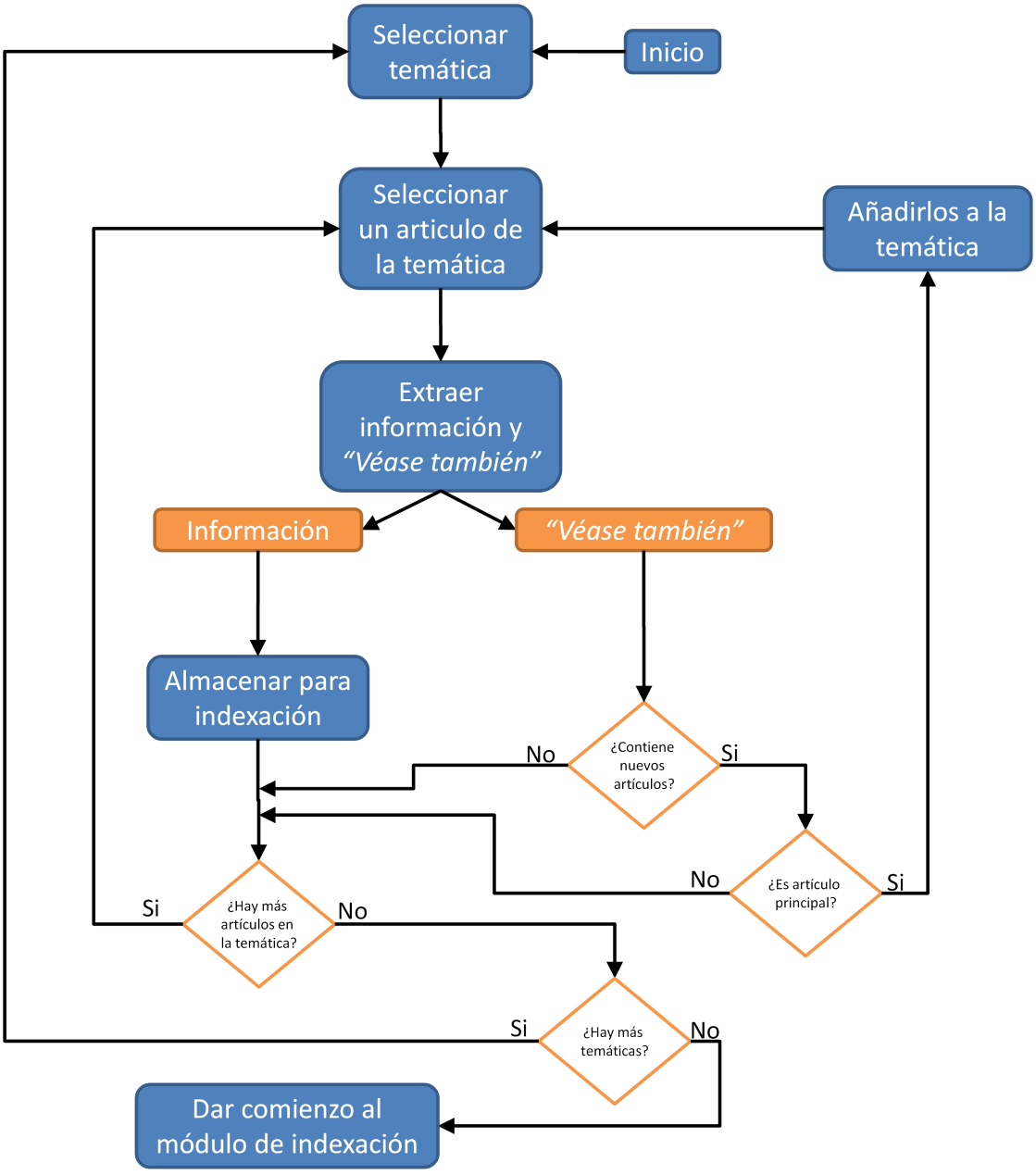


Figura 3.4: Metodología de extracción de datos de Wikipedia del enfoque 3.

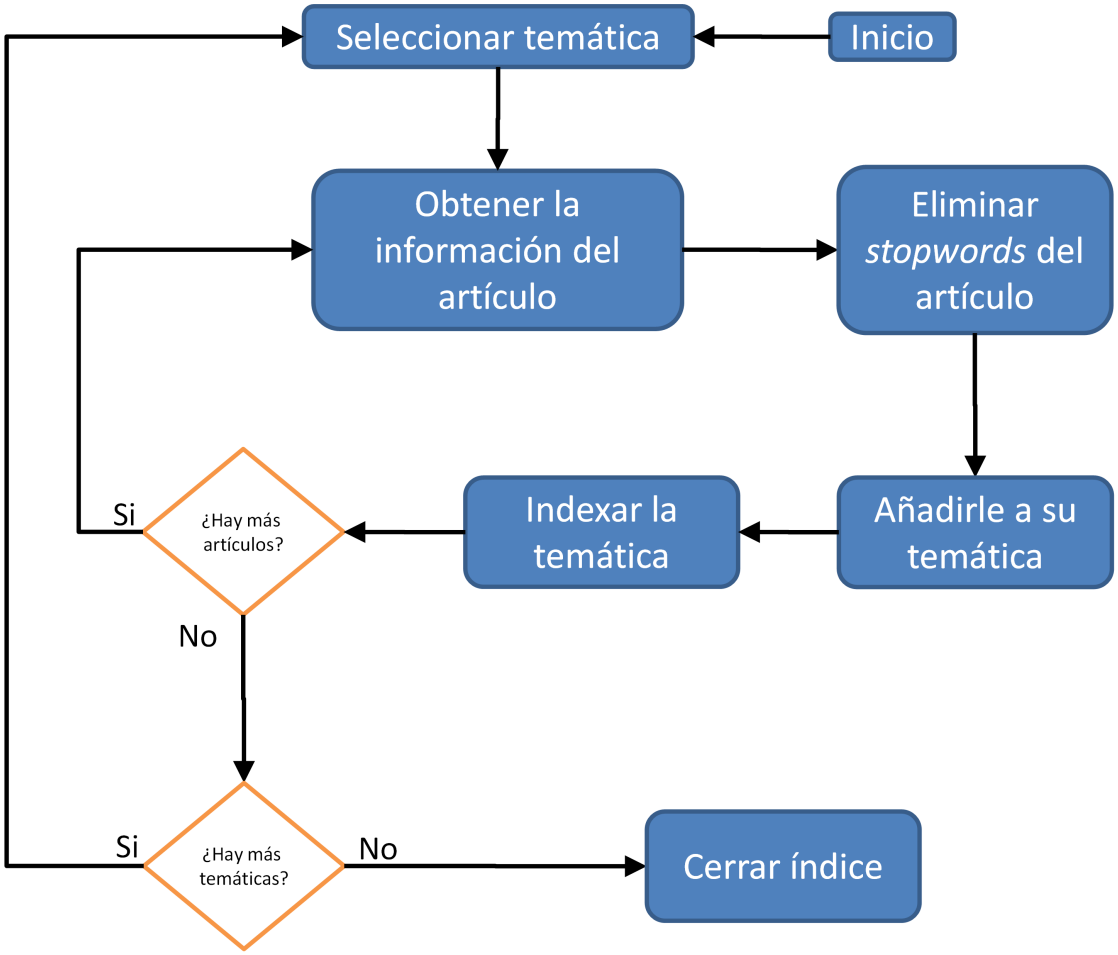


Figura 3.5: Metodología de indexación de los datos obtenidos de Wikipedia en el experimento 3.

una metodología específica para unir los artículos en categorías según su temática principal. De esta manera, se lograba abarcar de manera más amplia los ámbitos a los que pudiera hacer referencia esta categoría. Cabe destacar que aunque se modifiquen las versiones de esta herramienta, en lo único que afectan estos cambios es en la manera de almacenar los datos. Esto se debe a que la obtención de las frecuencias y demás valores que evalúan los resultados, son desarrollos propios y no están afectados por las diferentes versiones de Lucene. Este segundo proceso unía los diferentes artículos obtenidos de Wikipedia que estaban dentro de una temática definida y los almacenaba en el índice correspondiente.

El proceso comenzaba recorriendo cada temática existente obteniendo todos sus artículos. De estos, se obtenía toda la información, se eliminaban las stopwords y se añadía dicha información generada como un campo del índice. De este campo, el elemento diferenciador era el nombre de la temática y el campo de los datos era esta información obtenida de los artículos.

Este proceso se realizaba para todas las temáticas y después se cerraba el índice para poder utilizarlo en los experimentos.

A modo de conclusión, con esta fuente de datos los resultados de las pruebas aumentaron significativamente el acierto. El problema era que siempre era necesario aumentar la cantidad de elementos de la base de conocimiento para mejorar los resultados. Si no se aumentaban, ciertos resultados no se detectaban. El motivo no era porque fueran complicados de detectar, sino porque la base de conocimiento no tenia nada o la suficiente información acerca de la temática.

La idea de utilizar Wikipedia se debe a que está muy extendido su uso en investigación de Web Semantica. Muestra de ello son los trabajos de Völkel et al. (2006), Gabrilovich y Markovitch (2007) o Strube y Ponzetto (2006) entre otros. Además existen investigaciones, como la de Hadj Taieb et al. (2013), que utilizan Wikipedia como fuente para medir la relación semántica de los diferentes artículos y el contenido de ellos con otros artículos. También existen investigaciones, como la de Moro y Navigli (2012), que utilizan la información de los artículos de Wikipedia para construir redes semánticas basadas en conceptos, ontologías y relaciones etiquetadas. Como se puede ver, el uso de Wikipedia para generar bases de conocimiento útiles para la investigación es extenso. Una base de conocimiento que contiene más de 10 millones de entidades y datos sobre los artículos de Wikipedia es YAGO[6] que actualmente esta por la versión 3. Este monumental trabajo también contiene información de WordNet[7] para mejorar los resultados y se ha utilizado en varias investigaciones, como las de Hoffart et al. (2013), Hoffart et al. (2010) y Hoffart et al. (2011), para

[6] www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago

[7] wordnet.princeton.edu

generar bases de conocimiento y metodologías de extracción de datos . Uno de los usos que se le da a esta base de conocimiento, aparte de las pruebas en investigaciones, es la del uso en sistemas SPARQL. SPARQL es un lenguaje estandarizado para la consulta de grafos RDF (*Resource Description Framework*), normalizado por el RDF Data Access Working Group (DAWG) del World Wide Web Consortium (W3C).

Con esta prueba que realizamos con Wikipedia, quedo claro que este era un camino a seguir, pero que era necesario recuperar mucha más información y etiquetarla correctamente.

Además, se planteo otro gran problema. La obtención de los artículos estaba automatizada, pero para comenzar la obtención de elementos era necesario un prerrequisito, introducir los temas y artículos importantes, o al menos, los que se pensaba que serían lo suficientemente relevantes.

Este punto planteó un inconveniente. No se podía tener siempre actualizada la base de conocimiento sin una persona que fuese actualizando la configuración para obtener los artículos nuevos. Además, no se sabía qué nuevos artículos se habían creado desde la última obtención de datos. Por lo que se pensó en otra solución, aunque ya estábamos encaminados hacia la solución final.

3.2.5 Enfoque 4, Artículos de DBpedia

Antes de comenzar con la explicación del proceso seguido, es necesario explicar ciertos temas y conceptos correspondientes a los recursos utilizados.

Conceptos y definiciones sobre los recursos utilizados

Los recursos que se han utilizado en este enfoque tienen un significado y objetivo específico. Además son estándares extendidos en el área de la Web Semántica, pero no todo el mundo conoce su definición, objetivo y uso. Es por ello que en esta sección se va a explicar los referentes a los que se han utilizado en este enfoque.

Las categorías de "*Simple Knowledge Organization System*" (SKOS) son un recurso en forma de aplicación de "*Resource Description Framework*" RDF que se creó por el W3C[8] para proporcionar un modelo que represente la estructura básica y el contenido de esquemas conceptuales como listas, encabezamientos de materia, taxonomías, esquemas de clasificación, tesauros y cualquier tipo de vocabulario controlado.

La codificación RDF fue originalmente diseñado para ser un sistema de modelado de datos para metadatos, pero ha llegado a ser utilizado como un método general para la descripción conceptual o modelado de la información que se implementa

[8] www.w3.org

en los recursos Web, utilizando una variedad de notaciones de sintaxis y formatos de serialización de datos. El modelo de datos de SKOS es una ontología definida con "*Web Ontology Language*" OWL en su vertiente completa o Full. Dentro de OWL existen 3 variantes, Lite que es la más sencilla, DL que contiene más información y la versión Full. La diferencia entre estas versiones radica en la posibilidad de resolver una sentencia en tiempo finito. Las versiones Lite y DL pueden ser resueltas en tiempo finito debido a que no contienen bucles cuando se analizan las sentencias. La versión Full contiene bucles y las sentencias pueden no ser nunca resueltas. El sistema OWL también forma parte de W3C.

El sistema SKOS, descrito en el trabajo de Miles y Bechhofer (2009), está dentro de muchas investigaciones, como las de Schandl y Blumauer (2010), Morshed et al. (2010), Ma et al. (2011), Mayr et al. (2010) y Sánchez (2013), debido al modelado de datos que realiza. Estos modelados se utilizan en la creación y gestión de los tesauros de diferentes ámbitos y sus relaciones. También se ha utilizado, tal y como se puede ver en los trabajos de Tuominen et al. (2009), Cohen (2013) y Haslhofer et al. (2013), para la extracción y generación de información de la Web semántica mediante ontologías y servicios específicos y Lacasta et al. (2013) lo ha utilizado en su trabajo para mostrar de manera gráfica las diferentes relaciones entre las palabras y sus significados .

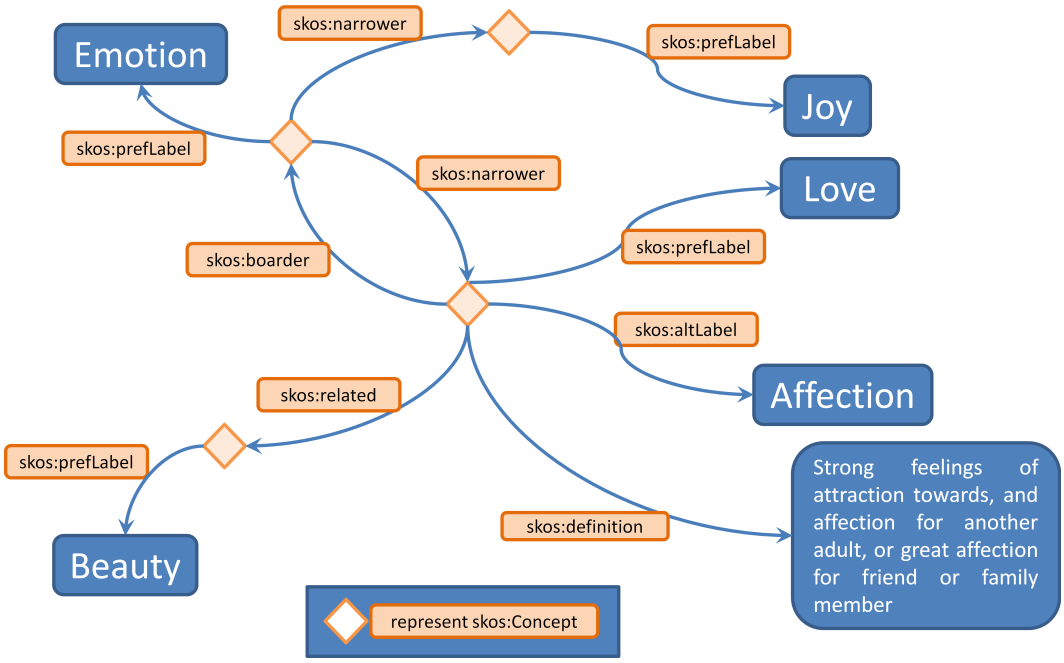


Figura 3.6: Esquema de la relación en SKOS

En la estructura de los modelos SKOS, los conceptos se identifican por referen-

Categoría	Vocabulario
Conceptos y esquemas de conceptos	skos:Concept, skos:ConceptScheme, skos:inScheme, skos:hasTopConcept
Etiquetas léxicas	skos:prefLabel, skos:altLabel, skos:hiddenLabel
Relaciones semánticas	skos:semanticRelation, skos:broaderTransitive, skos:narrowerTransitive, skos:broader, skos:narrower, skos:related
Documentación	skos:note, skos:scopeNote, skos:historyNote, skos:changeNote, skos:definition, skos:editorialNote, skos:example
Colecciones de conceptos	skos:Collection, skos:OrderedCollection, skos:member, skos:memberList
Propiedades de mapeado	skos:mappingRelation, skos:exactMatch, skos:broadMatch, skos:narrowerMatch, skos:relatedMatch
Notaciones	skos:notation

Tabla 3.2: Vocabulario SKOS

cias en formato "*Uniform Resource Identifier*" URI. Los conceptos pueden etiquetarse en varios idiomas formando cadenas de texto que se relacionan entre sí mediante relaciones semánticas de diversa tipología. Este sistema ofrece la opción de buscar conceptos de diferentes esquemas, así como definir colecciones agrupadas y ordenadas de conceptos. Además, también permite establecer relaciones entra las etiquetas asociadas a los conceptos. El sistema SKOS, se diseñó para crear sistemas organizativos nuevos y poder migrar los ya existentes para poder adaptarlos para su uso en la Web Semántica de manera sencilla y poder utilizarlo conjuntamente con OWL. Una de las características más importantes del sistema SKOS es que ofrece un lenguaje sencillo y un modelo intuitivo para poder utilizarlo junto con OWL o de manera independiente. Esto marca a SKOS como un paso intermedio entre el desorden de la actual Web y el riguroso protocolo, formalismo y descriptivo orden de las ontologías definidas con OWL.

En la figura 3.6 se muestran las diferentes uniones que existen entre diferentes conceptos y entidades que contienen una misma descripción, y en la tabla 3.2 el vocabulario que se utiliza.

Método del enfoque 4

En este enfoque, para obtener una base de conocimiento que fuera útil, tanto para esta investigación como para el uso de toda la comunidad científica, la base ha sido las fuentes de DBpedia descritas en el trabajo de Auer et al. (2007).

Además, se han tenido en cuenta las experiencias y resultados obtenidos en anteriores enfoques como son:

- No obtener únicamente definiciones, sino unir la definición con el entorno de uso.
- Obtener suficiente información para poder abarcar todos los elementos posibles de la conversación.
- El formato de los datos de las fuentes debe estar previamente definido para suplir la necesidad de actualizaciones periódicas de estas bases de conocimiento de manera automática.

Esta decisión se debe a que DBpedia es un proyecto para la extracción de datos de los artículos que existen y se crean dentro de la enciclopedia libre Wikipedia, de manera estructurada. Además, está actualizada, revisada por dicha comunidad y ofrece relaciones entre los diferentes artículos que posee la enciclopedia libre.

Estas fuentes contienen mucha información sobre los documentos que contiene Wikipedia, como por ejemplo los títulos de los artículos, resúmenes de unas 500 letras sobre el contenido del documento, el propio documento completo, las imágenes,

geo-posicionamiento, la relación entre categorías y conceptos, enlaces a diferentes bases de conocimiento como DBTune[9], DBLP[10], Freebase[11], BBC Wildlife Finder[12], MusicBrainz[13], New York Times[14], WordNet, YAGO y mucha más información.

Por otra parte, en el ámbito de la experimentación, existen investigaciones, como las de Kobilarov et al. (2009) y Auer et al. (2007), que utilizan DBpedia junto con otras fuentes de información y tecnologías semánticas para integrar sistemas que logren la unión de documentos mediante su contenido. Estos proyectos están dentro del proyecto LinkingOpenData[15]. Otras investigaciones, como la de Lehmann et al. (2007), sirven para la generación de herramientas que ayudan a explotar las relaciones semánticas entre objetos Web que contienen conocimiento y están etiquetadas. También hay investigaciones, como la de Aggarwal y Buitelaar (2012), que usan "*Natural-Language-Query*" sobre DBpedia para la extracción de información útil sobre los elementos que buscan. Otras investigaciones, como la de Zhao et al. (2013), unen las búsquedas en DBpedia con Lucene para la unión de entidades en documentos, utilizando el "*part-of-spech*" y la desambiguación de los elementos de dichos documentos. Algunos enfoques como el de Boo y Anthony (2012), incluso utilizan algoritmos como "*Page-Rank*", desarrollado por Page et al. (1999), para analizar la estructura de las uniones de DBpedia y mejorar las recomendaciones que este algoritmo ofrece para las páginas Web. Otro aspecto, para el cual también se utilizan los recursos de DBpedia, es el reconocimiento de entidades nombradas (NER) como se muestra en el trabajo de Santos et al. (2013). También se utilizan los recursos geoposicionales de DBpedia y la información obtenida de un dispositivo GPS dentro de un teléfono móvil inteligente o *Smartphone*, como se muestra en el trabajo de Becker y Bizer (2009), para ofrecer información sobre el lugar en el cual, se encuentra el usuario.

Como puede verse, el uso de estas fuentes es una base para los estudios sobre Web Semántica, para las técnicas del procesamiento del lenguaje natural y demás experimentos que requieran un contexto de información especifico. Es por esto que se ha usado esta fuente en las investigaciones.

La metodología que se siguió para lograr las tres bases de conocimiento parte de obtener los archivos del repositorio de DBpedia[16]. Aunque el sistema estaba pensado para poder ser utilizado con cualquier idioma que tiene disponible y que soporta DB-

[9] dbtune.org
[10] dblp.uni-trier.de
[11] freebase.com
[12] www.bbc.co.uk/nature/wildlife
[13] musicbrainz.org
[14] www.nytimes.com
[15] www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData
[16] wiki.dbpedia.org/Downloads38

pedia, para poder aprovechar mejor el potencial de todo su conocimiento se escogió en inglés.

La versión que se usó es la 3.8 y los archivos fueron *long_abstracts_en.ttl*, *skos_categories_en*, *article_categories_en* y *instance_types_en*. Los tres últimos archivos, se han utilizado para crear las temáticas principales, aunque cada recurso fue analizado para ver su importancia con respecto al cometido que queríamos lograr. El motivo es que contienen las categorías en las cuales se encapsulan todos los artículos de la Wikipedia como se muestran en los ejemplos de las figuras 3.7, 3.8 y 3.9.

```
<http://dbpedia.org/resource/Autism>
<http://purl.org/dc/terms/subject>
<http://dbpedia.org/resource/Category:Autism>.
<http://dbpedia.org/resource/Category:Communication_disorders>.
<http://dbpedia.org/resource/Category:Mental_and_behavioural_disorders>.
<http://dbpedia.org/resource/Category:Neurological_disorders>.
<http://dbpedia.org/resource/Category:Neurological_disorders_in_children>.
<http://dbpedia.org/resource/Category:Pervasive_developmental_disorders>.
<http://dbpedia.org/resource/Category:Psychiatric_diagnosis>.
<http://dbpedia.org/resource/Category:Learning_disabilities>.
<http://dbpedia.org/resource/Anarchism>
<http://purl.org/dc/terms/subject>
<http://dbpedia.org/resource/Category:Anarchism>.
<http://dbpedia.org/resource/Category:Political_culture>.
<http://dbpedia.org/resource/Category:Political_ideologies>.
<http://dbpedia.org/resource/Category:Social_theories>.
<http://dbpedia.org/resource/Category:Anti-fascism>.
<http://dbpedia.org/resource/Category:Greek_loanwords>.
<http://dbpedia.org/resource/Agricultural_science>
<http://purl.org/dc/terms/subject>
<http://dbpedia.org/resource/Category:Agronomy>.
{...}
```

Figura 3.7: Formato de *article_categories*.

```
<http://dbpedia.org/resource/Autism>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/ontology/Disease>
<http://www.w3.org/2002/07/owl#Thing>
<http://dbpedia.org/resource/Animal_Farm>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://schema.org/Book>
<http://dbpedia.org/ontology/Book>
<http://purl.org/ontology/bibo/Book>
<http://www.w3.org/2002/07/owl#Thing>
<http://dbpedia.org/ontology/WrittenWork>
<http://dbpedia.org/ontology/Work>
<http://schema.org/CreativeWork>
{...}
```

Figura 3.8: Formato de *instance_types*.

Para la obtención de las temáticas, uno de los archivos elegidos ha sido *skos_categories_en* debido a que, es un recurso en forma de aplicación de RDF. El resto de archivos elegidos han sido *article_categories_en* debido a que contiene los enlaces entre los

```
<http://dbpedia.org/resource/Category:Futurama>
<http://www.w3.org/2004/02/skos/core#Concept>
Futurama"@en .
<http://dbpedia.org/resource/Category:Works_by_Matt_Groening>
<http://dbpedia.org/resource/Category:Comic_science_fiction>
<http://dbpedia.org/resource/Category:Wikipedia_categories_
named_after_American_animated_television_series>
<http://dbpedia.org/resource/Category:World_War_II>
<http://www.w3.org/2004/02/skos/core#Concept>
"World War II"@en
<http://dbpedia.org/resource/Category:Wikipedia_categories_
named_after_wars>
<http://dbpedia.org/resource/Category:20th-century_conflicts>
<http://dbpedia.org/resource/Category:Global_conflicts>
<http://dbpedia.org/resource/Category:1940s_conflicts>
<http://dbpedia.org/resource/Category:1930s_conflicts>
<http://dbpedia.org/resource/Category:Conflicts_in_1939>
{...}
<http://dbpedia.org/resource/Category:Conflicts_in_1945>
<http://dbpedia.org/resource/Category:Global_conflicts>
<http://dbpedia.org/resource/Category:Modern_Europe>
<http://dbpedia.org/resource/Category:Modern_history>
<http://dbpedia.org/resource/Category:Wars_involving_Albania>
<http://dbpedia.org/resource/Category:Wars_involving_Argentina>
<http://dbpedia.org/resource/Category:Wars_involving_Australia>
{...}
```

Figura 3.9: Formato de *skos_categories*.

conceptos y las categorías de los documentos mediante el vocabulario SKOS, tal y como se muestra en el trabajo de Bizer et al. (2009). El lenguaje SKOS esta descrito en la Tabla 3.2. El archivo *instance_types_en* porque contiene las tripletas en formato *$<Objeto><rdf:type><$clase>* en base a la extracción ontológica del documento y el archivo *long_abstracts_en.ttl*. Este último contiene el título del articulo al que se refiere y toda la información que este contiene.

En los ejemplos de las figuras 3.7, 3.8, 3.9 y 3.10 se muestran trozos de los archivos que se han utilizado. En ellos se muestra que disponen de un formato predefinido en base a "*Uniform Resource Locator*" (URL) y elementos de texto. La estructura es semejante a un archivo "*eXtensible Markup Language*" (XML).

Los elementos que se han utilizado para construir la base de conocimiento están resaltados en diferentes colores. El rojo es para los nombres de los artículos que contienen la información, el color verde para las categorías a las que pertenece y el color azul es para la información que contiene, en Wikipedia, el artículo. En los ejemplos 3.7, 3.8 y 3.9 están marcadas las categorías a las que pertenece cada uno de los artículos y en el ejemplo 3.10 se ha remarcado el artículo, el cual contiene toda la información.

Para generar la base de conocimiento se creó un sistema que estaba compuesto por 3 subsistemas. Los dos primeros subsistemas se encargan de preparar los datos correctamente para su tratamiento y el último se encarga de la generación de los

```
<http://dbpedia.org/resource/Autism>
<http://dbpedia.org/ontology/abstract>
"Autism is a disorder of neural development characterized by impaired
social interaction and communication, and by restricted and repetitive
behavior. The diagnostic criteria require that symptoms become apparent
before a child is three years old. Autism affects information processing
in the brain by altering how nerve cells and their synapses connect and
organize; how this occurs is not well understood. It is one of three
recognized disorders in the autism spectrum (ASDs), the other two being
Asperger syndrome, which lacks delays in cognitive development and
language, and pervasive developmental disorder, not otherwise specified
(commonly abbreviated as PDD-NOS), which is diagnosed when the full set
of criteria for autism or Asperger syndrome are not met. Autism has
a strong genetic basis, although the genetics of autism are complex
and it is unclear whether ASD is explained more by rare mutations,
or by rare combinations of common genetic variants. {...} An autistic
culture has developed, with some individuals seeking a cure and others
believing autism should be accepted as a difference and not treated as a
disorder."@en . {...}
```

Figura 3.10: Formato de *long_abstracts*.

índices de la base de conocimiento, partiendo de los recursos previamente creados.

Los nombres de estos subsistemas, por orden de utilización, son *Generador de Archivos de DBpedia*, *Etiquetador Lingüístico* y *Generador de índices*.

3.2.5.1 **Generador de Archivos de DBpedia**

Para obtener los datos necesarios de cada archivo seleccionado acorde con las necesidades que se necesitaban, se generó un sistema de 7 pasos automatizados. Estos pasos debían ser secuenciales para lograr un sistema de actualización de la base de conocimiento que no necesitase interacción humana y que fuera fiable, siempre y cuando los sistemas de etiquetado de los recursos de DBpedia no variaran.

Cada paso actuaba sobre ficheros específicos que han sido comentados previamente. El objetivo era generar un sistema automático, a la vez que supervisable, para poder evaluar en cualquier momento, y antes del siguiente paso, si los resultados de dicho paso eran correctos.

El primer paso consistía en limpiar las carpetas en las cuales se iban a guardar los resultados. Las carpetas que se utilizaban eran 6. La de resultados que era de carácter temporal, la que contenía los datos de las categorías (*skos_categories_en*, *instance_types_en* y *article_categories_en*), la que contenía toda la información sobre los documentos de Wikipedia (*long_abstracts_en.ttl*), la que almacenaba los posibles fallos que ocurrieran con los diferentes artículos, la que almacenaba los resultados de la extracción de todas las categorías y la que unía los documentos de Wikipedia con cada categoría asociada. Este primer paso, aunque pueda parecer inútil, hacía una función muy importante. Esta función era la de actualizar completamente la base de

datos, eliminando los recursos previos si los hubiera. Al eliminar los recursos previos, en el caso de que hubiera una base de conocimientos ya generada, no se estaba eliminando dicho conocimiento. El funcionamiento de los recursos de DBpedia se basa en ofrecer la información que ya existía previamente, subsanando errores en el caso haber fallos, junto con las nuevas actualizaciones y relaciones.

En el segundo paso, se generaban las clases relativas a las categorías existentes en los archivos de categorías. Estos archivos eran los que se encuentran en la carpeta de datos de categorías. El proceso recorría estos ficheros con elementos que gestionan cada uno de ellos, para obtener las categorías a las que pertenece cada recurso. Por ejemplo y refiriéndonos a la figura 3.7, las categorías serían los elementos en verde y cada recurso, que al final es un documento con información en Wikipedia, sería el elemento en rojo. Lo mismo se puede aplicar a las figuras 3.8 y 3.9. El resultado de este proceso era generar archivos que se refirieran a las diferentes categorías. Además, dentro de la información de cada categoría estaban toda la información de los recursos de esa misma categoría. Estos recursos se almacenaban en la carpeta de resultados temporal. En este punto puede surgir la pregunta de para qué se utilizaban esos tres archivos de categorías si, en principio, todas las categorías podían estar en cada uno de esos archivos. La respuesta es sencilla, no existen las mismas referencias a categorías en un solo archivo, pero en la suma de los tres, sí existen.

El tercer paso consistía en eliminar los recursos repetidos de cada categoría. La función de este paso era generar categorías que no tuvieran elementos repetidos. El proceso constaba de recorrer todos los archivos que correspondían a categorías, que estaban almacenadas en la carpeta de resultados temporales, e ir almacenando temporalmente cada recurso. De esta manera, cuando se obtuviera un recurso que ya se tenia previamente almacenado, simplemente se obviaba. Una vez que se terminaba de recorrer dicha categoría, se escribían los resultados que estaban almacenados y no estaban repetidos. Los resultados se almacenaban en la carpeta de resultados de la extracción de todas las categorías.

El cuarto paso era de limpieza. Este paso se ocupaba de limpiar el directorio que contenía los resultados temporales para poder continuar utilizándolo.

El quinto paso consistía en obtener la información relativa a un documento específico existente en Wikipedia. La información se extraía del archivo *long_abstracts_en.ttl*. Estos archivos con la información referente a los documentos de Wikipedia se almacenaban en la clase temporal de resultados, que previamente se había limpiado en el paso anterior. El proceso recorría el fichero mencionado previamente, obteniendo la información referente al artículo en cuestión que existía en Wikipedia. Este archivo sigue el formato que se muestra en la figura 3.10. El proceso generaba tantos archivos como artículos existieran dentro del fichero. La nomenclatura que se eligió para etiquetar cada fichero fue utilizar el nombre, o título del articulo, como nombre

del archivo generado, y que toda la información de ese artículo fuera el contenido del fichero. Para que se pueda ver de forma más visual, teniendo en cuenta que el archivo que se ha seleccionado es el *long_abstract* y que en la figura 3.10 se muestra un artículo a modo de ejemplo, el nombre del fichero generado sería lo marcado de color rojo y el contenido del archivo lo marcado en color azul.

El sexto paso de este proceso era la unión de la temática y los archivos generados con la información sobre los artículos de Wikipedia. En el paso 2 se habían generado los archivos de las temáticas, que dentro contenían los nombres de los artículos referentes a dichas temáticas, y en el paso 5 se habían generado los archivos que contenían la información de cada artículo. Este paso lo que hacía era recorrer cada archivo de temática que se había generado en el paso 2, obteniendo todos los nombres de los artículos que contenía y, con cada nombre de artículo, obtenía el archivo del artículo y toda su información generada en el paso 5. Una vez que se tenia la temática, el nombre del artículo y toda la información sobre el propio artículo, se almacenaba esa información de la siguiente manera; se generaba un archivo cuyo nombre era el nombre de la temática y dentro se almacenaba el nombre del articulo junto con su información. Este proceso se hacía con todas las temáticas y artículos. De esta manera se generaban archivos etiquetados por temática que dentro contenían toda la información referida a dicha temática existente en Wikipedia. Este método era una manera de poder agrupar los diferentes elementos específicos o relacionados con una temática. Es importante mencionar que diferentes temáticas pueden contener al mismo artículo ya que, por ejemplo, el artículo de “casa” puede estar dentro de “vivienda”, de “construcciones”, de “vecindario”, etc. La idea era no filtrar ningún sentido que se le pueda aplicar a un elemento especifico y, con el resto de la información que facilitada, poder discernir a cual de esos temas que se han comentado, pertenece la información que se nos ha suministrado.

Por último, el paso 7 únicamente limpiaba los archivos que se habían generado y que ya no servían, para no almacenar datos inútiles que ocupen espacio innecesariamente.

Con estos pasos se generaron más de 2 millones de ficheros sobre temáticas que contenían toda la información de los artículos y sus nombres existentes en Wikipedia. Estos ficheros, como se ha comentado previamente, han sido descargados de los recursos que tiene DBpedia *on-line*, y se han procesado según los 7 pasos que se han comentado. El proceso de generación de estos recursos ha sido costoso debido a la gran cantidad de datos que eran y a los diferentes símbolos y demás elementos que contenían los artículos. Estos elementos se han intentado mantener lo máximo posible dentro de las temáticas para poder asegurar la esencia principal del artículo que los contenía. Los problemas que principalmente se han presentado al realizar esta parte del proceso han sido el análisis de las temáticas, de los artículos, su almacenamiento y la extracción de la información. Pero con los pasos previamente descritos,

se han solucionado.

Hasta este punto, se han generado los recursos ordenados por temáticas que después hemos usado para la generación de los diferentes índices mediante el sistema Etiquetador Lingüístico.

3.2.5.2 **Etiquetador Lingüístico**

Las clases generadas, una vez unidas, se introdujeron en el sistema de etiquetación lingüística que se ha desarrollado. Este sistema hace uso de la herramienta FreeLing desarrollada por Carreras et al. (2004b). Esta herramienta posee diferentes interfaces para utilizar su sistema lingüístico con otros lenguajes de programación diferentes al que está implementado. En este caso se utilizó una de esas interfaces. Concretamente la de Java y el sistema está desarrollado en Linux. Una vez lograda la comunicación con la herramienta, se generó un sistema propio que gestionaba diferentes aspectos semánticos y lingüísticos, siendo además utilizado en investigaciones posteriores con sistemas supervisados como los de Santos et al. (2012) y Santos et al. (2014), y en sistemas semisupervisados como los de de-la Peña-Sordo et al. (2013) y de-la Pena-Sordo et al. (2014).

A continuación, se mostrará cómo está desarrollado el sistema para obtener toda la información morfológica, sintáctica, semántica y pragmática de las diferentes sentencias. El sistema recoge cada sentencia que se le entrega en un contenedor específico llamado *Frase*. Este contenedor almacena los diferentes aspectos que se van a guardar sobre cada elemento de dicha frase en otro tipo de contenedor llamado *Terna*.

El contenedor *Terna* almacena todo lo relativo al elemento específico que se está analizando de la frase. Esta información alberga la propia palabra, la raíz de la palabra, el tipo de sintagma general y específico, tanto en la codificación de FreeLing como de manera legible para los humanos, la posición que ocupa dentro de la frase, el número de persona al que hace referencia, el género y toda la información que se muestra en la Tabla 3.3. Una vez definidos los contenedores donde se va a almacenar toda la información sobre la palabra dentro de la frase, pasamos a describir el sistema propio diseñado.

El sistema obtiene una frase específica que se almacena en el contenedor *Frase* y que es analizada con la herramienta FreeLing. Los procesos que realiza esta herramienta a la frase son:

- Análisis morfológico para determinar la forma, clase o categoría gramatical de ésta.
- Análisis de la categoría gramatical o "*part-of-speech*" para clasificar cada pala-

Categoría	Valores
Palabra Inicial	Es la palabra que existen dentro de la frase y la que se está analizando en este contenedor.
Palabra Raíz	Es la palabra raíz de la palabra principal.
Información del sintagma	Hay tres campos, uno es el que contiene el código obtenido de FreeLing , otro es el que contienen el sintagma principal de forma comprensible para humanos y el último es el que contiene el detallado y específico del principal. [ej. PP3CPD00, PRONOMBRE, PERSONAL]
Posición	Se almacena la posición exacta de la palabra principal del contenedor dentro de la frase.
Persona	Hace referencia en que persona está escrita la palabra, primera, segunda o tercera persona.
Género	Hace referencia al género que tiene la palabra principal. [ej. Masculino, femenino, común o neutro]
Número	Hace referencia al estado plural o singular de la palabra principal.
Caso	Hace referencia a las distintas formas que puede adoptar una palabra según su función sintáctica.
Tiempo	Hace referencia al tiempo en el cual está escrita la palabra. [ej. PRESENTE, IMPERFECTO, FUTURO, PASADO, CONDICIONAL, INDEFINIDO]
Modo	Hace referencia al modo verbal en que esta expresada la acción del verbo, teniendo en cuenta que la palabra principal es un verbo.
Sentidos	Es la lista de códigos referentes al elemento desambiguado dentro del contexto de la frase. Un ejemplo de estos códigos está en la Tabla 3.4 y su significado se muestra en la Tabla 3.5.
Grupos	Es la lista de grupos a los que pertenece la palabra principal dentro del contexto de la frase. Un ejemplo de estos grupos están en la Tabla 3.6.

Tabla 3.3: Información contenida dentro de cada contenedor Terna

bra que la forma según su tipo.

- Análisis y clasificación del reconocimiento de entidades nombradas "*Name Entity Recognition*" NER para reconocer nombres propios, de organizaciones, localizaciones, expresiones horarias, cantidades, valores monetarios, etc.

- Análisis semántico para obtener el significado, sentido e interpretación del elemento analizado mediante estructuras semánticas.

- Análisis para desambiguar las palabras que tengan varios significados según el contexto de la frase.

- Análisis sintáctico superficial o "*Chunk parsing*" que sirve para identificar los elementos constituyentes de la frase principal como son los grupos nominales, los verbos, etc., sin especificar las estructuras internas ni la función que desempeña en la frase.

- Análisis de dependencias "*Dependency Parser*" para mejorar y enriquecer el árbol generado en análisis previos.

Una vez realizados estos análisis, se extraen los resultados generados por cada palabra para rellenar toda la información en el contenedor *Terna* asociado a cada palabra de la frase. La información que se obtiene y todos los análisis realizados dependen del idioma en el cual esté la frase. Para poder gestionar de una manera más cómoda la información que de entrada, se ha creado un sistema de idiomas. Este sistema decodifica la información que devuelve FreeLing para rellenar la información de las *Ternas*. Esta información está basada en las directrices de "*Expert Advisory Group on Language Engineering Standards*" (EAGLES), un proyecto Europeo que ofrece unas pautas y guías para estandarizar la etiquetación e identificación de elementos lingüísticos y léxicos de los corpus de texto, independientemente del idioma en el que está escrito. El objetivo es poder obtener toda la información relativa a los análisis que realiza FreeLing de manera específica y automatizada, para que los sistemas que se utilicen después, puedan comprender, analizar y procesar la información, conociendo los datos que obtienen y, además, que a los seres humanos a la hora de gestionar errores o demás imprevistos, nos sea más fácil.

La información para realizar esta decodificación se ha extraído de diferentes lugares. La base principal fue obtenida del manual de usuario de FreeLing[17] en castellano. Para los diferentes idiomas, como el inglés y siguiendo las directrices del proyecto "*Penn Treebank Project*" descrito en el trabajo de Marcus et al. (1993), la

[17] nlp.lsi.upc.edu/freeling/doc/tagsets/tagset-es.html

información se obtuvo de lugares que la contenían[18]. El problema surgió con algunos códigos que generaba FreeLing y que no estaban detallados en la codificación de estos nuevos recursos. Para solucionar este problema, se obtuvieron empíricamente las restantes etiquetas para generar sus respectivas codificaciones en el sistema de idiomas.

Una vez creado el sistema de idiomas, solamente era necesario analizar las frases para procesarlas y obtener toda la información imprescindible para el resto de procesos. En las versiones superiores a la 1990, FreeLing es capaz de detectar el idioma en el cual está escrita la frase, pero ese análisis tarda bastante tiempo. Así, para reducir el tiempo de proceso, se le indica al sistema el idioma en el cual se encuentran escritas cada una de las frases a analizar. El sistema se encarga de obtener, del subsistema de idiomas, toda la codificación necesario para generar las *Ternas* con la información correspondiente al idioma y sus codificaciones.

El sistema de idiomas se encarga de traducir los códigos EAGLES (en la figura 3.11 se puede ver un ejemplo de la información obtenida mediante este sistema de la palabra "*dieron*" dentro de la frase "*Los chicos buenos dieron de comer al animal.*") Esta información es la que esta descrita en la tabla 3.3 y que después servirá para generar los índices de sentidos y el de grupos.

Con este sistema se han generado dos nuevos recursos. Cada uno de ellos conteniendo más de 1 millón de ficheros sobre la información lingüística de los artículos de DBpedia, ordenados por temáticas. La codificación que se ha seguido ha sido la misma que en el sistema anterior. El nombre del archivo que contiene la información corresponde al título de la temática. El proceso de generación de estos recursos ha sido muy costoso debido, de nuevo, a la gran cantidad de datos a procesar y a los diferentes símbolos y demás elementos que contenían los artículos. Estos elementos se han intentado mantener para evitar perder la esencia principal del artículo que los contenía y las referencias a la temática principal. Los problemas que principalmente se han presentado al realizar esta parte del proceso han sido el procesamiento de los ficheros, la generación de las diferentes codificaciones para los idiomas, el almacenamiento y la necesidad de utilizar maquinas un poco más potentes.

De estos dos nuevos recursos, el primero de ellos contiene los archivos con el análisis de los sentidos de toda la información de los artículos que alberga una temática concreta. Para crearlo se ha utilizado la codificación de la Tabla 3.4. El significado del código que muestra la Tabla 3.5 no se almacena debido a que con el propio código se podía obtener el significado. El segundo incluye los archivos que contienen los grupos a los cuales pertenecen los elementos con los sentidos del anterior recurso, utilizando la codificación de la Tabla 3.6.

Una vez terminado este último proceso, se han generado tres recursos. El prime-

[18] bulba.sdsu.edu/jeanette/thesis/PennTags.html

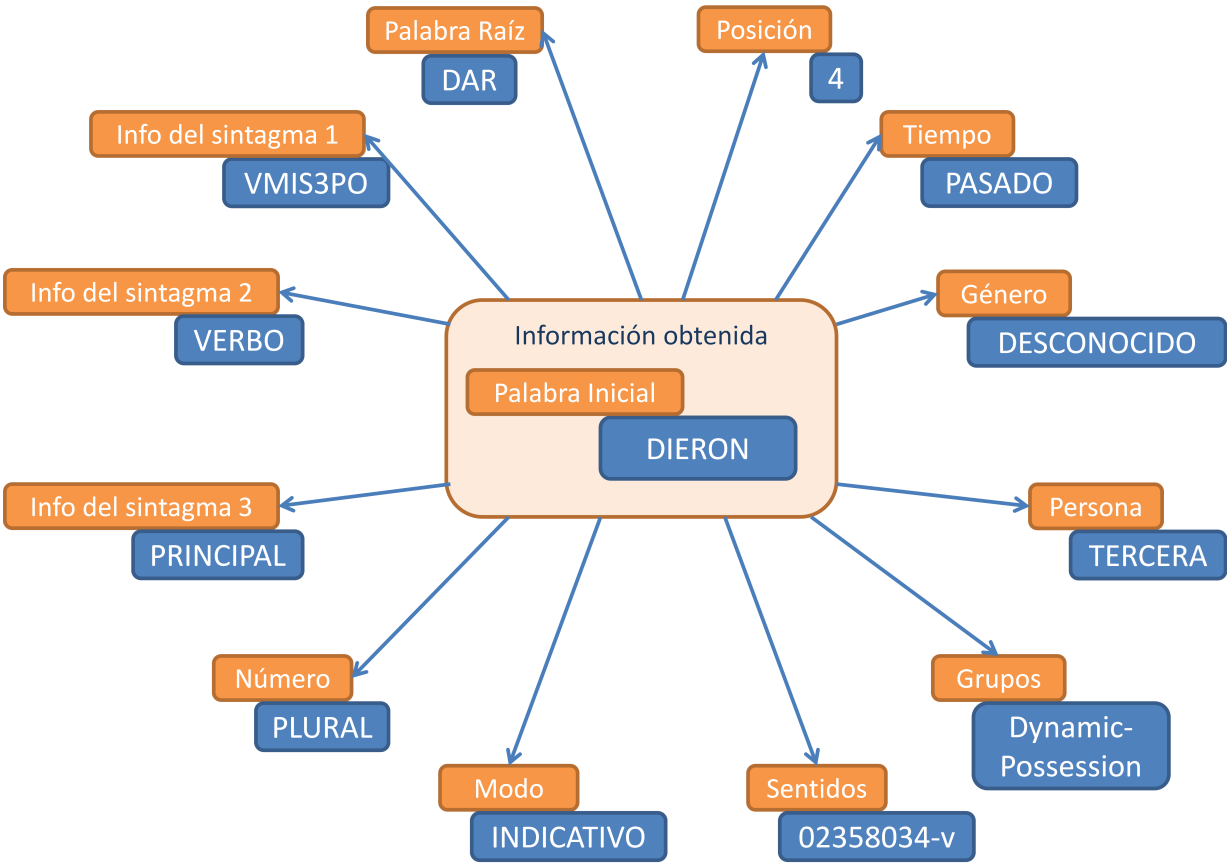


Figura 3.11: Ejemplo de elementos de una *Terna* obtenidos de la palabra *dieron* dentro de la frase *Los chicos buenos dieron de comer al animal.*

Elemento	Código
AMERICAN_FOOTBALL	00469651-n
BAND	08249038-n
WRITER	10794014-n
SOUTH_KOREA	08955626-n
WRESTLING	01504699-v

Tabla 3.4: Códigos de los elementos ambiguos

Elemento	Significado
AMERICAN_FOOTBALL	american_football american_football_game
BAND	band
WRITER	author writer
SOUTH_KOREA	replublic_of_korea south_korea
WRESTLING	wrestle

Tabla 3.5: Significado de los elementos ambiguos

Elemento	Descripción
AMERICAN_FOOTBALL	Agentive-BoundedEvent-Cause-Dynamic-Physical-Purpose-Recreation-Social-UnboundedEvent
BAND	Function-Group-Human
WRITER	Function-Human-Living-Object-Occupation
SOUTH_KOREA	Natural-Object-Part-Place
WRESTLING	Agentive-Physical-Social-UnboundedEvent

Tabla 3.6: Grupos de los elementos ambiguos

ro son las temáticas que contienen la información de los artículos de DBpedia y el segundo y tercero son los recursos que se han obtenido de este sistema. Estos tres nuevos recursos serán el punto de partida para el siguiente sistema, el generador de índices.

3.2.5.3 **Generador de índices**

Por último, se ha creado el sistema para generar los índices sobre los que se va a trabajar. Este sistema parte de los recursos que se han generado previamente con los sistemas Etiquetador Lingüístico y Generador de Archivos de DBpedia, tal y como se han descrito previamente.

Los índices que se han creado, utilizan la tecnología de Lucene[19] para generar los índices y hacer búsquedas sobre ellos. La versión que se ha utilizado de Lucene es la v4.0.

La herramienta Lucene, que ha sido descrita en el apartado 3.2.1.2, ha sido utilizada en varios experimentos con DBpedia como, por ejemplo, investigaciones como la de Zhao et al. (2013), que unen las búsquedas en DBpedia con Lucene para la

[19]lucene.apache.org/core/

unión de entidades en documentos, utilizando el "*part-of-spech*" y la desambiguación de los elementos de dichos documentos. También se utiliza esta herramienta en la creación de nuevos *frameworks*, como el de Mendes et al. (2011), para almacenar y hacer búsquedas sobre los nuevos datos de mejoras ontológicas en DBpedia entre otros experimentos y publicaciones.

Debido a las características que se han detallado en el apartado 3.2.1.2 se ha elegido esta herramienta para la indexación de los recursos generados por los sistemas Etiquetador Lingüístico y Generador de Archivos.

El sistema Generador de índices consta de un módulo para la indexación de los recursos generados. Este módulo se encarga de obtener los datos de los recursos de manera progresiva, pre-procesando los recursos y después indexándolos, generando un índice con 2 campos.

El proceso de generación de un índice parte de recorrerse todos y cada uno de los archivos de un recurso previamente generado. De este recurso, se obtienen todos los elementos que lo componen. En nuestro caso de los tres recursos que hemos generado, puede ser el título de un artículo, toda la información de dicho artículo, una palabra con su código desambiguado o el grupo al que pertenece dicha palabra. Esta información como tal, no siempre se puede utilizar para generar un índice, por lo que hay que pre-procesarla previamente.

El pre-procesado elimina los elementos comodín o *Wildcards* de la información. Estos elementos son utilizados por Lucene para generar búsquedas más detalladas y enriquecidas[20], pero si el texto que se quiere almacenar los contiene, el índice se crea de manera errónea.

Existe la posibilidad de escapar estos elementos y poder indexarlos junto a la información, pero se ha tomado la decisión de no mantenerlos ya que dichos elementos pueden interferir con emoticonos y demás codificaciones de los mensajes cortos que no deseamos tener y que no aportan nada, ya que para estos elementos, se necesita un sistema aparte para su procesamiento.

Aparte de eliminar los *Wildcards* de la información, el sistema obtiene el nombre del fichero, que es la temática sobre lo que trata todo su contenido. Al nombre de la temática también se le pre-procesa los *Wildcards* para eliminarlos.

Después de pre-procesar los elementos, se genera un analizador para indexar los elementos que contienen los recursos. Este analizador se encarga de separar en palabras o tokens cada frase de los elementos de la información y de eliminar las stopwords[21] previamente cargadas por el sistema. El sistema utiliza el analizador estándar de Lucene para realizar esta tarea.

[20] lucene.apache.org/core/2_9_4/queryparsersyntax.html
[21] paginaspersonales.deusto.es/patxigg/recursos/EnglishStopWords.txt

Una vez que se ha pre-procesado la información y la temática y se ha generado el analizador para el texto, se ha de que almacenar dicha información.

Lucene indexa elementos en base a campos. Cada campo contiene ciertas características según el uso que se le vaya a dar. En este caso solo se necesitan dos campos, a los cuales se les ha bautizado como *CLASE* e *INFO* y se muestran en la figura 3.12.

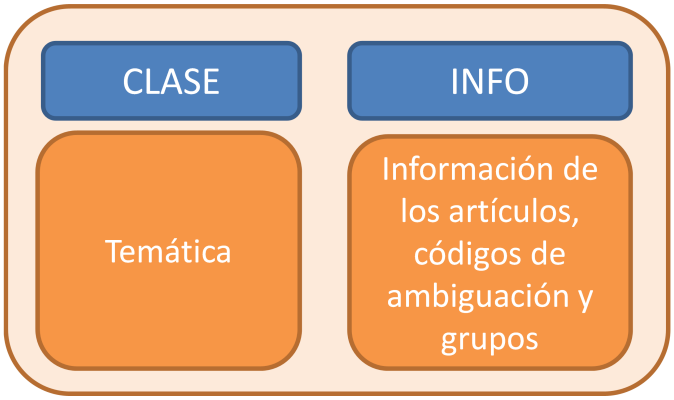


Figura 3.12: Campos CLASE e INFO con sus posibles valores.

El campo *CLASE* es el que contiene el nombre de la temática. Las características de este campo son que no se indexa, pero que se almacena en el índice. Esto quiere decir que el campo está almacenado dentro del índice, pero que el analizador que se ha configurado para indexar la información, no lo ha procesado y por lo tanto, no ha acortado y dividido su contenido para que las búsquedas puedan obtener sus resultados fácilmente.

Esto se debe a que el objetivo del sistema es que, hecha una búsqueda, devuelva la temática a la que puede pertenecer dicha información. Para ello, devuelve el campo *CLASE*. Si este campo está dividido, no devolverá el contenido completo, por lo que no se sabrá exactamente la temática.

El campo *INFO* es el que contiene toda la información sobre los artículos de cada temática, sobre los códigos ambiguos y grupos de las palabras que contienen los artículos. Evidentemente, como se han creado tres índices, cada uno de ellos contiene diferentes informaciones. Uno tiene la información sobre los artículos, otro sobre los códigos ambiguos y otro sobre los grupos de esas palabras ambiguas.

Este campo es diferente al anterior en varios aspectos. El primero y obvio, es el de la indexación. Este campo, a la vez que se almacena en el índice, se indexa por el analizador configurado para realizar búsquedas sobre él. Esto es lo que sirve para realizar las búsquedas y que devuelva resultados.

Por otra parte, tienen una configuración extra para futuros experimentos. Esta configuración extra es que almacena las frecuencias de los elementos, en este caso

palabras, que forman el contenido del campo.

Esta configuración se utiliza, por ejemplo, para generar máquinas de soporte vectorial o "*Support Vector Machines*" (SVM).

Estas máquinas son conjuntos de algoritmos de aprendizaje supervisado que se utilizan para la clasificación y regresión de elementos, normalmente texto. Su funcionamiento parte de tener un conjunto de entrenamiento etiquetado y entrenar los algoritmos para generar un modelo que pueda predecir la clase de una nueva muestra. Este modelo representa la muestra como puntos en el espacio de ejes, separando las clases por un espacio lo más amplio posible. Su funcionamiento es, dada una nueva muestra, genera un vector que represente dicha muestra y lo coloca en correspondencia con el resto de vectores ya generados en el mismo espacio de ejes. En función de la proximidad a otros vectores, se puede clasificar de una clase o de otra. En principio, para una mejor clasificación, cuanto más separados estén los vectores del modelo entrenado en el espacio de ejes, mejor será la clasificación.

Este tipo de sistema se utiliza en muchos campos del procesamiento del lenguaje natural, por ejemplo Drucker et al. (1999) lo utiliza en la clasificación de correo basura. Incluso hay investigaciones importantes, como la de Santos et al. (2011), sobre el filtrado de este tipo de correo basura o SPAM que lo utilizan.

Una vez que ya tenemos el elemento generado, el sistema revisa si existe esa misma temática dentro del índice. Si no existe, simplemente lo inserta, y si existe, la obtiene y la actualiza uniendo la información de ambos para después volver a meterla en el índice.

Este sistema se realiza para todos los elementos de los recursos generados en los sistemas anteriores. De esta manera se obtienen tres índices en los cuales se puede buscar texto mediante Lucene.

Con este sistema se han generado tres nuevos recursos que contienen toda la información sobre las temáticas de los artículos de DBpedia, su información desambiguada y los grupos a los que pertenecen las palabras desambiguadas. Estos índices tienen un tamaño de entre 16 GB y 25 GB.

El proceso de generación de estos recursos ha sido el más costoso debido a la gran cantidad de datos a procesar y a la necesidad de ordenadores potentes para generar los índices. Los problemas que principalmente se han presentado al realizar esta parte del proceso han sido el procesamiento de los ficheros, el almacenamiento y la necesidad de utilizar maquinas un poco más potentes.

Llegados a este punto, se han generado seis nuevos recursos. Los tres primeros ya se describieron en los sistemas anteriores y los tres restantes son el resultado de este último sistema. Estos tres últimos contienen de la información de los artículos de DBpedia, su desambiguación y los grupos de las palabras desambiguadas de manera

indexada y con los vectores de SVM preparados para su posterior procesamiento, utilizando los campos descritos en la figura 3.12. Estos índices o bases de conocimiento, se crearon para apoyar las decisiones a la hora de seleccionar la temática o temáticas de una frase.

La idea era que para cada temática, las palabras ambiguas y sus definiciones, normalmente se repetirían. Entonces, estas bases de conocimiento podrían ayudar a mejorar los resultados de detectar la temática de las frases cortas.

Los resultados de la obtención de este conocimiento han servido para generar una metodología capaz de agrupar artículos por temática de manera automática y obtener bases de conocimiento construidas con los artículos de DBpedia y la información lingüística de sus palabras.

3.2.5.4 **Problemas con los nuevos índices**

Una vez que ya se obtuvieron los nuevos índices, se comenzó a realizar diferentes análisis de la eficacia de los índices en la categorización de frases cortas aleatorias. En este punto apareció el problema de la verificación. Actualmente no existe un dataset con los elementos contenidos en los índices, por lo que la verificación no se podía hacer de manera sencilla. La solución fue realizar un pequeño dataset con 40 frases obtenidas de una conversación mantenida entre dos miembros del laboratorio S3Lab de Deustotech.

El dataset era demasiado pequeño como para que los resultados pudieran ser relevantes, pero para realizar las primeras pruebas antes de intentar generar un dataset mayor, era válido.

El proceso seguido para esta tarea fué:

- Etiquetación de las conversaciones con topics existentes en los índices por un experto.
- Procesamiento de las conversaciones y autoetiquetación del sistema en base al índice de definiciones generado.
- Análisis de los topics obtenidos por el sistema automático.

Siguiendo el proceso, la etiquetación de las diferentes conversaciones por un experto fue una tarea muy laboriosa y complicada. El problema radicaba en conocer la definición de cada topic, el ámbito que abarcaba, las uniones entre muchos de ellos y a la gran cantidad de topics que se habían generado. Aun así, se etiquetaron esas 40 conversaciones.

El siguiente paso del proceso, el procesamiento de las conversaciones y la etiquetación automática, se realizó sin problemas. Se desarrolló un sistema para realizar búsquedas de términos dentro de los índices generados. Para la etiquetación, se realizaron dos pruebas. La primera de ellas utilizando la frase completa sin ningún tipo de filtro salvo mantener los caracteres alfanuméricos. La segunda prueba se hizo un preprocesado a las frases para eliminar las stopwords.

Por último, se analizaron los resultados obtenidos de ambas etiquetaciones. Los topics obtenidos para las frases sin preprocesado fueron muy difusos. La gran mayoría de los topics que se habían etiquetado para cada frase no abarcaban, para nada, su ámbito de contexto y, por lo cual, no eran válidos. Por otra parte, los etiquetados con el preprocesado de stopwords, a primera vista, la mayoría de los topics obtenidos con mayor peso no tenían demasiado sentido, pero tras realizar un análisis más completo a la frase y el contenido de los topics que había devuelto, si tenían un poco más de sentido. El problema acaecía en que muchos de los topics indexados tenían elementos del contexto de la frase, pero no eran el contenido real.

Un ejemplo, son las siguientes frases: "*my wife is gonna kill me*" y "*i like small cats and dogs*". Una vez preprocesadas, se obtenían las siguientes frases: "*my wife gonna kill me*" **(FraseA)** y "*i like small cats dogs*" **(FraseB)**. Los topics obtenidos de cada una, en formato *TOPIC RELEVANCIA*, se muestran en la tabla 3.7.

Como se puede apreciar en los resultados de la **FraseA**, el principal topic es el mundo de la música y la televisión y, después una entidad criminal. En principio esta clasificación podría afinarse más, pero los temas seguirían siendo poco relevantes. Por otra parte, en la **FraseB** el principal topic que se puede extraer es el deporte, después algo sobre animales, películas de dibujos y música y, por último, drogas.

La explicación a este tipo de fenómeno es que, para la **FraseA**, existen muchas canciones que hablan sobre matar a una mujer, lo mismo que películas y series. Para la **FraseB**, el significado es que muchas mascotas de equipos de futbol, tanto americano como europeo, así como de baloncesto, rugby, etc. son estos animales. La aparición de los topics de música y dibujos se debe a que muchos protagonistas de canciones o dibujos animados, son perros y gatos y, por último, el topic de drogas se debe a que muchos de los productos considerados drogas, son previamente probados en animales, principalmente perros y gatos.

Las conclusiones que se obtuvieron de este enfoque fueron bastante interesantes. Una de las más importantes era que no se tenia un recurso valido y etiquetado, respecto a nuestro sistema, para verificar su funcionamiento. Esto planteaba varios interrogantes. ¿Era factible crear un dataset etiquetado con todos los topics que teníamos? ¿Era más viable hacer encuestas ofreciendo los diferentes topics a los usuarios para etiquetar frases? ¿De qué tipo y fuente íbamos a sacar las frases para etiquetar? ¿Era suficiente con la generación de índices de DBpedia? Por otra parte,

FraseA *"my wife is gonna kill me"*	FraseB *"i like small cats and dogs"*
SONG 0.00936913	CANADIANFOOTBALLTEAM 0.02435189
MUSICRECORDING 0.00936913	MAMMAL 0.01919148
SINGLE 0.0088782795	NATIONALFOOTBALLLEAGUESEASON 0.018084105
CRIMINAL 0.008766842	HOLLYWOODCARTOON 0.01749864
MUSICALWORK 0.007769657	SPORTSTEAMSEASON 0.015606525
FICTIONALCHARACTER 0.0074815466	ARACHNID 0.015545097
TELEVISIONEPISODE 0.0074008442	DRUG 0.015013451
TVEPISODE 0.0074008442	CARTOON 0.014278983
MUSICALBUM 0.00724405	SONG 0.014191399
ALBUM 0.00724405	MUSICRECORDING 0.014191399

Tabla 3.7: Resultados de los topics extraidos de los indices generados con *SKOS*

la gran cantidad de topics que se tenía, hacía inviable una etiquetación correcta de las frases. Esto se debía a que era necesario conocer todos los aspectos que abarcaba cada topic para poder realizar una etiquetación correcta y, además, debido a la gran cantidad que había y las uniones entre ellos, se hacía inviable la etiquetación.

3.2.6 Enfoque 5, Artículos de DBpedia con nuevas etiquetas

Partiendo del enfoque anterior, en el cual se solucionaron problemas como la fuente de datos actualizable y la estandarización de los datos obtenidos, en este nuevo enfoque, uno de los objetivos era validar un nuevo tipo de idea para esquematizar mejor los topics y que abarcaran áreas más específicas. A la vez, también se intentó que no se solapasen entre sí los contenidos de dichas áreas. Para ello, se utilizó la ontología que ofrece DBpedia con sus recursos[22].

Una ontología nos ofrece un esquema conceptual muy riguroso dentro de los dominios, en este caso, los artículos que tiene DBpedia. Los términos incluidos deben ser parte de un vocabulario fijo, es decir, no debe dar motivo de duda la definición de dicho término. Además, estos deben describir su contenido y sus términos hijo de manera inequívoca.

Este tipo de recursos sirven para representar conocimiento mediante esquemas, crear una estructuración completa de datos que contiene todas las entidades relevantes y sus relaciones dentro del dominio. El objetivo de la ontología es facilitar la comunicación, el intercambio de información entre diferentes sistemas y entidades, ayudar al razonamiento inductivo y ayudar en problemas de clasificación.

El problema que surge es que, en ocasiones, las ontologías representan pobremente ciertos dominios. Debido a esto, se deben crear esquemas más especializados para convertir en útiles los datos que ofrecen. Otro problema que tienen las ontologías es la interpretación. Cada ontología intenta representar un dominio concreto, pero cada persona puede ver de diferente forma dicho dominio y ver invalida dicha ontología.

En el caso de DBpedia, su ontología está constituida por dominios cruzados de poco alcance. Esto significa que las relaciones entre clases son bastante claras y directas. Esta ontología ha sido creada de manera manual por la comunidad de Wikipedia, basándose en los “infoboxes” de los artículos. La comunidad de DBpedia ha puesto a disposición de todo el mundo un Wiki[23] para que todo el mundo pueda escribir asignaciones para los “infobox” de los artículos y, de esta manera, mantener el crecimiento de la ontología y el equilibrio de agradar a la mayoría. La ontología que se ha utilizado para este experimento es la referente a la versión 3.8 de las fuentes de

[22] mappings.dbpedia.org/server/ontology/classes/
[23] mappings.dbpedia.org/index.php/Main_Page

DBpedia[24].

Para este trabajo, se generó con la base de esta ontología, un recurso en forma de taxonomía que tenía registrados todos los padres de los elementos más especificos. Un ejemplo se pude ver en la figura 3.13

```
agent
{...}
agent-person
{...}
agent-person-athlete
{...}
agent-person-athlete-motorsportracer
{...}
agent-person-athlete-motorsportracer-motorcyclerider
{...}
agent-person-athlete-motorsportracer-motorcyclerider-speedwayrider
{...}
```

Figura 3.13: Formato del archivo con la estructura de la taxonomía.

El elemento más pequeño de la taxonomía y, a su vez el más específico, es el de color **purpura**. Este tiene como padre el color **naranja**, que a su vez tiene el padre de color **marrón**, que a su vez tiene el padre de color **cyan**, que a su vez tiene el padre de color **azul** y, como padre de todos está el color **rojo**. El elemento de color rojo es el más genérico y, en este caso corresponde al elemento "agent".

Además, también se realizó una descripción de las áreas que abarcaban cada topic. Esto supuso la lectura de gran cantidad de artículos de cada topic para poder describir mejor el ámbito de uso y el alcance de cada topic. En las tablas 3.8, 3.9, 3.10, 3.11, 3.12 y 3.13 se muestran los 34 topics que existen en la taxonomía junto con las descripciones generadas.

Para obtener los elementos de esta taxonomía, se desarrolló un sistema capaz de obtener, de manera automática, los valores de la taxonomía de DBpedia[25]. Este sistema almacena todos los nodos de la taxonomía y la relación entre ellos. Se almacena el nombre del nodo y el padre directo. Así por todos los nodos salvo el primero, que es cosa, en inglés "*thing*", y que no tiene padre. El recurso que obtuvimos es el mostrado en la figura 3.13.

La etiquetación automática se realizó utilizando sistemas previamente diseñados en enfoques anteriores. Se creó un índice con Lucene, utilizando el contenedor mostrado en la figura 3.12 del apartado 3.2.5.3 con el mismo tipo de información. La *CLASE* era el topic y la *INFO* era la información que contenia dicho topic.

En este caso, la creación sufrió unas pequeñas alteraciones. En primer lugar se realizó un sistema para unir la información de los nodos sin hijos con los nodos de

[24] mappings.dbpedia.org/server/ontology/classes/
[25] mappings.dbpedia.org/server/ontology/classes/

Etiqueta	Descripción
Ninguno	Se usa cuando el topic de la frase no coincide con ningún elemento de la lista o simplemente no tiene ningún topic.
Activity	Se usa cuando la frase contiene algún tipo de actividad relacionada **unicamente** con **juegos** o **deportes menos el billar**. Los juegos a los que se refiere son de tablero, de jugar en el patio del recreo (cartas, lego, warhammer, rol). En lo relativo al deporte, se refiere a las acciones que se ejecutan al realizar el deporte (chutar, cazar). También entra en esta categoría el equipamiento que se usa, tanto en los juegos como en los deportes.
Agent	Se usa para describir a una entidad, ya sea persona u organización. Cualquier tipo de organización o agrupación de personas y sus actividades entran dentro de esta categoría. También entra las acciones que esas personas puedan realizar. Por ejemplo, matar podría ser de asesino (**agent/person**), lo relacionado con juicios podría ser de juez (**agent/person**). Por la parte de las organizaciones, las acciones que llevan a cabo dentro de su actividad empresarial también pueden ser marcadas. Una mujer, un niño, una madre, un padre entran dentro del tipo (**agent/person**).
AnatomicalStructure	Se usa con todo lo relacionado con partes **fisicas** del cuerpo. Ya sean elementos **internos** o **externos** como, por ejemplo hígado, brazo, arterias, huesos, ligamentos, nervios, venas, cerebro, músculos.
Award	Se usa cuando se hace mención con elementos relacionados con premios físicos y sus entregas. Entra todo tipo de entrega de premios y también los conocidos como pueden ser los Grammys, los Oscar, los Nobel.

Tabla 3.8: Descripciones de los topics de nuestra taxonomía (1/6)

Etiqueta	Descripción
CareerStation	Se usa con todo lo relacionado con etapas en la carrera profesional de una persona conocida o famosa.
CelestialBody	Se usa cuando se hace referencia a cuerpos celestes como planetas, estrellas, galaxias, cometas estelares, asteroides. Si a una persona, animal, cosa se le dice que es como un sol, entraría dentro de esta categoría. Si se refiere a la Tierra (nuestro planeta) también entra dentro de esta categoría.
ChemicalSubstance	Se usa con todo lo relacionado con elementos químicos, sus compuestos, así como minerales, gases, líquidos que sean compuestos por elementos de la tabla periódica. El agua y alcohol también pueden entrar aquí.
Colour	Se usa con todo lo relacionado con los colores, su descripción, definiciones y los diferentes tipos de colores que existen.
Currency	Se usa con todo lo relacionado con las monedas existentes en el mundo. Dólares, euros, francos, libras y demás tipos de monedas del mundo. Esta categoría no tiene que ver con los bancos, solo con monedas o unidad monetaria que existen en cada país. Los bancos serían organizaciones (**agent/organisation**) y los banqueros serían personas (**agent/person**).
DataBase	Se usa con todo lo relacionado con las diferentes bases de datos BD existentes en el mundo. Estas BD no son del tipo Oracle, MySQL o semejantes, son BD de consultas que han sido rellenadas por personas o empresas. Principalmente hay metida información sobre elementos biológicos (bacterias, virus), familias de esos elementos biológicos y sus componentes.

Tabla 3.9: Descripciones de los topics de nuestra taxonomía (2/6)

Etiqueta	Descripción
Device	Se usa con todo lo relacionado con diferentes dispositivos, entre los que se encuentran armas (**solo las armas como instrumento, no la persona que las usa**), dispositivos eléctricos y electrónicos (**móviles, gadgets, consolas, televisiones, cámaras, entran las fotos**) y lo relacionado con los motores de combustión (**solo los motores, los vehiculos no**).
Disease	Se usa con todo lo relacionado con cualquier tipo de enfermedad, dolencia o estado enfermizo, ya sea físico o mental.
Drug	Se usa con todo lo relacionado con cualquier tipo de droga no legalizada o que sea necesario una receta o permiso específico para su obtención. Entran medicamentos, analgésicos, drogas comunes y de diseño.
EthnicGroup	Se usa con todo lo relacionado con los diferentes tipos de grupos de personas que existen en el mundo. Estos grupos van desde los gitanos, cristianos árabes (**personas que viven en zonas árabes y que son cristianos**), sherpas hasta los amish. No entran los grupos de personas que son tratadas diferente por tener una enfermedad.
Event	Se usa con todo lo relacionado con un evento, ya sea social, deportivo, musical, cinematográfico, una convención, conflicto militar, misión espacial, o cualquier evento que conlleve una gran expectación o un importante peso histórico. Los eventos que son comunes tales como las bodas, cenas de gala, entrega de premios, huelgas entran en esta categoría.
Food	Se usa con todo lo relacionado con los diferentes tipos de comida, así como con los ingredientes, elementos que la componen (**sin ser los cubiertos ni los espacios**), y la preparación de estos. También entra la bebida como los zumos, bebidas alcohólicas y cocteles.

Tabla 3.10: Descripciones de los topics de nuestra taxonomía (3/6)

Etiqueta	Descripción
GeneLocation	Se usa para los elementos relacionados con los diferentes genes existentes dentro de las cadenas de ADN, tanto en humanos como en ratones.
Holiday	Se usa con todo lo relacionado con las diferentes fiestas en todo el mundo. Por ejemplo el día de San Patricio en Boston. También entran las épocas de fiestas como navidad, fines de semana o vacaciones.
Language	Se usa con todo lo relacionado con los diferentes lenguajes naturales (**idiomas**) existentes en el mundo, aunque sean inventados como el klingon. Los lenguajes de programación no entran porque no son naturales.
MeanOfTransportation	Se usa con todo lo relacionado con cualquier tipo de viaje, excursión o desplazamiento que conlleve viajar. También entran los diferentes tipos de transporte y los modelos existentes, como por ejemplo avión y el modelo boeing, tren, coche, barco, nave espacial, cohete.
Name	Se usa cuando aparece el nombre propio de una persona y con todo lo relacionado con nombres propios. Aquí entra también los nombres personales existentes.
PersonFunction	Se usa cuando se describe una actividad concreta relacionada con el trabajo o actividades laborales de una persona.
Sales	Se usa con los elementos relacionados una transacción comercial. También entran elementos como rebajas o descuentos.
SnookerWorldRanking	Se usa con todo lo relacionado con el mundo del billar y sus jugadores. El billar, aunque sea también un deporte (**activity/sport**) no entra en esa categoría.

Tabla 3.11: Descripciones de los topics de nuestra taxonomía (4-6)

Etiqueta	Descripción
SportCompetitionResult	Se usa para referirse a competiciones olímpicas.
Place	Se usa con todo lo relacionado con lugares. También con estructuras arquitecturales como edificios, castillos, hospitales, hoteles, faros, museos, restaurantes, rascacielos. También con infraestructuras como aeropuertos, plataformas de lanzamiento, estaciones eléctricas, estaciones de tren, bus, metro y demás. Los tipos de carreteras como puentes, autopistas, vías, túneles, cruces. Los parques, lugares históricos, monumentos, sitios naturales, reservas, cuevas, ríos, montañas, valles, lagos, volcanes, lugares poblados, ciudades, continentes, regiones, islas, pueblos, villas, lugares con intereses turísticos o científicos, zonas de esquí, estadios, hipódromos, zonas de vendimia y zonas de cultura o patrimonio cultural.
Species	Se usa con todo lo relacionado con cualquier tipo de animal que no sea el ser humano. Las plantas, animales, bacterias, microorganismos, insectos y toda la fauna que existe en el planeta Tierra. Frutas como los plátanos son comida, pero también son plantas.
SportsSeason	Se usa cuando se hacen referencia a las competiciones que son de temporada como las ligas, la Champions, la copa de la UEFA, Formula 1, Nascar y todo lo que tenga que ver con los equipos en esas temporadas. También entran las ligas escolares y universitarias.
TimePeriod	Se usa todo lo referente a periodos de tiempo como días, meses, años, décadas, siglos, años. En principio las horas y minutos también pueden entrar en esta categoría.

Tabla 3.12: Descripciones de los topics de nuestra taxonomía (5/6)

Etiqueta	Descripción
TopicalConcept	Se usa para todo lo relacionado con la música, sus diferentes estilos, géneros, tipos, instrumentos.
Biomolecule	Se usa con todo lo relacionado con elementos, tanto de investigación como de uso normal, en el área de la medicina. Las biomoleculas, biomateriales, genes de (**humanos y de ratones**), enzimas, proteínas.
UnitOfWork	Se usa esta categoría para referirse a casos judiciales, legales o proyectos. Los casos judiciales son de cualquier tipo y los proyectos también. En esta categoria entran los proyectos de investigación.
Work	Se usa esta categoría con todo lo relacionado con el trabajo, independientemente del tipo. También entran los elementos como revistas y comics como trabajo escrito, el desarrollo de aplicaciones, los dibujos, las películas, los episodios de series y los shows, las webs, los libros. Todo lo que sea un trabajo remunerado.

Tabla 3.13: Descripciones de los topics de nuestra taxonomía (6/6)

sus padres directos por cada topic. Es decir, se toma un topic, por ejemplo *agent*, y se navega hasta los nodos hoja (los que no tienen hijos) de esta categoría. Una vez en ellos, se obtiene toda la información que contiene y se vuelca en el nodo padre directo, sumándose a la información que este ya contenia. Una vez volcada esta información, se pasa al nodo padre, convirtiéndose este de manera momentánea en el nodo hijo. Se verifica que este nodo no es el padre de esta categoría, es decir el nodo "agent". Si no lo es, se vuelve a repetir el mismo proceso hasta llegar al nodo padre de la categoría. En la figura 3.14 se puede apreciar el flujo de este proceso.

Los elementos en azul muestran topics de la taxonomía en herencia natural. Estos están unidos a su hijo por la flecha negra. Los elementos naranja son la información nueva generada, resultado de la suma de los topics previos. Cada elemento naranja provee información para crear el índice de ese nivel. Por último, el resultado final en el padre, es decir, la suma de toda la información de los hijos más la propia del padre, es la información del nivel inicial. Este último nivel contiene toda la información de todos los hijos de los topics iniciales.

El objetivo de este proceso era crear índices de niveles para poder hacer búsquedas más eficientes por elementos relacionados. Además de mejorar en el filtrado de búsquedas y desechar elementos que no contienen resultados óptimos para nuestra búsqueda.

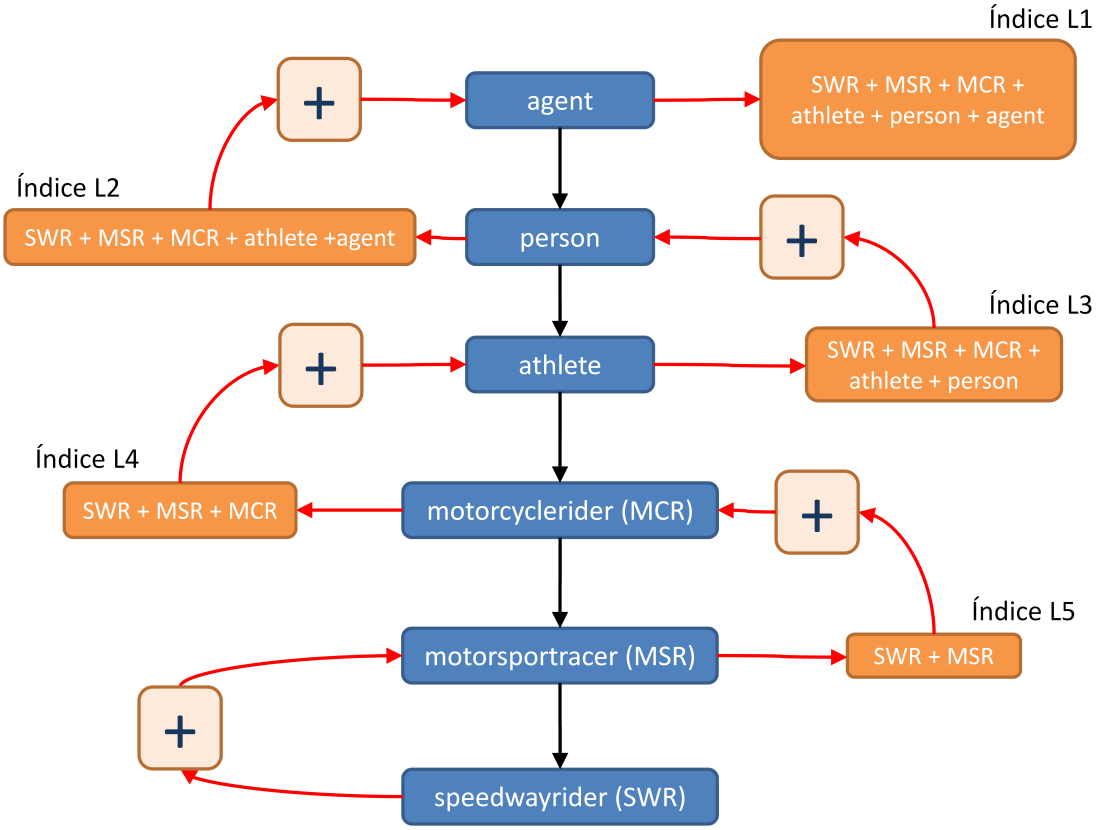


Figura 3.14: Flujo de generación de índices por niveles.

Se generaron tantos índices como niveles de nodos hijos tenían los topics. El resultado final fueron 92 índices, contando el nivel 0 con la unión de todos los topics principales, y una profundidad de 5 niveles.

Una vez creados todos los índices, se utilizó el sistema para obtener topics de índices, desarrollado en el aparatado 3.2.5.4, para etiquetar ciertas frase de control que habían sido utilizadas en anteriores enfoques para conocer la calidad del recurso creado. Estas frases, no se muestran debido a que, como luego se verá, generamos 2 nuevos recursos o "datasets" para poder evaluar los índices y los algoritmos.

Los resultados que se obtuvieron fueron buenos. Esta valoración estaba realizada a juicio del mismo evaluador que había estado realizando las valoraciones de los anteriores enfoques. Además, el número de topics había sido reducido y etiquetado para poder determinar, de una manera más sencilla, el ámbito de uso de cada uno de ellos. Esto permitía poder generar un recurso etiquetado manualmente por expertos para evaluar si realmente era efectiva la base de conocimiento creada.

Finalmente, viendo que se había conseguido un recurso que satisfacía las necesidades que inicialmente teníamos, se aceptó esta forma de actuar y el recurso generado para realizar los experimentos de este trabajo.

3.2.7 **Enfoque 6, Categorías de productos de Amazon**

Una vez que ya se había obtenido una metodología, que se había decidido que era eficiente y que resultaba útil para nuestros propósitos, quedaba generar la parte de marketing para brindar productos a los usuarios en base al contexto de sus conversaciones cortas.

Para escoger la fuente de datos, pensamos en empresas, supermercados, tiendas y cadenas de grandes almacenes entre otros. Estas entidades las desechamos debido a varios factores: no prestan demasiada atención a su web, los productos no están bien ordenados, no existen definiciones aceptables de los productos, utilizan tecnologías que impiden obtener datos, etc. Finalmente, la fuente de datos elegida para este proceso fue Amazon. Esta tienda online tiene más de 300 millones de productos a la venta[26], permite a los clientes comentar los productos que han comprado, evaluarlos y, además, muchos de los productos incluyen un resumen o descripción de lo que contienen o son. También viene información de en qué puede ser utilizado cada producto, una foto, etc. Por otra parte, en la comunidad científica existen varios trabajos que utilizan Amazon como fuente para datos. La mayoría están basados en el análisis de opinión en los comentarios como, por ejemplo, trabajos como el de Tsur et al. (2010) que estudian el sarcasmo, Blitzer et al. (2007) estudia los sentimientos, Pang et al. (2002) que estudia la polaridad de las opiniones o Mooney y Roy (2000),

[26] askville.amazon.com/items-sale-Amazon/AnswerViewer.do?requestId=10285231

que estudia las recomendaciones. Pero no todo se basa en el estudio de lo que dicen y como lo dicen, otros como Lim et al. (2010) se centran en la identificación de "*spammers*" o Chong et al. (2015) utilizan el "BigData" con las revisiones de los usuarios de Amazon para predecir las demandas de productos según las promociones.

Como el catálogo de productos de Amazon es demasiado grande y obtener toda la información es una tarea complicada, se decidió acotar el área de los productos a la tienda de libros de Amazon en inglés [27]. El motivo por el cual se seleccionó esta gama de productos se debe a que, esta tienda posee millones de ejemplares diferentes y, además de estar dividido en categorías, existe un libro para cada situación que pueda surgir en la vida cotidiana. Se buscaron datasets" o fuentes de datos de Amazon que contuvieran de manera específica o semejante, información que fuera útil como por ejemplo, de cada libro, la categoría a la que pertenecía y la descripción del libro. Se empezó por Amazon Web Services Data Sets[28], Common Crawl[29], SNAP[30] de la Universidad de Stanford, *Intitut of Informatics from Freiburg* IFF[31], *School of Information & Computer Sciences* (UCI)[32] y en Google Books[33], pero ninguno de estos recursos contenía la información necesaria para el trabajo. La mejor solución que se encontró fue crear una fuente propia de datos o "dataset". Este recurso debería de contener los resúmenes que provee Amazon sobre cada libro, así como la categoría a la que pertenece.

El recurso que se creó, tenía la estructura de árbol que posee la tienda de libros de Amazon con las categorías. En esta estructura están contempladas todas las categorías de libros desde "*Arts & Photography*" hasta "*Travel*" y sus correspondientes subcategorías, sumando un total de 31 categorías y más de 4000 subcategorías. Cada nodo del árbol generado contiene todos los libros de esa categoría específica junto con sus resúmenes de Amazon y la información de sus sub-nodos. Cada categoría posee información de entre 8 y 1600 libros. Con este enfoque de granularidad, se consigue aumentar la precisión para poder detectar los libros objetivos. La idea era obtener un recurso que se asemejase a la taxonomía obtenida para DBpedia para poder utilizar la misma metodología. En la figura 3.15 se muestra la taxonomía que hemos generado, a partir de las categorías de la tienda de libros de Amazon.

Al igual que con la taxonomía que se había generado de DBpedia, el elemento más pequeño de la taxonomía y, a su vez el más específico, es el de color **purpura**. Este tiene como padre el color **naranja**, que a su vez tiene el padre de color **marrón**,

[27] www.amazon.com/books-used-books-textbooks/b/ref=sd_allcat_bo?ie=UTF8&node=283155
[28] aws.amazon.com/datasets/
[29] commoncrawl.org
[30] snap.stanford.edu/data/amazon-meta.html
[31] www2.informatik.uni-freiburg.de/ cziegler/BX/
[32] archive.ics.uci.edu/ml/datasets.html
[33] storage.googleapis.com/books/ngrams/books/datasetsv2.html

```
science_&_math
{...}
science_&_math-nature_&_ecology
{...}
science_&_math-nature_&_ecology-conservation
{...}
science_&_math-nature_&_ecology-conservation-energy
{...}
science_&_math-nature_&_ecology-conservation-energy-alternative_&_renewable
{...}
science_&_math-nature_&_ecology-conservation-energy-alternative_&_renewable
-hydroelectric
{...}
```

Figura 3.15: Taxonomía generada de las categorías de productos de Amazon.

que a su vez tiene el padre de color **cyan**, que a su vez tiene el padre de color **azul** y, como padre de todos está el color **rojo**. El elemento de color rojo es el más genérico y, en este caso corresponde a la categoría "science_&_math".

Además, también se realizó una descripción de las áreas que abarcaban cada categoría. Esto supuso la revisión de todos los contenidos de todas las categorías para poder describir mejor el ámbito de uso y el alcance de cada categoría. En las tablas 3.14, 3.15, 3.16, 3.17 y 3.18 se muestran las 32 categorías que existen en la taxonomía junto con las descripciones generadas.

Para obtener la taxonomía de Amazon que se ha mostrado, y la información que necesitaba la base de conocimiento de productos, se diseñó un sistema capaz de recoger todas las categorías de libros de la tienda de Amazon, así como sus subcategorías y libros que contenían. La jerarquía que se obtuvo era semejante a la de DBpedia, por eso se pudo utilizar la taxonomía igual.

Para el proceso de indexación, utilizamos la metodología y herramientas creadas para DBpedia con el mismo formato y estructura en los recursos, pero con la información de las categorías de Amazon. Como el sistema ya estaba creado, su generación fue sencilla. Se siguieron los mismos pasos que para la generación de los índices de DBpedia descrita en la sección 3.2.6. Una vez generado los índices, las pruebas que se hicieron resultaron satisfactorias. Estas pruebas se realizaron con la misma persona y los mismos recursos que para DBpedia.

3.2.8 Motivación para realizar las búsquedas dentro de los indices creados

Los indices generados y descritos en las secciones 3.2.6 y 3.2.7, contienen los elementos mas representativos o importantes dentro de su categoría. Por ejemplo, y tomando como referencia un elemento de DBpedia, en la categoría de mamíferos, se encuentran todos los elementos referentes a los animales que son mamíferos co-

Etiqueta	Descripción
Ninguno	Se usa cuando el topic de la frase no coincide con ningún elemento de la lista o simplemente no tiene ningún topic.
Arts_&_Photography	Se usa cuando la frase contiene algún tipo de relación con artes, ya sea por arquitectura o edificios, como artes de negocios, exhibiciones, catálogos de arte, decoraciones, diseño de elementos, Dibujos, cuadros, moda, historia del arte, criticas del arte, artistas como pintores o arquitectos, la música, cualquier tipo de elemento multimedia que se considere arte, pintura, artes escénicas, fotografía y video, elementos del arte religioso, esculturas enseñanza del arte y multimedia y vehículos (cualquier tipo, aviones barcos, coches, camiones que sean considerados arte).
Biographies_&_Memoirs	Se usa para describir a una persona y su vida en una biografía. Aquí entran las memorias de la gente, profesionales académicos, líderes mundiales, gente histórica, líderes étnicos y nacionales, exploradores y viajeros, criminales, deportistas.
Business_&_Money	Se usa con todo lo relacionado con los negocios, finanzas, economía, negocios, gestión de personal, inversiones, bolsa, cuentas millonarias.
Calendars	Se usa cuando se hace mención con fechas o calendarios, los calendarios contienen animales, coches, películas, mapas, música, elementos naturales, fotografías, etc. A todo el mundo se le puede regalar un calendario con imágenes sobre un tema específico.
Children's_Books	Se usa con todo lo relacionado aprendizaje de niños en las diferentes áreas. Explicación de la historia a los niños, la ciencia, los deportes, el humor, geografía, comics, actividades, juegos, aventuras, folclore, literatura y ciencia ficción, religión.
Christian_Books_&_Bibles	Se usa con todo lo relacionado con la religión cristiana, sus libros, sus sectas, su educación, su historia, sus ministerios, la literatura, la teología, las devociones, las iglesias, la biblia, biografías de personas de la iglesia cristiana.
Comics_&_Graphic_Novels	Se usa cuando se hace referencia a comics, novelas gráficas, ciencia ficción, manga, superhéroes, arte manga y arte del comic.

Tabla 3.14: Descripciones de las categorías de nuestra taxonomía (1/5)

Etiqueta	Descripción
Computers_&_Technology	Se usa con todo lo relacionado con la tecnología, lenguajes de programación, dispositivos electrónicos y eléctricos, internet, web, redes sociales, empresas tecnológicas, móviles, tablets, software, ciencias de la computación, videojuegos, bases de datos, certificaciones informáticas, sistemas operativos, seguridad informática, encriptación y securización, redes, hardware, etc.
Cookbooks,_Food_&_Wine	Se usa con todo lo relacionado con la comida, métodos de cocinar, productos para la comida, postres, diferentes comidas del mundo, dietas, vegetarianos, veganos, tipos de platos, ingredientes, etc.
Crafts,_Hobbies_&_Home	Se usa con todo lo relacionado con hobbies, entretenimiento, diseño y mantenimiento de jardines, mascotas, bodas, antigüedades, coleccionables, vida sostenible en casa, mejoras en la casa y rediseño, etc.
Education_&_Teaching	Se usa con todo lo relacionado con la educación primaria, secundaria, preparación de exámenes, universidad, libros de clase, etc.
Engineering_&_Transportation	Se usa con todo lo relacionado con el transporte y la ingeniería de los métodos de transporte. El coche, el metro, el tren, el avión, los barcos, motos. También entra la ingeniería como la aeroespacial, la química, la civil, la biológica, la construcción, las telecomunicaciones, la producción de energía, la industria manufacturera, la ingeniería naval, la ingeniería militar, los elementos eléctricos y electrónicos, el diseño y los materiales, etc.

Tabla 3.15: Descripciones de las categorías de nuestra taxonomía (2/5)

Etiqueta	Descripción
Gay_&_Lesbian	Se usa con todo lo relacionado con aspectos relacionados con la sexualidad de las personas gays y lesbianas, su historia, biografías, viajes específicos, complejidades y problemas familiares y literatura especifica de este tema, etc.
Health,_Fitness_&_Dieting	Se usa con todo lo relacionado con la salud, adicciones y recuperación, medicinas alternativas, estilos de vida, belleza, salud, ejercicio, salud mental, salud sexual, seguridad en la salud, primeros auxilios, desórdenes alimenticios y psicológicos, nutrición, etc.
History	Se usa con todo lo relacionado con la historia. La historia de cualquiera de los 5 continentes, de cualquier país, de cualquier hecho histórico. Las culturas extinguidas, antiguas civilizaciones, y su estudio, etc.
Humor_&_Entertainment	Se usa con todo lo relacionado con el humor. Diferentes temas sobre el humor, risas. El humor en las películas, el las comedias, en el teatro, la cultura pop, juegos de humor, radio de humor, televisión, música, etc.
Law	Se usa con todo lo relacionado con las leyes. Terminología y diccionarios sobre leyes, leyes administrativas, biografías de gente de leyes, negocios entorno a la ley, leyes constitucionales, leyes criminales, leyes medioambientales, testamentos, responsabilidad legal, ética legal, leyes internacionales y nacionales, leyes educacionales, practicas legales, historia de las leyes, sistemas legales, filosofía, procedimientos y reglas, especializaciones legales, tasas legales, etc.
Literature_&_Fiction	Se usa con todo lo relacionado con la literatura antigua, clásica contemporánea, drama, erótica, ensayos, ciencia ficción, géneros de ciencia ficción, lenguajes de la ciencia ficción (klingon), mitología, poesía, historias cortas, etc.
Medical_Books	Se usa con todo lo relacionado con temas relacionados con medicina como su gestión, administración, trabajos dependientes de la medicina (ambulancias, dietistas, asistentes médicos, radiología), historia de la medicina, enfermería, sistemas informáticos médicos, farmacia, psicología, investigación médica, veterinaria, etc.

Tabla 3.16: Descripciones de las categorías de nuestra taxonomía (3/5)

Etiqueta	Descripción
Mystery,_Thriller_&_Suspense	Se usa con todo lo relacionado con el misterio, el suspense, elementos supernaturales, detectives, crímenes, espías, psicológicos, etc.
Parenting_&_Relationships	Se usa con todo lo relacionado con relaciones dentro de la familia. Adopciones, padres de acogida, actividades familiares, salud familiar, fertilidad, relaciones entre los componentes de la familia, necesidades especiales, matrimonio, embarazos, relaciones adultas, nacimientos.
Politics_&_Social_Sciences	Se usa con todo lo relacionado con la política, gobiernos, gobernantes, movimientos sociales, ideologías, filosofía, antropología, arqueología, ciencias sociales, sociología, etc.
Reference	Se usa con todo lo relacionado con referencias de mapas, albaranes, catálogos y directorios, etiquetas, enciclopedias, tesauros, diccionarios, aprendizaje de otros idiomas, genealogía, referencias (citas), etc.
Religion_&_Spirituality	Se usa con todo lo relacionado con todo lo relacionado con la religión, sea cual sea, sus cultos, sus historias, todo lo relacionado con la religión y sus acciones como rezos, lapidamiento, etc.
Romance	Se usa con todo lo relacionado con el romance, historias de amor, de cualquier tipo. Gay, lesbiano, militar, paranormal, de vampiros, de ciencia ficción, comedia, gótica, multicultural.
Science_&_Math	Se usa con todo lo relacionado con la ciencia. La de agricultura, arqueología, astronomía y ciencias del espacio, ciencias del comportamiento (antropología y psicología), biología, química, ciencias de la tierra y del medio ambiente, ensayos sobre ciencia y matemáticas, ciencia de la evolución, experimentación y medición de datos, ciencias de la historia y la filosofía, las matemáticas, las ciencias naturales y la ecología, ciencias de la física (gravedad, acústica, nanoestructuras, física nuclear, microscopios electrónicos, ciencias aplicadas), ciencia para niños y tecnología, etc.

Tabla 3.17: Descripciones de las categorías de nuestra taxonomía (4/5)

Etiqueta	Descripción
Science_Fiction_&_Fantasy	Se usa con todo lo relacionado con elementos de la ciencia ficción y la fantasía. Abarca elementos mágicos, paranormales, de superhéroes, juegos de rol, estrategia fantástica, warhammer, dragones y mazmorras y ese tipo de juegos.
Self-Help	Se usa con todo lo relacionado con la autoayuda y los trastornos. Abusos, gestión de la ira, ansias y fobias, creatividad, miedo a la muerte, sueños y sus significados, desórdenes alimenticios, hipnosis, análisis de la escritura, gestión del niño interior, mejora de memoria, ayuda en el sexo, transformación personal y espiritual, gestión del stress y gestión de sucesos traumáticos, etc.
Sports_&_Outdoors	Se usa con todo lo relacionado con los deportes al aire libre. Béisbol, baloncesto, futbol, biografías de deportistas, entrenadores, como entrenar, deportes extremos, pesca, caza, pájaros, avistamiento de animales salvajes y de insectos. Todo lo que se puede hacer al aire libre. También entra el avistamiento de planetas, sus satélites, los tipos de rocas y minerales, los árboles, las flores. Golf, montañismo, deportes individuales, viajes naturales (viajes para ver paisajes o cosas por el estilo), supervivencia, deportes acuáticos y de invierno, etc.
Teen_&_Young_Adult	Se usa con todo lo relacionado con biografías de niños y adolescentes, su educación, salud (mental y física), su cuerpo, religión para los niños, problemas sociológicos, deportes, hobbies, literatura, etc.
Travel	Se usa con todo lo relacionado con los viajes. Cualquier tipo de viaje a cualquier parte del mundo. Viajes de comida, métodos en los que se transportan, viajes especiales como los de luna de miel y todos los países que hay para ver en los viajes, etc.

Tabla 3.18: Descripciones de las categorías de nuestra taxonomía (5/5)

mo pelo, hocico, glándulas mamales, sangre caliente, marsupial, gato, perro, patas, aletas, etc. Como puede comprobarse, algunos de los elementos que contiene esta categoría pueden estar compartidos con otras categorías, pero la selección de los términos únicos y la unión de los elementos comunes, filtran por si solos los resultados. El uso de estos términos de manera única ha sido muy utilizado en numerosas ocasiones, pero después de obtener los términos importantes, es necesario hacer un análisis de los mismos para poder obtener el ente al que se refieren. Ademas, un contexto determinado como pueden ser los mamíferos abarca muchos mas elementos que un gato, sangre caliente y todos los términos nombrados anteriormente. Utilizando estos contextos, se reduce la etiquetación de cada termino y se agrupan términos afines en una sola categoría. Por otra parte, la utilización de este sistema de contextos no implica que no se puedan utilizar la categorización de los términos, junto con los propios términos, en futuros experimentos. Por estos motivos se optó por utilizar la categorización de DBpedia para disminuir este análisis final y poder obtener los entes a los que se refieren en cada frase.

3.3 Creación del mapa relacional entre Amazon y DBpedia

Tener un elemento que vincule los términos existentes entre 2 bases de conocimiento es un recurso muy importante e interesante. Importante porque es una forma de vincular dos elementos que contienen un significado o referencia común con una base determinada. Interesante porque teniendo un recurso que vincule dos bases de conocimiento muy extensas se pueden realizar muchos experimentos con diferentes enfoques, metodologías y procesos. Por estos motivos, se ha creado un mapa que vincule los topics de DBpedia con las categorías de los libros de la tienda de Amazon.

3.3.1 Etiquetación experta

Al igual que en la generación de la base de conocimiento, para crear el mapa con las relaciones entre Amazon y DBpedia, era necesario tener identificado todas las categorías y las definiciones de éstas antes de empezar a etiquetar. Para ello, se utilizaron las taxonomías que se habían generado previamente, las cuales se muestran en las figuras 3.13 para DBpedia y 3.15 para Amazon, y las definiciones que se muestran en las tablas 3.8, 3.9, 3.10, 3.11, 3.12 y 3.13 para DBpedia y en las tablas 3.14, 3.15, 3.16, 3.17 y 3.18 para Amazon.

Para generar el mapa, se partió de las clases de DBpedia para etiquetar las de Amazon. El motivo es que DBpedia tiene menos clases y es más sencillo conocer su

contenido. Se extrajo únicamente el texto y se generó un dataset con estas frases. La idea era etiquetar cada una de las categorías de Amazon con los topics de DBpedia.

El resultado de la tarea de etiquetación fue un nuevo recurso, denominado mapa de relaciones que contiene tres campos. El primero indica si la línea está procesada, el segundo identifica la categoría de Amazon y, por último, en el tercero se incluyen los topics que el experto define para dicha categoría. Se puede ver un ejemplo de elemento etiquetado en la figura 3.16. En él, el color rojo corresponde a la categoría de Amazon y el color azul a los elementos etiquetados de DBpedia, en base al criterio del experto, para la categoría actual de Amazon.

```
True; medical_books medical_books[*-*]veterinary_medicine;
medician;disease;educationalinstitution;comicscharacter;university;
scientist;book;library;writtenwork;non_profitorganisation;comicscreator;
academicjournal;drug;collegeoruniversity;fictionalcharacter;publisher;
writer;periodicalliterature;educationalorganization;hospital
```

Figura 3.16: Labelled sentences examples.

Las categorías de Amazon contienen toda la ruta de la categoría debido a que existen varias subcategorías con el mismo nombre. De esta manera cada categoría y subcategoría queda perfectamente identificada inequívocamente, manteniendo la coherencia con la categoría principal y el resto de subcategorías.

3.3.2 Sistema de etiquetación manual

Al igual que con anteriores recursos, la generación de este mapa requirió bastante tiempo debido a la cantidad de datos. Para aliviar un poco esta tarea, se generó un sistema de etiquetación para que el experto pudiera etiquetar de manera más sencilla todas las clases.

El sistema constaba de 4 apartados como puede verse en la figura 3.17. El primer apartado, señalado como A, es donde se muestra cada una de las categorías de Amazon y que en el recurso final etiquetado (véase figura 3.16) tiene el color rojo. En el apartado B está toda la estructura, en forma de árbol, de los topics disponibles de DBpedia para etiquetar la categoría que se muestra en el apartado A. Estos topics son seleccionables. El apartado C es el buscador de topics de DBpedia. Sirve para buscar topics en base a sus letras. El objetivo de este apartado es facilitar la búsqueda de elementos para la etiquetación. El apartado D es en el que se quedan reflejados los topics seleccionados y que van a ser los que queden etiquetados en el recurso final. En la figura 3.16 estos elementos tienen el color azul.

El recurso generado contiene las 4227 categorías de libros de Amazon vinculadas con los 392 topics de DBpedia. Existen topics que no tienen ninguna unión debido

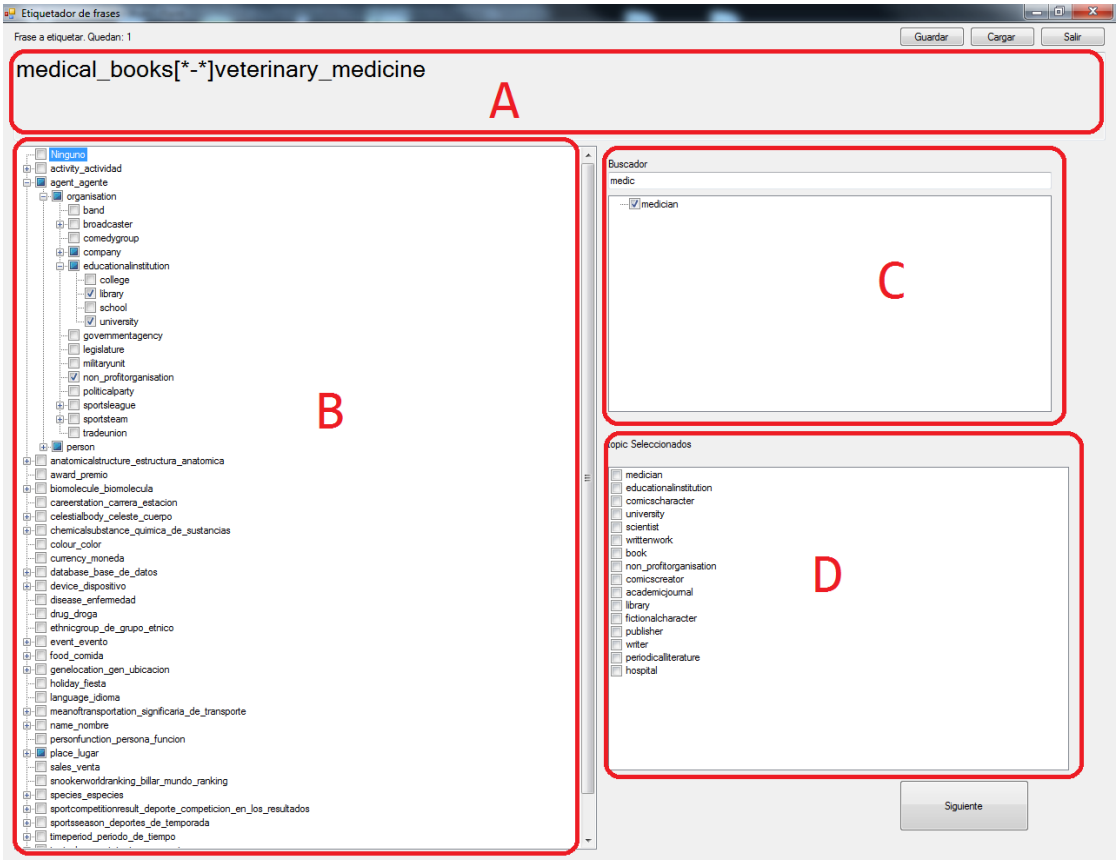


Figura 3.17: Labelling program.

a que no coinciden o entran dentro del ámbito, en base al juicio del experto, de las categorías de Amazon, pero el número de estos elementos no supera los 40 topics.

"No malgastaré la tiza"

Bartholomew JoJo Simpson
(1978 –)

4

Investigación

Una vez llegados a este punto, hemos creado una base de conocimiento que contiene contextos y productos. Además, también hemos creado un mapa que une ambas partes. Ahora se describirán los experimentos que hemos realizado para intentar responder a la pregunta de si estos nuevos recursos son útiles para validar la hipótesis inicial. En este apartado se define la metodología que se ha seguido para generar los datasets que nos han ayudado en la experimentación. Además, se exponen detalladamente los experimentos realizados. El primero de ellos es el experimento base o baseline, el segundo utiliza buscadores web para ampliar el contexto, el tercero utiliza tecnología de "*DeepLearning*" para ampliar el contexto con la ayuda del mapa de traducción entre recursos de DBpedia y Amazon y, por último, el experimento de traducción directa de términos de DBpedia a Amazon utilizando el mapa generado y sin aplicar ninguna otra técnica en el preprocesado y postprocesado.

4.1 **Baseline**

Al ser esta investigación un trabajo con nuevos recursos y procesos, es necesario partir de un punto inicial evaluable para poder valorar si las opciones propuestas realmente son viables para mejorar el sistema y explorar nuevas opciones de investigación. Como punto de inicio o "*baseline*" de nuestro trabajo, hemos realizado un experimento sencillo con la base de conocimiento generada. Para la validación hemos generado unos recursos etiquetados y unas métricas que evalúan el funcionamiento de los algoritmos propuestos.

4.1.1 Nuevos recursos etiquetados

Uno de los objetivos, siendo el más importante desde el punto de vista científico, era el de obtener un recurso que sirviera para validar la investigación que se estaba realizando. Estos recursos, llamados datasets, son elementos que contienen muestras de ejemplo etiquetadas. Para los trabajos de investigación, estos recursos son muy importantes debido a que se utilizan para verificar si una idea, enfoque, técnica o algoritmo tiene un comportamiento adecuado dentro de un área específica. Los datasets son generados normalmente por personas de manera manual. De esta manera, los procesos que se quieren validar se basan en el comportamiento que una persona cree que debería tener en base a ejemplos determinados.

Nosotros necesitábamos un dataset que contuviera frases cortas etiquetadas porque, de acuerdo con Sebastiani (2002), la detección de temáticas en grandes cantidades de documentos con mucho texto es un problema resuelto.

En primer lugar, buscamos diferentes datasets en la comunidad científica. Desgraciadamente no encontramos uno que se adecuase a nuestras necesidades. Encontramos trabajos relacionados con la taxonomía de DBpedia para catalogar automáticamente nuevos artículos (Gangemi et al., 2012) (Aprosio et al., 2013) (Giovanni et al., 2012). También diferentes datasets sobre DBpedia y el contenido de los artículos[1] y sobre diferentes esquemas y sobre el uso de SPARQL[2]. Ninguno de ellos satisfacía nuestra necesidad, por lo que la solución que encontramos fue crear nuestro propio dataset para realizar las pruebas.

Para la generación de este nuevo recurso, la primera decisión fue decidir de dónde íbamos a obtener las frases cortas. Se nos ocurrió la idea de obtener los diálogos de las películas. Teniendo en cuenta que debían de ser en inglés y que existen multitud de sitios web que ofrecen los subtítulos en diferentes idiomas, era una buena opción. La siguiente decisión fue elegir la película. Revisamos algunas películas pero no encontramos ninguna que nos gustara. Entonces se nos ocurrió utilizar series televisivas. Por aquella época la serie *Cómo conocí a vuestra madre*, o en inglés "*How I met your mother*" (HIMM), estaba en los últimos capítulos y pensamos en utilizar sus subtítulos para nuestro objetivo. El argumento de la serie era perfecto ya que no abordaba un tema concreto como el policiaco, el médico, el deportivo, el musical o el de aventuras. El argumento era bastante abierto y encajaba con conversaciones que las personas suelen tener.

El siguiente paso era obtener los diálogos de los subtítulos. Para ello utilizamos los que ofrece la Web "Subtitulos.es". Esta web es un referente para los internautas a la hora de conseguir los subtítulos en varios idiomas de sus series favoritas.

[1] http://blog.dbpedia.org/category/dataset-releases/
[2] http://datahub.io/dataset/dbpedia

Nosotros descargamos los subtítulos[3] de los cuatro primeros capítulos de la novena temporada. El formato de estos subtítulos se muestra en la figura 4.1.

```
1
00:00:05,672 ->00:00:08,107
Narrator:
Kids, barney and robin's
wedding turned out

2
00:00:08,109 ->00:00:10,476
To be a life-changing
weekend
for all of us.

3
00:00:10,478 ->00:00:12,277
Well, not just us.

4
00:00:12,279 ->00:00:14,013
One ticket to farhampton,
please.

{...}
```

Figura 4.1: Formato del archivo con los subtítulos.

Los elementos marcados en color rojo son el texto correspondiente a cada frase realizada por cada uno de los actores. El resto es información sobre la posición dentro de las traducciones que ocupa el elemento y, para que los procesadores de subtítulos sepan cuando mostrar el texto, el tiempo de comienzo y final de la frase. Nosotros hemos extraído únicamente el texto y hemos generado un dataset con estas frases. La idea era etiquetar cada una de las frases con los elementos de la taxonomía que creamos de DBpedia y de Amazon y que hemos mostrado en las figuras 3.13 y 3.15.

El resultado de la unión de los recursos obtenidos de los subtitulos junto con la etiquetación por parte de expertos fueron dos datasets. Estos contenían el estado del procesamiento de la frase, la propia frase en inglés y por último, los topics o categorías de nuestras taxonomías de DBpedia y Amazon. Podemos ver un ejemplo de ambos datasets en la figura 4.2.

Donde los elementos de color **rojo** muestra si la frase se ha procesado o no. Los de color **azul** muestran la frase en inglés y de color **verde** los topics que el experto ha seleccionado para esa frase.

Como también se puede ver en el ejemplo, existe una clase catalogada como 'Ninguno". El significado de esta clase se debe a que no siempre las frases tienen una clase (topic o categoría) relacionada con nuestra taxonomía. Debido a este escenario, se decidió insertar esta nueva clase para que todas las frases estuvieran etiquetadas.

[3]http://www.subtitulos.es/show/31

```
Frases etiquetadas con las
categorías de Amazon
True;
Kids, barney and robin's wedding
turned out;
crafts,_hobbies_&_home[*-*]weddings
{...}

Frases etiquetadas con los topics
de DBpedia
True;
Kids, barney and robin's wedding
turned out;
person;societalevent;givenname
{...}

Frases etiquetadas con la clase
Ninguno
True;
Well, not just us.;
Ninguno
{...}
```

Figura 4.2: Ejemplos de frases etiquetadas con los datasets de DBpedia y Amazon.

Este trabajo es la base para poder ofrecer productos de Amazon, o de cualquier otra empresa, en base a los contextos de las clases existentes en DBpedia. Nosotros hemos etiquetado ambos datasets por un motivo simple, verificar que los resultados de los clasificadores, con respecto al criterio de los expertos, no sean demasiado diferentes para poder elegir la mejor configuración posible, tanto para la elección del contexto como para la elección de la familia de productos. Si bien es cierto, el proceso final del sistema que se plantea es que los resultados obtenidos de la clasificación de DBpedia sea la entrada para clasificar los productos de Amazon, pero hemos visto necesario poder cuantificar cada enfoque y configuración para poder elegir, en ambos pasos, el mejor sistema posible. Estas decisiones tendrán que valorarse según la necesidad que se tenga. El prototipo que se muestra en este trabajo utiliza el mapa de traducciones, los resultados de DBpedia como entrada y, como resultado, las familias de productos de Amazon con el enfoque de deep learning descrito en la sección 4.3. Además, permite variar el nivel de tolerancia y los diferentes modelos para obtener resultados. El objetivo es ver el potencial de uno de los enfoques planteados.

4.1.1.1 Generación de los datasets y etiquetación

La generación de nuestros datasets para validar el trabajo con las taxonomías de DBpedia y Amazon, fue un proceso lento y costoso debido a la cantidad de datos a analizar y catalogar. Los pasos que se siguieron para generarlos fueron: i) utilizar la identificación y definiciones de los ámbitos de cada clase obtenidos en la creación de la base de conocimiento (véase secciones 3.2.6 y 3.2.7), ii) generar un sistema para

etiquetar cómodamente las frases de la serie HIMM y iii) enseñar a cada experto que áreas existían y la definición de cada una de ellas.

Como la parte de definición de clases estaba realizada, se utilizo el sistema de etiquetación desarrollado previamente y que se ha descrito en la sección 3.3.2. En este caso, el sistema mostraba, en la parte superior, la frase junto con la burda traducción automática. En la parte inferior izquierda, el árbol de todos los nodos obtenidos para crear nuestra taxonomía. Estos nodos eran seleccionables. En la parte inferior derecha existían dos elementos; el superior era un buscador para encontrar nodos de manera más ágil y el inferior eran los elementos que el experto había seleccionado y que iban a etiquetar a la frase actual. En la imagen 4.3 se ve la pantalla principal del sistema para este paso. Este programa se utilizo para etiquetar tanto DBpedia como Amazon por parte de los expertos.

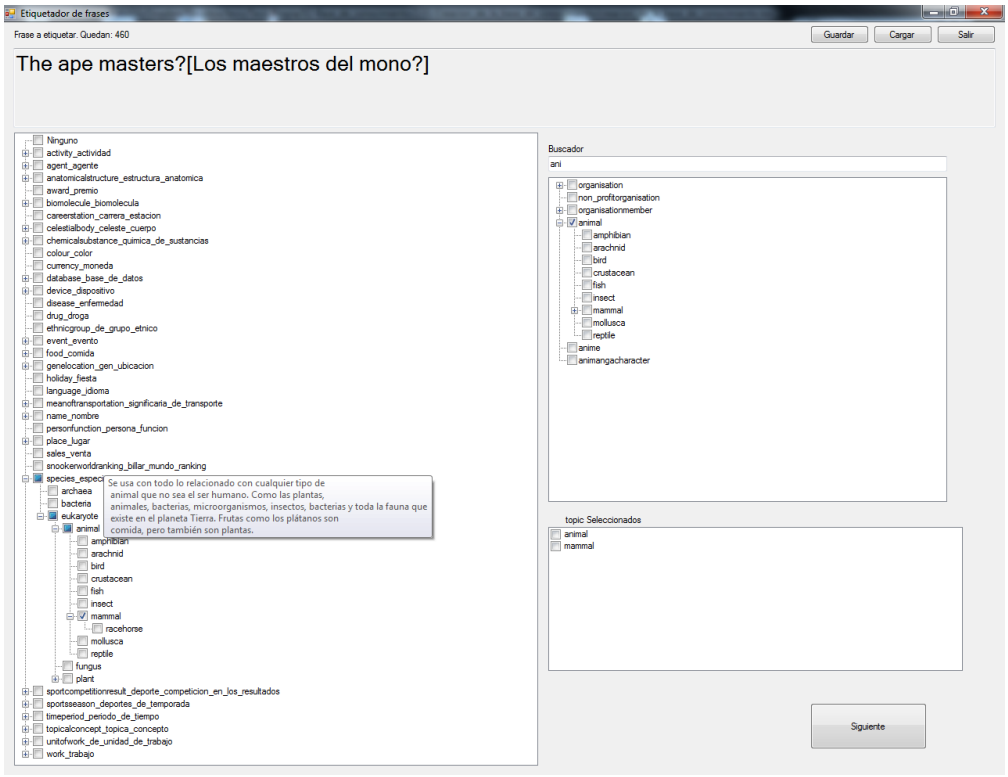


Figura 4.3: Pantalla principal del sistema para etiquetar frases con la taxonomía de DBpedia.

Con esta herramienta, cinco expertos etiquetaron los textos de los subtítulos (véase figura 4.1) de los cuatro primeros capítulos de la novena temporada de HIMM en base a las taxonomías generadas de DBpedia y Amazon mostradas en las figuras 3.13 y 3.15. El resultado fueron dos nuevos recursos, de los cuales, se muestran trozos

en la figura 4.2. Esta fue la manera manual de etiquetar las frases, para generar los datasets y utilizarlos en el experimento baseline y para poder verificar lo efectivo de nuestro sistema automático y sus ampliaciones.

4.1.2 Definición del experimento

Este experimento es la base de la que hemos partido para realizar el trabajo completo. Además, es con el primero que hemos utilizado los nuevos recursos generados como los índices y los datasets para validar el trabajo. Los datasets utilizados para la validación, que se han explicado en la sección 4.1.1, contienen 1919 frases cortas etiquetadas por 5 expertos. Las 1919 frases están filtradas para eliminar las los elementos repetidos y que son exactamente iguales. Por otra parte, las clases con las que etiquetamos las frases cortas de este dataset provienen de la generación de los índices de DBpedia y Amazon (véanse secciones 3.2.6 y 3.2.7). El número de clases con las que se pueden etiquetar las frases cortas son 4227 para Amazon y 392 para DBpedia.

Para realizar los experimentos, hemos utilizado únicamente las frases sin las etiquetas de los expertos. Además, hemos elevado las clases en el árbol hasta la clase principal, es decir, la clase padre que está en el índice 1 (véase sección 3.2.6, figura 3.14). La razón era obtener una versión inicial de los resultados y poder verificar si era viable continuar por este camino. Además, la utilización del resto de niveles era para especificar, más concretamente, a qué tipo de clase principal pertenecía la frase.

Con el recurso de las frases sin etiquetar y los índices generados, etiquetamos cada una de las frases de los capítulos de HIMM, utilizando el sistema automático desarrollado en el aparatado 3.2.6. Los resultados de DBpedia obtuvieron 4068 etiquetas y los de Amazon 5606. Hay que tener en cuenta que, debido a la naturaleza de las conversaciones, algunas de ellas carecen de contenido trascendente (i.e. conversaciones banales) y que existe gran cantidad de etiquetas con la clase "**Ninguno**". Concretamente aparece **789** para DBpedia y **817** para Amazon.

Utilizando estos 2 recursos, se han realizado 2 experimentos:

1. En el primer experimento, se intento validar la eficacia del método propuesto mediante la comparación de los resultados obtenidos por el sistema automático y los de los expertos.

2. En el segundo experimento, se sustituyeron iterativamente cada experto con los resultados del sistema automático (i.e. si tenemos 5 expertos, utilizamos los resultados de 4 expertos junto con los resultados del sistema automático para compararlo con los resultados de los 5 expertos) para verificar si es posible

sustituir a un experto con nuestro sistema automático. Este tipo de validación es el seleccionado para entregar el premio Loebner en Inteligencia Artificial[4]. Este premio esta basado en el paper de Alan Turing *Computing Machinery and Intelligence* Turing (1950).

4.1.3 Metodología para realizar el experimento baseline

Para realizar los experimentos, hemos utilizado 2 índices diferentes por cada fuente (DBpedia y Amazon) y un sistema basado en configuraciones. Además, se han definido 3 pasos básicos para obtener los resultados utilizando los recursos generados:

1. Preprocesar cada frase como se describe en la sección 4.1.3.2.
2. Eliminar los resultados que no superen un límite mínimo establecido. Este paso puede ser ejecutado por cada nivel de búsqueda o al finalizar todas las búsquedas, de acuerdo a la configuración seleccionada, descrita en la sección 4.1.3.3.
3. Generar los resultados ordenados y filtrados por el "*Lucene score*". Este valor será comparado contra el límite mínimo definido en la configuración.

4.1.3.1 Descripción de los índices utilizados

Los índices generados están divididos en 2 tipos según la fuente seleccionada para crearlo (Amazon y DBpedia). A su vez, cada fuente también ha sido dividida en 2 tipos. El primer tipo contiene únicamente el índice de primer nivel (nivel 1), es decir, la clase más generalista y todo el conocimiento del área al que pertenece. Este índice ha sido denominado como nivel 0 o **N0** y que en la figura 3.14 corresponde a la información almacenada en *Index L1*. El segundo tipo se le ha llamado índice **NX** y contiene el índice de N0 junto con el resto de índices, desde el de nivel 1 hasta el de nivel 4 (véase figura 3.14). NX contiene información más granulada de todo el conocimiento almacenado. La razón para realizar esta división era comprobar si era posible obtener buenos resultados utilizando únicamente información muy generalista que contiene toda la información en una única capa o, por el contrario, utilizando información más específica, dividida en niveles, y que puede recuperar información oculta a capas superiores.

4.1.3.2 Descripción del preprocesamiento de las frases

Para preprocesar cada frase, se han seguido 3 pasos. Primero se han cambiado todas las palabras de la frase a minúsculas para estandarizar toda la información, tal

[4]http://www.loebner.net/Prizef/loebner-prize.html

y como esta en los índices generados. Segundo, se han eliminado las palabras que no aportan significado a la frase, en inglés *stopwords*[5], que corresponden al idioma de la frase. En este caso el inglés. "*En informática, stopwords es el nombre que reciben las palabras sin significado como artículos, pronombres, preposiciones, etc. que son filtradas antes o después del procesamiento de datos en lenguaje natural (texto). No existe una única lista de estas palabras utilizadas por los sistemas de NLP. Además, no todas las herramientas de NLP necesitan que estas palabras sean eliminadas para poder realizar búsquedas de frases completas. La categorización de las palabras como stopwords viene dada en base al contexto en el que se quieren utilizar y el propósito.*"[6]. Las stopwords han sido utilizadas en trabajos como el de Rajaraman y Ullman (2011) y Wilbur y Sirotkin (1992) para utilizarlas en las consultas a motores de búsqueda y, de esta manera, optimizar el rendimiento. Este paso ha sido aplicado únicamente a los datos que han sido almacenados en el campo **INFO** de los índices (véase sección 3.2.5.3). Finalmente, también se han eliminado de los datos los caracteres especiales que afectan al motor de búsqueda Lucene. Esta eliminación es importante debido a que, estos caracteres pueden interferir dentro del proceso de búsquedas del motor. Los caracteres eliminados son +, -, &&, (), !, {}, [], ∧ , ",∼, *, ?, :, \\, /.

4.1.3.3 **Descripción de la configuración utilizada**

Para poder evaluar todas las posibilidades que se contemplaban para estos experimentos, se utilizó un sistema de configuraciones. Además, con este sistema se intento encontrar las variables óptimas para el sistema de recuperación de la información. Cada configuración está compuesta por 5 campos que se centran en diferentes aspectos de las búsquedas.

1. El tipo de índice ha sobre el que hacer las búsquedas. Más información en la sección 4.1.3.1.

2. El limite mínimo para filtrar resultados.

3. El tipo de filtrado.

   Completo, definido en inglés como **whole filtering type** (WFT). Este tipo está activado con el valor **false** en las configuraciones y su función es filtrar los resultados al finalizar las búsquedas en todos los niveles. Es decir, realizar las búsquedas de clases en cada nivel y filtrar estas búsquedas una vez se hayan recuperado todos los resultados de todos los índices. El valor del filtro es el límite mínimo, definido previamente, y el valor contra el que se compara es el

[5] http://paginaspersonales.deusto.es/patxigg/recursos/EnglishStopWords.txt
[6]https://en.wikipedia.org/wiki/Stop_words

"*Lucene score*". Este valor se obtiene del motor de búsqueda Lucene y expresa la similitud entre la consulta realizada y la información almacenada).

Por nivel, definido en inglés como **step filtering type** (SFT). Este tipo está activado con el valor **true** en las configuraciones y su función es filtrar resultados de las búsquedas por cada búsqueda en cada nivel. Es decir, realizar una búsqueda en el nivel correspondiente y filtrar en ese mismo momento los resultados con el limite mínimo y el "*Lucene score*". Solo se seguirá buscando en los resultados que superen esta criba. De esta manera, la búsqueda se centra mucho más en áreas específicas.

4. Eliminar las *stopwords*". En la configuración se utiliza **Yes** para eliminarlas y **No** para dejarlas. Esta variable se aplica a cada frase antes de cada petición que se le hace al motor de búsqueda.

5. El número máximo de posibles resultados por cada búsqueda, en cada nivel, y de manera global. Es decir, al final solo se obtienen el número de resultados que marca esta variable.

Los valores de algunas variables de la configuración como el máximo número de posibles resultados o el límite mínimo, han sido seleccionadas teniendo en cuenta la experiencia de trabajos previos en el mismo área como, por ejemplo el de Laorden et al. (2014). Para este trabajo decidimos utilizar 3 valores como límites mínimos, 0,2 para el más restrictivo, 0,02 para el normal y 0,002 para el más permisivo. Para el número máximo de resultados, decidimos que el valor fuera 10 debido a que, el número máximo de soluciones dadas por los expertos eran 8 y las posibilidades máximas de respuesta de los recursos eran, 31 para Amazon y 33 para DBpedia. Esto se debe a que el sistema estaba limitado, independientemente del tipo y fuente de índice a utilizar, a devolver clases que estuvieran en el *Index L1* o N0. En este caso, el resto de niveles solo servía para otorgar más votos e importancia a una clase principal. Finalmente, el número de configuraciones que utilizamos fue de 24. Con ellas abarcábamos todas las posibilidades en las que habíamos pensado para el experimento. Estas configuraciones están enumeradas en la figura 4.4.

4.1.4 Métricas utilizadas

Para la evaluación de los resultados, se han creado 3 nuevas métricas para comprobar el correcto funcionamiento de los algoritmos diseñados y el etiquetado del sistema automático. Estas nuevas métricas las hemos llamado Precisión de Ventana Local, en inglés **Local Window Precision** (LWP), Exhaustividad de Ventana Local, en inglés **Local Window Recall** (LWR) y Media Geométrica de Coincidencia Local,

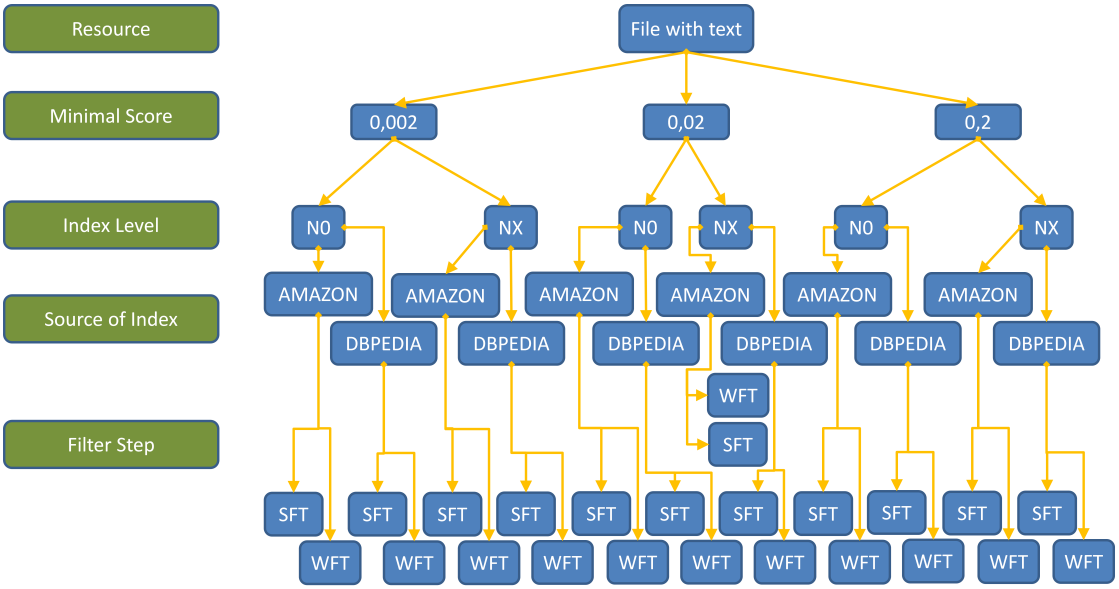


Figura 4.4: Configuraciones utilizadas para realizar las búsquedas del experimento baseline en el indice.

en inglés **Local Coincidence Geometric Mean** (LCGM). Las variables utilizadas en las formulas son:

- C: El número de coincidencias entre las clases que han dado los expertos y las propuestas por el sistema automático.
- E: El número de clases únicas dadas por los expertos.
- M: El número de clases (siempre únicas) dadas por el sistema automático.
- R_E: El número máximo de posible clases que pueden dar los expertos (máximo número de resultados), el cual, esta restringido según la fuente elegida (DBpedia o Amazon).
- R_M: El número máximo de posible clases que puede dar el sistema automático, el cual, está restringido por configuración (véase sección 4.1.3.3).

Además de estas métricas, también se utilizaron otras más comunes como distancia entre vectores, en inglés **distance between vectors**, la media de la desviación de la raíz cuadrada, en inglés **root mean squared error** (RMSE), y la media del error absoluto, en inglés **mean absolute error** (MAE).

Las medidas utilizadas sirven para calificar cada uno de los enfoques que hemos desarrollado. Debido a que no existen medidas estándar para evaluar este sistema,

nos hemos basado en las medidas existentes, pero adaptándolas al funcionamiento de nuestro sistema. Debido a esto, hemos pensado que mostrar los resultados de las 3 medidas, junto con las medidas de los sistemas de recomendación y el de los vectores ofrece a la persona que lee este trabajo una visión más extensa de como evaluar cada configuración de cada enfoque para, en caso de utilizar dicho sistema, tener en cuenta que medida necesita para evaluar su sistema en base al objetivo deseado.

4.1.4.1 **LWP**

La métrica LWP se centra en la evaluación de la "precision" de los resultados con el objetivo de proveer la probabilidad de relevancia de las clases obtenidas incluyendo una pequeña penalización. Esta ligeramente inspirada en la medida de "recall" por contemplar los errores cometidos (hay que tener en cuenta que si el resultado del sistema automático es 0 coincidencias, el valor de la función debería de ser 0, a pesar de las penalizaciones):

$$LWP = f_a - (f_b \cdot f_c) \tag{4.1}$$

donde f_a corresponde al valor máximo que puede alcanzar la función en base a las coincidencias encontradas entre los expertos y el sistema automático:

$$f_a = \frac{C}{M} \tag{4.2}$$

donde f_b corresponde a la unidad de penalización por cada clase no encontrada por el sistema automático. Este valor es calculado teniendo en cuenta que la máxima penalización nunca será igual a obtener una coincidencia menos (e.g., 5 correctas y 1 error nunca será mejor que solo 4 correctas):

$$f_b = \frac{\frac{C}{M} - \frac{C-1}{M}}{R_E - (C+1)} \tag{4.3}$$

donde f_c es el número de clases que el sistema automático no ha detectado:

$$f_c = E - C \tag{4.4}$$

Finalmente, si unimos las formulas anteriores y las simplificamos, obtenemos:

$$LWP = \frac{\left(C - \frac{E-C}{R_E-(C+1)}\right)}{M} \tag{4.5}$$

4.1.4.2 LWR

La métrica LWR se centra en la evaluación del "recall" de los resultados con el objetivo de calcular la proporción de documentos relevantes recuperados, comparados con el total de documentos que son relevantes según la etiquetación de los expertos. Es decir, la cobertura del sistema automático. Además, se añade una pequeña penalización. Esta medida está ligeramente inspirada en la medida de "precision" dada por sus errores (hay que tener en cuenta que si el sistema automático da como resultado 0 coincidencias, el valor de la función deberá ser 0, a pesar de las penalizaciones).

$$LWR = f_a - (f_b \cdot f_c) \tag{4.6}$$

donde f_a corresponde al máximo valor que puede ser alcanzado por la función, basándose en las coincidencias entre el sistema automático y los expertos.

$$f_a = \frac{C}{E} \tag{4.7}$$

donde f_b corresponde a la unidad de penalización por cada clase mal detectada. Este valor es calculado teniendo en cuenta que la máxima penalización nunca será igual a obtener una coincidencia menos (e.g., 5 correctas y 1 error nunca será mejor que solo 4 correctas):

$$f_b = \frac{\frac{C}{E} - \frac{C-1}{E}}{R_M - (C+1)} \tag{4.8}$$

donde f_c es el número de intentos extra incorrectos obtenidos por el sistema automático.

$$f_c = M - C \tag{4.9}$$

Finalmente, la fórmula utilizada es:

$$LWR = \frac{\left(C - \frac{M-C}{R_M-(C+1)}\right)}{E} \tag{4.10}$$

Consideraciones a tener en cuenta

Hay una importante consideración a tener en cuenta con respecto a LWP y LWR debido a la naturaleza de la validación propuesta. Debido a que el sistema automático actúa como lo haría un experto humano, tiene la posibilidad de proporcionar solamente dos soluciones: i) 1 o varias clases o ii) la clase "**Ninguno**". No es posible encontrar la clase "**Ninguno**" y otras clases en la misma solución del sistema

automático. Esto significaría que, si nuestro sistema no encontrara ninguna clase para una frase concreta y cualquiera de los expertos estaría de acuerdo, el resultado no debería de ser completamente bueno porque solo coincide con un experto. Sin embargo, como el objetivo de este enfoque es demostrar que el sistema automático puede sustituir a un experto, se deben tomar las siguiente consideraciones en cuenta:

1. Si en los resultados de los expertos no aparece la clase "**Ninguno**", las métricas LWP y LWR son usadas como previamente han sido definidas.

2. Si en los resultados de los expertos aparece la clase "*Ninguno*", se añaden ciertas consideraciones al cálculo de LWP y LWR.

   Si el sistema automático obtiene la clase "*Ninguno*" y cualquiera de los expertos (1 o más, pero no todos) coincide, LWP y LWR tendrán el valor de 86 %. Es un valor alto porque el sistema puede sustituir a, al menos, un experto, pero no es perfecto porque no coincide con el resto de los expertos. Además, la ausencia de la clase "*Ninguno*" no penaliza tanto el resultado como la ausencia de las otras clases. Por el contrario, si algún experto dice la clase "*Ninguno*", pero el sistema automático obtiene cualquier clase que no sea "*Ninguno*", el valor del experto se elimina de los resultados de los expertos y el cálculo de LWP y LWR se harán como se han descrito anteriormente con la salvedad de que, su máxima "precision" será del 95 %. Este valor de penalización se ha elegido mediante estudio empírico y teniendo en cuenta que, no encontrar la clase "*Ninguno*" cuando hay otras clases no es tan malo como no encontrar una clase especifica que ha sido etiquetada por un experto.

3. Si todos los resultados de todos los expertos es la clase "*Ninguno*", se aplican las siguientes consideraciones a LWP y LWR. Si el sistema automático también da como resultado la clase "*Ninguno*", el resultado final es de completo acuerdo, es decir 100 %. Sin embargo, si el sistema automático da cualquier clase cuando los expertos han sido unánimes diciendo la clase "*Ninguno*", el resultado final es un completo error, es decir 0 %.

4.1.4.3 **LCGM**

La métrica LCGM es la media geométrica de la probabilidad de que una clase encontrada sea correcta y la probabilidad de que una clase de la frase sea encontrada.

$$LCGM = \frac{C}{\sqrt{E \cdot M}} \tag{4.11}$$

4.1.4.4 **Distancia de vectores**

Además, para el cálculo de la similitud entre los resultados de los expertos y los del sistema automático, hemos representado cada frase como vectores utilizando VSM. Los puntos de los vectores están basados en los pesos de las clases y el cálculo de la similitud se calcula en base a la distancia entre vectores. Este tipo de métrica para calcular y cuantificar la similitud entre textos está muy utilizado en trabajos científicos. Por ejemplo para identificar personas como en los trabajos de Dehak et al. (2011), Ali et al. (2014), Liu y Guan (2014) y Galán-García et al. (2014), para clasificar sentimientos como en el trabajo de Zhang et al. (2015a) o para identificar y clasificar conceptos del comercio electrónico o "e-commerce" como en el trabajo Cao et al. (2014).

Para el cálculo de los pesos de cada clase y su posterior representación, hemos seguid los siguientes pasos:

1. Obtener el número de clases por cada frase que dan como resultado cada experto/sistema automático y almacenar el número total de clases por cada frase.
2. Normalizar (entre 0 y 1) las clase por cada frase en base al número total de clases encontradas para cada frase, de modo que la suma de los pesos sea igual a 1.
3. Eliminar la clase "*Ninguno*" y su correspondiente peso de toda la frase (una vez que la frase ya ha sido normalizada en el paso anterior, se puede detectar que en esa frase existía la clase "*Ninguno*" debido a que la suma total de todos los pesos no es 1. De esta manera, su representación es más sencilla).
4. Por último, generamos un único vector para cada frase. En el caso de los expertos, lo llamamos el "experto común". Este vector se crea utilizando la media geométrica de las clases obtenidas por todos los expertos para cada frase.

Una vez que hemos generado los vectores para cada frase etiquetada por expertos, podemos calcular la similitud con los vectores de clases que obtenemos del sistema automático. Por ejemplo, para calcular la distancia entre los vectores $\vec{Av}$, $\vec{Av} = x_a, y_a$ y $\vec{Bv}$, $\vec{Bv} = x_b, y_b$, utilizaríamos la ecuación:

$$distancia = \sqrt{((x_a - x_b)^2 + (y_a - y_b)^2} \tag{4.12}$$

4.1.4.5 **RMSE y MAE**

Finalmente, hemos incluido 2 métricas que son comúnmente utilizadas en los sistemas de recomendación (Herlocker et al., 2004), en inglés "*Recommended Systems*" (RS) , RMSE y MAE.

$$RMSE = \sqrt{\frac{1}{|\gamma|}\sum_{(u,i)\in\gamma}(\hat{r}_{ui} - r_{ui})^2} \tag{4.13}$$

$$MAE = \sqrt{\frac{1}{|\gamma|}\sum_{(u,i)\in\gamma}|\hat{r}_{ui} - r_{ui}|} \tag{4.14}$$

En ambas, el sistema genera las predicciones $\hat{r}_{ui}$ para un conjunto de prueba γ de pares *usuario-respuesta* (u,i) para los cuales, el resultado r_{ui} es conocido. Normalmente r_{ui} es conocido para experimentos offline o porque ha sido obtenido mediante el estudio o la experimentación online. Una diferencia remarcable entre RMSE y MAE es que RMSE penaliza mucho más los grandes errores.

4.2 Buscadores Web

4.2.1 Introducción

Las frases cortas, normalmente no contienen una información clara acerca de su contexto y aunque uno de nuestros recursos, DBpedia, este pensado para obtener el contexto, queríamos comprobar si era viable ampliar un poco más el contexto de la frase mediante buscadores de contenido en Internet, es decir, buscadores Web y de esta manera mejorar los resultados.

Internet, hoy en día, es una fuente de datos grandísima. En ella se crean al día millones de nuevos elementos con nuevos contenidos. Estos contenidos pueden ser nuevos puntos de vista de otros contenidos que ya existen, más información sobre elementos ya publicados o simplemente ideas genuinas. Debido al gran número de habitantes que poblamos la Tierra y que utilizamos esta red mundial, las opiniones, los recursos y en general, todo el contenido que creamos, genera cantidades ingentes de nueva información. Buscar entre toda esta información puede llegar a ser un caos de proporciones épicas. Para ayudarnos en esta ardua labor existen los motores de búsqueda.

Estos sistemas ayudan a localizar archivos o recursos alojados en servidores Web mediante una técnica llamada "*Web crawler*"[7]. Dicha técnica utiliza sistemas informáticos llamados *robots* para revisar todos los recursos que existen en los servidores de manera metódica y automatizada. El resultado es la obtención de los datos más significativos de los recursos y, además, una copia de todas las páginas que visita para después ser indexadas por el motor de búsqueda. De esta manera se genera

[7] www.robotstxt.org/orig.html

un sistema de búsqueda rápido con todos los elementos de los servidores. El modo de funcionamiento de los *robots* es simple. Estos visitan las direcciones Web de los diferentes recursos que almacenan los servidores y identifican los elementos como imágenes, textos y enlaces a otros recursos Web. Estos últimos son añadidos de forma recursiva a una lista, previa criba con reglas preestablecidas, para seguir visitando e indexando una vez terminado el análisis al recurso actual. La manera de iniciar estos sistemas es mediante una lista de direcciones principales o iniciales, seguidamente el *robot* se conecta a estos recursos y comienza su descarga, analiza la página Web y busca dentro de la misma nuevos recursos para volver a empezar de nuevo este proceso con el nuevo recurso.

Este funcionamiento ha ido evolucionando a lo largo de los años. El primer motor de búsqueda de la historia se creó en 1990 y su nombre era *Archie* (Vaquero Pulido, 1997). Este sistema era una evolución del comando "*grep*" de UNIX para realizar búsquedas en servidores FTP. Su funcionamiento inicial era simple, se creaba previamente una lista con todos los archivos que contenía el servidor de contenidos FTP y *Archie* buscaba un nombre concreto dentro de esa lista pregenerada. El listado con los nombres de los documentos se almacenaba de manera local dentro del cliente y se actualizaba periódicamente para evitar saturar el servidor remoto. Posteriormente, se creó el "*front-end*" y "*back-end*" para mejorar la experiencia y usabilidad. El protocolo sobre el que se basaba este tipo de búsquedas era Gopher (Anklesaria et al., 1993).

En 1993, nace uno de los primeros motores de búsqueda o buscador Web, su nombre es "*Wanderer*" (Knoblock, 1997) por parte de Matthew Gray[8], de Massachusetts Institute of Technology. Era un robot escrito en el lenguaje Perl que analizaba los sitios Web generando un indice llamado "*Wandex*". Al año siguiente, en 1994, vio la luz el primer buscador de texto completo, su nombre era "*WebCrawler*[9]". Este nuevo motor de búsqueda era el primer metabuscador.

Un metabuscador es un buscador que carece de una base de datos propia, ya que localiza y utiliza la información que los motores de búsqueda más usados y populares le dan, mostrando una combinación de los mejores resultados de todos los buscadores utilizados. La descripción más explícita de un metabuscador es que és un "buscador de buscadores". La gran diferencia con los anteriores sistemas era que éste, permitía la búsqueda por palabras que estaban contenidas dentro de las páginas o recursos Web de los servidores llegando a convertirse esta metodología en un estándar para la mayoría de los buscadores.

Al mismo tiempo apareció *Lycos*[10]. Este sitio, es un portal Web que además, in-

[8] matthew.gray.org
[9] www.webcrawler.com
[10] www.lycos.es

cluía un buscador Web. Fue desarrollado por el Dr. Mauldin (1997), de la Universidad de Carnegie como un desarrollo de un motor de búsqueda llegando a convertirse en uno de los más importantes en el año 1995. Seguido de estos buscadores, aparecieron otros nuevos como *Excite*[11], *Infoseek*[12], *Inktomi*[13], *Northern Light*[14] y *Altavista*[15], los cuales competían con directorios de recursos e índices temáticos de Yahoo!. Estos índices, acabarón por añadirse a la tecnología de los buscadores para aumentar su funcionalidad.

En el año 2015, teniendo en cuenta que los buscadores se han convertido en una plataforma indispensable para navegar por Internet y que los smartphones se han convertido en la herramienta más utilizada para navegar, los datos de los buscadores han ido evolucionando hasta convertirse en los sistemas de entrada a cualquier recurso en Internet. Los usuarios ya no conciben Internet sin los buscadores que les ayudan a encontrar lo que están buscando. Para hacernos una idea de su utilización, en las figuras 4.5 y 4.6 se muestran los datos de uso de los motores más utilizados en ordenadores. Por otra parte, en las figuras 4.7 y 4.8 se muestran los buscadores más utilizados por los usuarios en dispositivos móviles. Estos datos están obtenido de las 2 fuentes más importantes de estadísticas de Internet, *StatCounter*[16] y *NetMarketShare*[17]. La diferencia entre la generación de estadísticas entre estos sitios son la muestra de sitios web analizados (40.000 para *NetMarketShare* y 3.000.000 para *StatCounter*) y la medición que utilizan. *NetMarketShare* mide usuarios únicos y *StatCounter* páginas vistas. Para explicar la diferencia pongamos el siguiente ejemplo: el usuario A utiliza Chrome y visita la portada de Facebook y la de Twitter, luego cierra el navegador. El usuario B usa Firefox, entra en Facebook, visita un montón de páginas y hace lo mismo en Twitter. En este caso, *NetMarketShare* mediría el mismo uso para ambos navegadores, mientras que *StatCounter* mediría un mayor uso de Firefox debido a que el usuario ha visto más páginas con este navegador. En este sentido es más fiable el método de *StatCounter*, ya que refleja el verdadero uso del navegador. Sin embargo, como es más sencillo falsear páginas vistas que usuarios únicos, el de *NetMarketShare* es menos propenso al fraude.

Viendo los datos recogidos, los buscadores más utilizados en Internet son Google, Bing y Baidu. De estos buscadores, los únicos que trabajan a nivel mundial con contenido de todos los países son Google y Bing. Estos motores de búsqueda son los que hemos escogido para realizar nuestro experimento.

[11] www.excite.com
[12] www.infoseek.co.jp
[13] Inktomi pertenece, desde finales de los 90, a Yahoo! y actualmente ha desaparecido
[14] w ww.nlsearch.com
[15] www.altavista.com
[16] https://statcounter.com
[17] https://www.netmarketshare.com

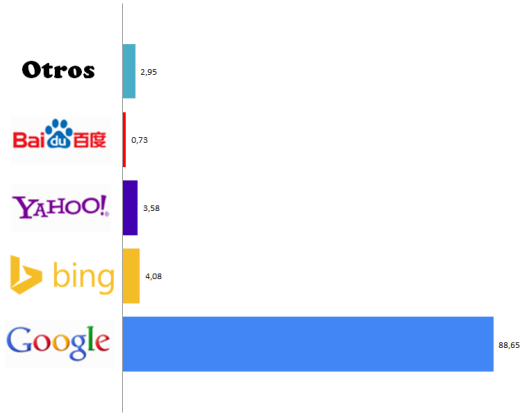


Figura 4.5: Estadística sobre el uso de buscadores Web en PC de *StatCounter*.

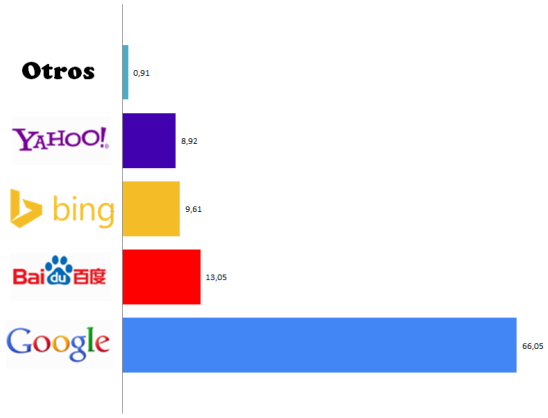


Figura 4.6: Estadística sobre el uso de buscadores Web en PC de *NetMarketShare*.

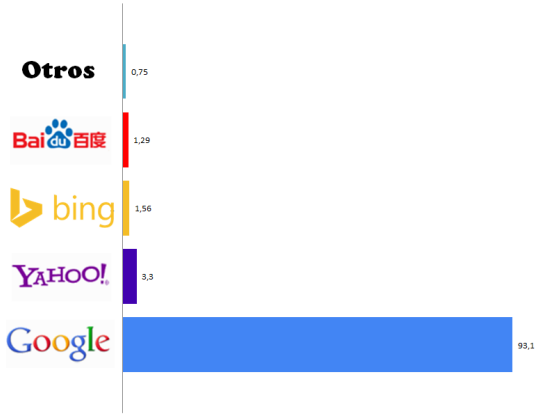


Figura 4.7: Estadística sobre el uso de buscadores Web en dispositivos móviles de *StatCounter*.

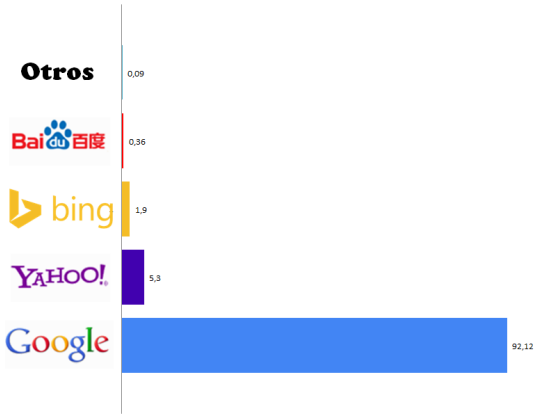


Figura 4.8: Estadística sobre el uso de buscadores Web en dispositivos móviles de *NetMarketShare*.

4.2.2 Ampliando el contexto

La ampliación del contexto es una técnica muy utilizada para poder descubrir el sentido de una palabra o conjunto de palabras. Una de las áreas en las que más se usa la ampliación del contexto es en la generación de cadenas de búsqueda o *querys* para reducir el número de documentos que un sistema de recuperación de información devuelve y no son relevantes. Estas cadenas de búsqueda contienen elementos que un usuario quiere encontrar dentro de una colección de recursos. El problema surge cuando los términos que utiliza el usuario para realizar la búsqueda no son exactamente iguales que los que están almacenados. En estos casos es necesario encontrar una relación entre los términos de búsqueda y los almacenados. En el trabajo de Xu y Croft (1996) se analizó la eficacia de los diferentes enfoques automáticos para la ampliación del contexto de las cadenas de búsqueda. También existen trabajos, como el de Pal et al. (2015) en el que exploran la categorización de las *querys* para determinar que estrategias de expansión son mejores. Pero uno de los problemas que tiene es el de mantener el sentido semántico de la cadena de búsqueda inicial. Trabajos como el de Jindal et al. (2015) se centran en esta temática para intentar mejorar la generación de nuevas cadenas, en base a su contexto y manteniendo un significado semántico semejante.

También existen estudios que utilizan este tipo de técnicas junto con conocidas bases de conocimientos, como por ejemplo YAGO, la cual describió Suchanek et al. (2007) y su segunda versión, YAGO2, descrita por Hoffart et al. (2011) y que están construidas con elementos de Wikipedia, WordNet y Nombres geolocalizados. Otros trabajos utilizan directamente sistemas de desambiguación contra Wikipedia, en inglés *Disambiguation to Wikipedia* (D2W). Estos sistemas comprenden la tarea

de unir conceptos dentro de un texto con las correspondientes entradas de la Wikipedia, tal y como mostraron Mihalcea y Csomai (2007) en su trabajo. Esta técnica, la utilizaron Cassidy et al. (2012) para ampliar el contexto de *tweets* y poder unirlos con sentidos más amplios y poder entender de que trataban. Otros enfoques, como el de Camacho-Collados et al. (2015), utilizan la ampliación del contexto mediante la extracción de información de diferentes fuentes como BabelNet[18] o Wikipedia, para realizar representaciones semánticas bastante flexibles e independientes del idioma y obtener representaciones vectoriales para después, poder comparar diferentes conceptos. Chen et al. (2014) enfoca el problema de representar los sentidos de las palabras mediante la obtencion de vectores individuales de sentidos utilizando, a mode de expansion de contexto, los *glosses* o resumenes de WordNet. El objetivo de utilizar este tipo de bases de datos o recursos es tener marcados ciertos contextos. Por ejemplo en el trabajo de Habib y Keulen (2012), el objetivo que persiguen es obtener más información sobre palabras de entidades nombradas para poder saber sobre que se está hablando. La expansión también se utiliza para traducir y reconocer las abreviaciones, tal y como hace Ammar et al. (2011). Otro punto importante en el que se utilizan estás técnicas es para la desambiguación de términos. Trabajos como el de Tacoa et al. (2012) y Rojas Lopez et al. (2015) lo demuestran. Además, en trabajos como el de Elsayed et al. (2008) juegan un papel importante en la detección de entidades dentro de emails. Estos son solo algunos ejemplos de las áreas en las cuales se utiliza la expansión de contexto como técnica importante.

En frases tan cortas como las que se abordan en este trabajo, la detección de la temática a la que se refieren es bastante difícil. Esto se debe a que, en el lenguaje hablado muchas veces existen uniones con contextos anteriores que ambos participantes conocen pero que no se ha introducido en la comunicación o, que por motivos de expresión corporal, se reconocen sin necesidad de nombrarlos, o simplemente es una divagación de ambos participantes. Sea cual sea el motivo, la falta de contexto es un fenómeno que se da mucho en este tipo de comunicación.

La metodología que se ha utilizado para aumentar el contexto y la información de las frases cortas consta de un proceso de 3 pasos, los cuales están determinados por la configuración de entrada seleccionada. Por otra parte, en base a las estadísticas mostradas en la sección 4.2.1 y a la facilidad de uso de los diferentes motores de búsqueda, para este experimento se han utilizado 2 motores de búsqueda web, Google[19] y Bing[20]. La solución que se ha planteado y aplicado a este problema ha sido utilizar los buscadores web para agregar contexto y más información a la frase principal, en base a los resultados obtenidos de los buscadores.

[18] http://www.babelnet.org/
[19] www.google.com
[20] www.bing.com

4.2.2.1 **Google**

El buscador de contenidos Google[21] fue creado por Larry Page y Sergey Brin como resultado de la tesis doctoral de ambos. La idea era mejorar las búsquedas en Internet, utilizando una familia de algoritmos llamado *Page Rank* (Page et al., 1999). Estos algoritmos asignan de forma numérica la relevancia a documentos o recursos indexados por el motor de búsqueda, en base a los enlaces de cada recurso con otros recursos que le apunten .

Una de las razones por las cuales ha elegido este buscador para este experimento es que actualmente, tal y como se ha mostrado en las figuras 4.8, 4.6, 4.7 y 4.5, es el buscador más popular y utilizado. Otra razón ha sido el gran soporte que ofrece Google para realizar desarrollos e investigaciones con su motor. La ayuda de la comunidad de Google Code[22] y de Google Developers[23] también ha sido un factor a tener en cuenta, ya que muchos de los problemas que se han presentado en la investigación han podido ser, en ocasiones, resueltos y en otras orientados hacia una solución.

4.2.2.2 **Bing**

El buscador de contenidos Bing[24] es la evolución o adaptación a los tiempos actuales del buscador web de Microsoft. Este buscador web ha tenido diversos nombres como *Live Search*, *Windows Live Search* o *MSN Search*. El cambio de cada nombre venia dado por cambios en la tecnología o enfoque entre otras circunstancias. Los cambios notables que han surgido en este buscador pasan por las sugerencias en tiempo real, las búsquedas relacionadas y el uso de tecnologías semánticas de empresas como *Powerset*[25]. La tecnología de esta compañía está basada en construir buscadores web que puedan encontrar respuestas adecuadas a las peticiones o preguntas de los usuarios. Un ejemplo de esta tecnología se da cuando un usuario realiza una pregunta en el buscador web. A modo de ejemplo, pongamos que la pregunta es *¿Cuál es la capital del mundo?*. La mayoría de los buscadores se centrarían en obtener los términos claves como **capital** y **mundo**, ofreciendo los resultados que coincidan con esos términos. La tecnología de *Powerset* utiliza técnicas de PLN para entender la pregunta y ofrecer resultados que contengan la respuesta.

Por otra parte, otra razón para elegir Bing en este experimento se debe a que, desde 2009, el motor de búsqueda de Yahoo![26], entre otros, ha adoptado la tecno-

[21] www.google.com
[22] https://code.google.com/
[23] https://developers.google.com/
[24] www.bing.com
[25] http://www.powerset.com/
[26] www.yahoo.com

logía de Bing en sus buscadores y, tal y como se muestra en las figuras 4.8, 4.6, 4.7 y 4.5, entre Bing y Yahoo! está el segundo puesto en el ranking de buscadores más utilizados. Otra razón a tener en cuenta sobre Bing es que también dispone de MSDN[27], su comunidad para ayudar en el soporte y de Dev Center[28] para la ayuda en el desarrollo.

4.2.3 Definición del experimento

Este experimento lo hemos realizado para analizar una posible rama que ayude en la detección de contextos y productos mediante la ampliación del propio contexto de las frases. La idea era verificar sí, utilizando los resultados de los motores de búsqueda web para ampliar el contexto de las frases se podía sustituir a un experto por dicho sistema. La ampliación de contexto con buscadores web nos ha servido para ampliar la información que teníamos de las frases, con contenidos actualizados que están en recursos web.

Internet, al ser un sistema vivo que genera cada día miles y miles de nuevos datos, siempre se mantiene actualizado. Aprovechándonos de esta cualidad, habíamos pensado que las modas lingüísticas, las abreviaturas, los coloquialismos, los lenguajes de tribus urbanas y, en definitiva, toda la nueva jerga que se va generando día a día, quedaría reflejada en la ampliación de contexto de las frases. Esto permitiría al sistema mantener una relación entre las nuevas modas lingüísticas y el lenguaje formal. Los recursos que hemos utilizado para realizar este experimento, como la base de conocimiento de contextos y productos y las frases etiquetadas, son las mismas que para el experimento baseline descrito en la sección 4.1.

Con la definición realizada y los recursos seleccionados, se han planteado 2 experimentos. Los resultados obtenidos se han comparado con los del baseline para comprobar los resultados:

1. En el primer experimento, se intentó validar la eficacia del método propuesto mediante la comparación de los resultados del baseline con los resultados obtenidos por el sistema automático. Los resultados comparados abarcan 3 tipos de resultados. 2 de ellos son los obtenidos mediante las búsquedas con los motores web con preprocesamiento y sin él. El tercero comprende los resultados sin utilizar buscadores web y con preprocesamiento. La decisión de realizar también las pruebas sin los resultados de los buscadores era para determinar si el aporte de los buscadores era significativo. Además, otro motivo es que las frases no contienen casi información y, además, vimos importante incluir

[27] https://msdn.microsoft.com/
[28] https://www.bing.com/dev/

la misma información con la que partían las configuraciones de los buscadores con preprocesamiento.

2. En el segundo experimento, se sustituyeron iterativamente cada experto con los resultados del sistema automático (i.e. si tenemos 5 expertos, utilizamos los resultados de 4 expertos junto con los resultados del sistema automático para compararlo con los resultados de los 5 expertos) para verificar si es posible sustituir a un experto con nuestro sistema automático. Este tipo de validación es el seleccionado para entregar el premio Loebner en Inteligencia Artificial[29]. Este premio esta basado en el paper de Alan Turing *Computing Machinery and Intelligence* Turing (1950).

4.2.4 Metodología para aumentar el contexto y la información

Para realizar los experimentos, hemos utilizado los mismos índices utilizados en el experimento baseline (véase sección 4.1) y un sistema basado en configuraciones. Además, el proceso está dividido y definido en 3 pasos básicos, según la configuración que se utilice, para obtener los resultados utilizando los recursos generados:

- Preprocesar cada frase tal y como se describe en la sección 4.2.4.1.
- Realizar búsquedas con la frase en el motor web seleccionado, obteniendo un número finito de resultados y de ellos, los resúmenes de cada resultado que contienen información acerca de la frase sin contexto.
- Postprocesar la frase y realizar búsquedas en el indice de Amazon y DBpedia.

4.2.4.1 Descripción del preprocesamiento utilizado

El preprocesamiento de las frases en este caso, es un paso que traduce del lenguaje vulgar al formal, detecta los términos importantes y les añade más información.

Para el paso de la traducción de las frases, se utilizó el servicio web de *lingo2word*[30]. Este servicio ha sido utilizado para traducir del lenguaje vulgar al formal los mensajes cortos o SMS en diferentes trabajos. Alguno de ellos como el de Hidalgo y Díaz (2012) para la detección de depredadores sexuales en salas de chat o el de Raghunathan y Krawczyk (2009) para la normalización de los mensajes SMS utilizando inteligencia artificial para su traducción. La traducción generada se almacena junto a la frase original. Una característica importante de este servicio es que,

[29] http://www.loebner.net/Prizef/loebner-prize.html
[30] http://www.lingo2word.com/

si la frase no está en lenguaje coloquial, el resultado que genera es el mismo que la entrada. Además, en ocasiones mejora a las propias frases originales mediante la utilización de sinónimos que disminuyen la ambigüedad (e.g., *car* se traduce a *automobile*).

Para la detección de los términos importantes hemos desarrollado una plataforma, con un cliente disponible en github[31], que tiene como base el sistema Freeling desarrollado por Carreras et al. (2004a). Nuestra plataforma, la cual ya hemos utilizado para categorizar automáticamente comentarios en el trabajo de Santos et al. (2012) nos permite, dada una frase en un idioma concreto, devolver un análisis completo de dicha frase. El proceso que hacíamos con esta plataforma era enviarle la frase obtenida del preprocesamiento en una primera instancia y en una segunda, la frase original. Ambas eran enviadas junto con el idioma en el que estaban escritas, inglés en este caso. El análisis de Freeling, nos devolvía las propiedades de cada palabra dentro del contexto de la frase y en que parte de la estructura de la oración se encuentra cada termino. De este análisis nos quedábamos únicamente con los términos que, a nuestro juicio, son más importantes. Estos términos son los nombres, verbos, adjetivos y adverbios. También recuperábamos la palabra raíz de los términos importantes y, ya que la plataforma nos devuelve la desambiguación que genera Freeling, también utilizamos los *synsetID* más probables, según la desambiguación, para WordNet 3.0[32]. La última versión de WordNet no la utilizamos debido a que, en el momento de hacer este estudio, la herramienta Freeling no la tenía incluida.

Una vez obtenida esa información de los análisis léxico semántico y morfosintáctico, entramos en el paso de utilizar WordNet 3.0. Este paso es completamente dependiente del paso anterior debido a que utiliza los *synsetIDs* para obtener más información y los tipos de palabras (verbo, nombre, adjetivo o adverbio) de la frase. El objetivo es obtener, en base al *synsetID* y al tipo de la palabra, todos los resultados de los elementos similares (e.g., *happy(a)* → [*blessed*, *blissful*, *bright*, *golden*, *halcyon*, *prosperous*, ...]), la descripción propia de la palabra en el contexto dado, el grupo verbal al que pertenece (e.g., *live* → [*dwell*, *inhabit*]) y la nominalizacion (e.g., *happiness(noun)* → [*happy*, *unhappy*], *happy(adjetive)* → [*happiness*, *felicity*]).

4.2.4.2 **Descripción del proceso de búsqueda en motores web**

El paso de obtener más información de buscadores web se realiza con los motores que se han seleccionado en la sección 4.2.2. El proceso, en primer lugar, elimina los elementos innecesarios como son los caracteres no alfanuméricos y las *stopwords*. El objetivo era utilizar solamente las palabras con contenido para realizar una búsqueda Web.

[31] https://github.com/Chispasgg/RESTFreelingClient
[32] https://wordnet.princeton.edu/

El siguiente paso es obtener los topics más representativos en base a la técnica *Latent Dirichlet Allocation* (LDA), desarrollada por Blei et al. (2003) y a la de *Term Frecuency* (TF). LDA es un modelo generativo que permite explicar similitudes entre conjuntos de datos o de elementos no observados en base a conjuntos de elementos observados. Por otra parte, TF es un proceso para obtener la frecuencia con la que un término aparece dentro de un texto. TF suele estar asociado a *Inverse document frequency* (IDF) generando TF-IDF[33]. Esta técnica mide numéricamente la frecuencia de ocurrencia del término en una colección de documentos para determinar lo relevante que es una palabra para un documento en la colección. En nuestro caso particular, no queríamos obtener la relevancia de una palabra en el conjunto de frases totales, sino solamente en su propia frase. Por ello, la parte de determinar si la palabra era común dentro de nuestra colección, es decir, el IDF no nos interesaba.

El uso de LDA o TF para filtrar elementos para la búsqueda viene predeterminado por la configuración. Esta configuración esta detallada en la sección 4.2.4.5. Los topics máximos que se obtienen vienen limitados por la frase. En el caso de LDA, los topics máximos que pueden obtenerse están entre 1 y 50, dependiendo de la longitud de la frase en este punto y, en el caso de TF vienen determinado por un cálculo de la longitud de los topics. Si el número de topics existentes son menos o igual a 10, se utilizan esos 10 topics, en caso contrario, se obtienen únicamente la mitad, es decir, si hay 11 topics, se realiza el cálculo de dividir los 11 topics entre 2 y redondear hacia arriba. El motivo de esta criba se debe a que más elementos introducirían mucho ruido. El valor mínimo de 10 se ha elegido debido para que como mínimo haya 6 topics con los que trabajar. Los topics que se obtienen para realizar las búsquedas están ordenados por la frecuencia de aparición. Es decir, los que más frecuencia de aparición tienen en el caso de TF, o los que tienen mayor peso en el caso de LDA.

El siguiente paso es ejecutar la búsqueda de los topics obtenidos en el paso anterior en los buscadores Bing o Google, según la configuración actual. En el caso de obtener resultados, solo nos quedamos con las 4 primeras entradas, en caso contrario se queda vacío y no se amplía contexto ya que no hay. Toda la información de los resultados obtenidos se almacena junto con la configuración utilizada hasta este punto para preparar el siguiente paso.

4.2.4.3 **Descripción del postproceso de los resultados**

Una vez realizada la búsqueda, se postprocesan los resultados aplicado un filtro de limpieza de valores alfanuméricos y de *stopwords*. Una vez terminado este proceso y en base a la configuración, se les aplica de nuevo la extracción de términos

[33] https://en.wikipedia.org/wiki/Tf-idf

Tabla 4.1: Configuraciones procesadas.s

Recurso	Motor de Búsqueda	Prefiltro	Postfiltro
Frase preprocesada	-	LDA	-
Frase preprocesada	-	TF	-
Frase preprocesada	Google	LDA	LDA
Frase preprocesada	Google	LDA	TF
Frase preprocesada	Google	TF	LDA
Frase preprocesada	Google	TF	TF
Frase preprocesada	Bing	LDA	LDA
Frase preprocesada	Bing	LDA	TF
Frase preprocesada	Bing	TF	LDA
Frase preprocesada	Bing	TF	TF
-	Google	LDA	LDA
-	Google	LDA	TF
-	Google	TF	LDA
-	Google	TF	TF
-	Bing	LDA	LDA
-	Bing	LDA	TF
-	Bing	TF	LDA
-	Bing	TF	TF

con las técnicas de TF y LDA, haciendo el mismo proceso que se ha descrito en el preprocesado de esta sección.

El último paso de la búsqueda en motores web consiste en almacenar los resultados obtenidos, junto con la configuración utilizada para calcular después los resultados. Las combinaciones que hemos utilizado están en la tabla 4.1. En estas combinaciones son las mismas que se utilizaron en el experimento baseline descrito en la sección 4.1. La diferencia con ellas es que se ha escogido las que no utilizaban un prefiltrado de *stopwords* para etiquetar los nuevos recursos. El motivo se debe a que el prefiltrado y postfiltrado de este experimento, para evitar términos no relevantes en los resultados de las búsquedas, eliminan las *stopwords* y los caracteres alfanuméricos evitando su influencia en los filtros de LDA y TF.

Finalmente, los recursos obtenidos son 18 con jerga traducida al lenguaje formal y otros 18 sin traducir y están enumerados en la tabla 4.1. Estos recursos son los que han sido finalmente etiquetados por el sistema de índices. Este sistema es el mismo que se ha utilizado para etiquetar los recursos del experimento baseline descrito en su metodología en la sección 4.1.3 saltándose el paso del preprocesado y con unas particularidades especificas debidas a la naturaleza de los nuevos recursos.

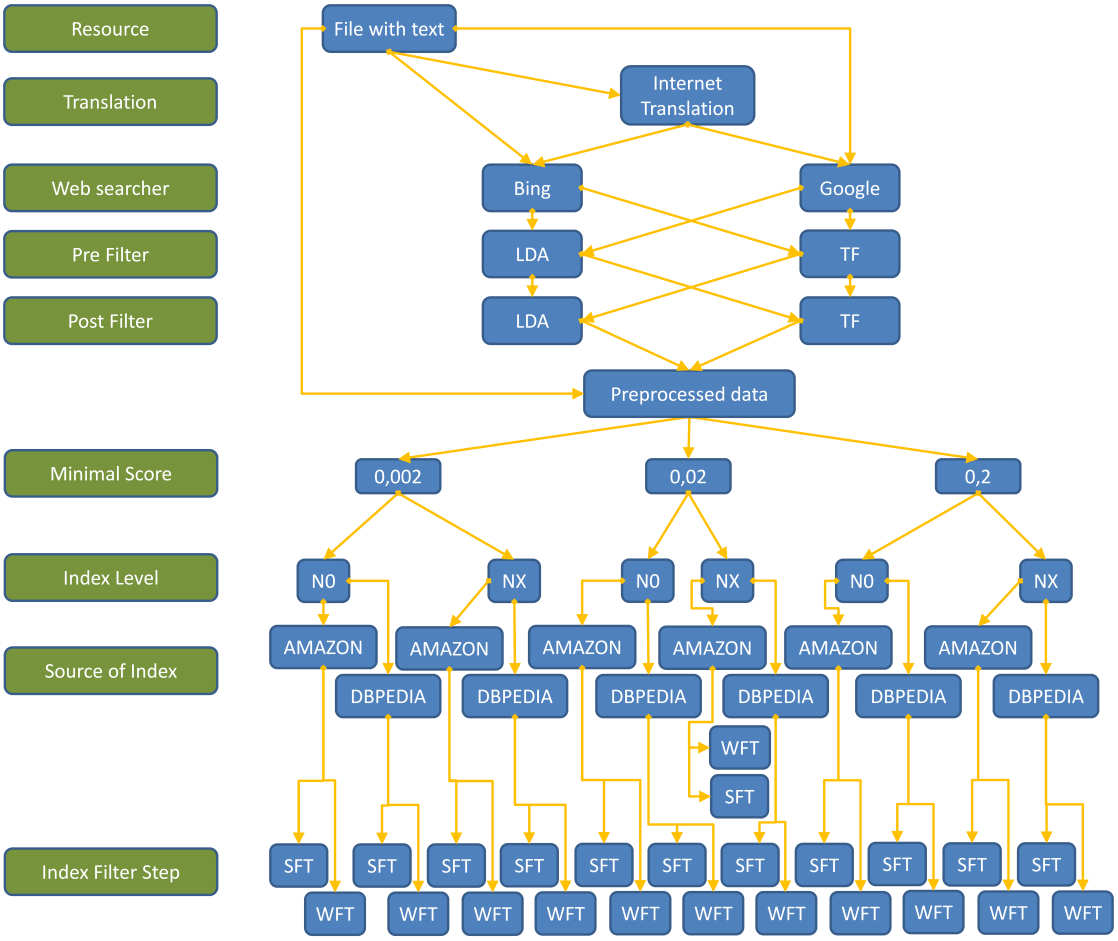


Figura 4.9: Configuraciones utilizadas para realizar las búsquedas del experimento de buscadores web en el indice.

4.2.4.4 Descripción del proceso de búsqueda en los índices

Una vez realizado el prefiltrado, la búsqueda en motores web, el postfiltrado y el almacenamiento de los datos obtenidos, solamente restaba realizar las búsquedas en los índices de Amazon y DBpedia. Las búsquedas en los índices se han hecho con los recursos obtenidos del paso anterior. Estas configuraciones suman 36 recursos, teniendo en cuenta los recursos traducidos y los que no lo están. A cada uno de estos recursos, se les ha aplicado las 24 configuraciones pensadas en el experimento del baseline y detalladas en la sección 4.1.3.3. Esto hace un total de 432 nuevos recursos etiquetados para los elementos traducidos y la misma cantidad para los no traducidos haciendo un total de 864 nuevos recursos para realizar las búsquedas en los índices y probar cual podría ser la mejor configuración. En la figura 4.9 se muestran las diferentes opciones.

Una vez obtenidos los nuevos recursos, el siguiente paso era realizar el procesamiento de dichos recursos. En este proceso se obtenía cada frase con su ampliación de contexto si existía. La ampliación contiene la frase traducida al lenguaje formal y los términos recuperados de los motores web y de WordNet. En el caso de que dicha ampliación no existiera, se realizaba la búsqueda con la frase original. Una vez realizada la búsqueda, los resultados eran almacenados y se ejecutaba el cálculo de comparación de estos resultados contra los generados por los expertos utilizando las métricas LWP, LWR, LCGM, RMSE y MAE, las cuales están detalladas en la sección 4.1.4. También se ejecutaba el cálculo para determinar si nuestro sistema, con una configuración específica, podría sustituir a un experto. Las métricas que se utilizaban para este propósito eran la distancia entre vectores, RMSE y MAE.

4.2.4.5 **Descripción de la configuración utilizada**

Para poder evaluar todas las posibilidades que se contemplaban para estos experimentos, se utilizó un sistema de configuraciones para poder encontrar las variables óptimas. Cada configuración está compuesta por 6 campos que se centran en diferentes aspectos de las búsquedas.

1. **Preprocesado**. Este elemento toma los valores **PS** o vacío y determina si es necesario hacer el preprocesado descrito en la sección 4.2.4.1.

2. **Motor de búsqueda web**. Este elemento toma los valores **BING**, **GOOGLE** o vacío y determina el tipo de motor web para realizar las búsquedas. En las configuraciones en las que no exista este valor, el paso de obtener resultados de motores web no se realiza.

3. **Prefiltrado**. Este elemento toma los valores **LDA** o **TF** y determina el tipo de técnica para prefiltrar los resultados antes de enviarlos a los motores de búsquedas web en el caso de usarlos o, en caso contrario, al sistema de búsqueda en los índices.

4. **Postfiltrado**. Este elemento toma los valores **LDA**, **TF** o vacío y determina el tipo de técnica para postfiltrar los resultados obtenidos de los navegadores antes de enviarlos al sistema de búsqueda en los índices. En las configuraciones en las que no exista este valor, el paso de postfiltrar los resultados de los motores web, no se realiza. Cuando solo exista un elemento de filtrado, se refiere al prefiltrado.

5. **Nivel de búsqueda**. Este elemento toma los valores **N0** o **NX** y determina el tipo de índice sobre el que hacer las búsquedas. Una explicación más detallada se encuentra en la sección 4.1.3.1.

6. **Score**. Este elemento toma los valores **0,2**, **0,02** o **0,002** y determina el límite mínimo para filtrar resultados de los índices.

7. **Filtrado del índice**. Este elemento toma los valores **WFT** o **SFT**y determina el tipo de filtrado.

   El filtrado completo, definido en inglés como **whole filtering type** (WFT). Este tipo está activado con el valor **false** en las configuraciones y su función es filtrar los resultados al finalizar las búsquedas en todos los niveles. Es decir, realizar las búsquedas de clases en cada nivel y filtrar estas búsquedas una vez se hayan recuperado todos los resultados de todos los índices. El valor del filtro es el límite mínimo, definido previamente, y el valor contra el que se compara es el "*Lucene score*". Este valor se obtiene del motor de búsqueda Lucene y expresa la similitud entre la consulta realizada y la información almacenada).

   Por nivel, definido en inglés como **step filtering type** (SFT). Este tipo está activado con el valor **true** en las configuraciones y su función es filtrar resultados de las búsquedas por cada búsqueda en cada nivel. Es decir, realizar una búsqueda en el nivel correspondiente y filtrar en ese mismo momento los resultados con el limite mínimo y el "*Lucene score*". Solo se seguirá buscando en los resultados que superen esta criba. De esta manera, la búsqueda se centra mucho más en áreas específicas.

8. El número máximo de posibles resultados por cada búsqueda, en cada nivel, y de manera global. Es decir, al final solo se obtienen el número de resultados que marca esta variable.

Los valores de algunas variables de la configuración se han determinado, al igual que en el experimento del baseline, en base a la experiencia obtenida en investigaciones previas.

4.3 Deep Learning

4.3.1 Introducción

El DeepLearning es una rama del aprendizaje automático que utiliza cantidades enormes de datos para entrenar modelos de inteligencia artificial, tal y como se puede ver en el trabajo de Bengio (2009). Se utiliza en muchas áreas científicas como el reconocimiento automático del habla, el reconocimiento de imagen, el procesamiento del lenguaje natural, en el descubrimiento de fármacos y toxicologías y en la gestión de las relaciones comerciales. Por ejemplo, en el área del reconocimiento facial, la imagen de una cara puede estar representada de diferentes maneras como

por ejemplo un vector de pixeles. Algunas de estas representaciones hacen más fácil a los sistemas aprender a discernir si esa imagen es una cara humana o no en base a millones de ejemplos. Además, utiliza redes neuronales para representar modelos o elementos del mundo real. El objetivo es generar un sistema que pueda aprender representaciones de datos. Los enfoques y la investigación en esta área intentan definir que representaciones son mejores y como crear los modelos para aprender estas representaciones.

En las áreas que se encuentra este experimento, han obtenido resultados muy satisfactorios. Por ejemplo, en el área de marketing, Fire y Schler (2015) han presentado unos algoritmos que utilizan técnicas de procesamiento de imágenes de DeepLearning, junto con aprendizaje automático y teoría de gafos para investigar como son los anuncios de imágenes en Internet y, en base a esta información, construir unos modelos predictivos que puedan ofrecer una imagen que satisfaga y llegue mejor al usuario. En el área de la recuperación de información, trabajos como el de Deng et al. (2013) muestran que las técnicas de deeplearning son muy útiles y que mejoran las actuales técnicas para recuperar información. Por otra parte, en el área de PLN, ha obtenido muy buenos resultados en varios enfoques. En el del reconocimiento del hablante, los trabajos de Heck et al. (2000), Konig et al. (1998) y Mesnil et al. (2013), entre otros, muestran que la utilización de enfoques y técnicas de aprendizaje profundo obtienen buenos resultados, siendo en algunos casos mejores. El enfoque de generación de vectores por cada topic en cada frase y contexto, también ha obtenido grandes resultados, prueba de ello son los trabajos de Huang et al. (2013) con la generación de la herramienta *sent2vec*[34] que esta enfocada hacia búsquedas web utilizando un enfoque más semántico y el trabajo de Mikolov et al. (2013a,b,c) que ha resultado en la herramienta *word2vec*[35] que trabaja con grandes corpus y representaciones vectoriales. Para este experimento hemos utilizado la herramienta *word2vec* porque la parte semántica, por el momento, no la hemos planteado ni contemplado.

4.3.2 *word2vec*, herramienta de Deep Learning

La herramienta word2vec es el resultado del trabajo de investigación deMikolov et al. (2013a,b,c) con grandes corpus, representaciones vectoriales y deeplearning. Esta herramienta proporciona una implementación eficiente de la técnica "*bag-of-words*" y las arquitecturas "*skip-gram*" para el cálculo de las representaciones vectoriales de palabras. Estas representaciones se pueden utilizar posteriormente en muchas aplicaciones de PLN y de más investigaciones del área. La herramienta necesita un corpus

[34]http://research.microsoft.com/en-us/downloads/731572aa-98e4-4c50-b99d-ae3f0c9562b9/
[35]https://code.google.com/p/word2vec/

de texto como entrada y produce los vectores de palabras como salida. En una primera iteración, construye un vocabulario con los datos de texto de entrenamiento y luego, entera con la representación vectorial de las palabras. Los vectores contenidos en el archivo vectorial resultante, se puede utilizar como características en muchas aplicaciones de PLN y aprendizaje automático. Hay dos algoritmos principales de aprendizaje en la herramienta word2vec: "*continuous bag-of-words*" y "*continuous skip-gram*". Estos algoritmos están disponibles para que el usuario pueda elegir que tipo de algoritmo de aprendizaje utilizar con los modelos. Ambos algoritmos aprenden la representación de una palabra que es útil para la predicción de otras palabras en la oración. Una forma sencilla de ver las representaciones que ha aprendido el modelo es encontrar las palabras más cercanas a una palabra especificada por el usuario. La herramienta de distancia, *distance tool*, sirve ese propósito.

Por ejemplo, si el usuario escribe "*france*", la herramienta de cálculo de distancia mostrará las palabras más similares y sus distancias a la palabra introducida. En la tabla 4.2 se muestra el resultado de las distancias del caso de ejemplo con la palabra *france*.

Tabla 4.2: Resultados generados con la herramienta word2vec para la palabra *france*.

Word	Cousine Distance
spain	0.678515
belgium	0.665923
netherlands	0.652428
italy	0.633130
switzerland	0.622323
luxembourg	0.610033
portugal	0.577154
russia	0.571507
germany	0.563291
catalonia	0.534176

Los modelos generados de vectores de palabras también se pueden utilizar para derivar clases de palabras a partir de conjuntos de datos grandes. Esto se consigue mediante la agrupación de los vectores de palabras con el algoritmo de K-means La salida es un archivo de vocabulario con palabras y sus correspondientes IDs para identificarlas.

4.3.3 Definición del experimento

Este experimento se ha realizado por dos motivos. El primero se debe a que los resultados obtenidos del experimento de buscadores nos mostraban que la ampliación de contexto con topics similares podría ser otra técnica eficaz para poder detectar las temáticas de las frases y, además, seguir pudiendo sustituir a un experto con nuestro

sistema automático. El segundo motivo se debe a que deeplearning es un enfoque muy interesante y que ha demostrado gran eficacia en el área del procesamiento del lenguaje natural y de recuperación de la información como bien se ha mostrado en la sección 4.3.1. En nuestro caso, hemos planteado la utilización de la herramienta *word2vec*, explicada en la sección 4.3.2, para utilizar en la expansión del contexto. Debido a que esta herramienta necesita modelos vectoriales para poder calcular las similitudes y distancias entre palabras, hemos utilizado dos modelos diferentes. Uno de ellos generado por nosotros mismos y otro obtenido mediante descarga de la web de la herramienta. Estos modelos son, el de Google News que es un modelo ya generado con los vectores de cada noticia creada para la plataforma Google News y que cuenta con más de 100 billones americanos de palabras y esta generado con *skip-ngrams*, y el de Wikipedia. Este último, es el volcado de todos los artículos de la enciclopedia libre Wikipedia. El volcado que nosotros utilizamos corresponde al mes de Febrero del año 2015. En este conjunto de datos existen más de 3 billones americanos de palabras y para la generación del modelo, también utilizamos *skip-ngrams*.

Los recursos que hemos utilizado para realizar este experimento, como la base de conocimiento de contextos y productos y las frases etiquetadas, son las mismas que para el experimento baseline descrito en la sección 4.1. Partiendo de estos recursos, se han planteado 2 experimentos, todos ellos se han comparado con el baseline para comprobar los resultados:

1. En el primer experimento, se intentó validar la eficacia del método propuesto mediante la comparación de los resultados del baseline con los resultados obtenidos por el sistema automático. Concretamente los resultados obtenidos por la herramienta con los diferentes modelos, tanto con las frases traducidas al lenguaje formal como sin la traducción

2. En el segundo experimento, se sustituyeron iterativamente cada experto con los resultados del sistema automático (i.e. si tenemos 5 expertos, utilizamos los resultados de 4 expertos junto con los resultados del sistema automático para compararlo con los resultados de los 5 expertos) para verificar si es posible sustituir a un experto con nuestro sistema automático. Este tipo de validación es el seleccionado para entregar el premio Loebner en Inteligencia Artificial[36]. Este premio esta basado en el paper de Alan Turing *Computing Machinery and Intelligence* Turing (1950).

[36]http://www.loebner.net/Prizef/loebner-prize.html

4.3.4 Metodología para aumentar el contexto

El proceso, utilizando los recursos seleccionados junto con un sistema basado en configuraciones, está dividido y definido en 3 pasos básicos, según la configuración que se utilice, para obtener todos los posibles resultados. Estos pasos son:

- Preprocesar cada frase de la misma manera que se describe en la sección 4.3.4.1.
- Realizar búsquedas con los términos de cada frase contra el modelo vectorial de *word2vec* seleccionado, obteniendo un número finito de resultados.
- Postprocesar la frase y realizar búsquedas en el indice de Amazon y DBpedia.

4.3.4.1 Descripción del preprocesamiento utilizado

El preprocesamiento de las frases en este experimento, es el proceso que traduce del lenguaje vulgar al formal los términos que contienen las frases y, después, elimina los elementos innecesarios como son los caracteres no alfanuméricos y las *stopwords*. El objetivo era utilizar solamente los términos importantes para obtener otros términos que tengan una distancia máxima determinada.

Para el paso de la traducción de las frases, al igual que en el experimento de los buscadores de la sección 4.2.4.1, se utilizó el servicio web de *lingo2word*[37]. La traducción generada se almacena junto a la frase original. Una característica importante de este servicio es que, si la frase no está en lenguaje coloquial, el resultado que genera es el mismo que la entrada. Además, en ocasiones mejora a las propias frases originales mediante la utilización de sinónimos que disminuyen la ambigüedad (e.g., *car* se traduce a *automobile*).

4.3.4.2 Descripción del proceso de ampliación de contexto con los modelos

El proceso de ampliación de contexto en este experimento constaba de introducir nuevos términos obtenidos de los modelos de *deeplearning* Google News y Wikipedia. Para ello, se procesaba cada término de la frase y se buscaban sus 5 términos más próximos según cada modelo y teniendo en cuenta las diferentes configuraciones que se muestran en la tabla 4.3. Estos términos nos los devuelve la herramienta *word2vec*. El resultado es una ampliación de las frases con elementos cercanos, según cada modelo, para ampliar su contexto.

Una vez realizada la ampliación de términos con los modelos vectoriales, no hace falta realizar ningún postprocesamiento especial debido a que todos los términos

[37] http://www.lingo2word.com/

Tabla 4.3: Configuraciones procesadas

Recurso	Modelo
texto sin traducir	Wikipedia
texto sin traducir	Google News
texto traducido	Wikipedia
texto traducido	Google News

obtenidos son importantes. Como en el paso del prefiltrado se eliminaron los valores alfanuméricos y las *stopwords*, estos no han sido utilizados para las búsquedas. Además, los modelos no devuelven este tipo de términos por lo que, en conclusión, no hace falta un postprocesado porque no hay elementos para eliminar. Lo único que se hace es almacenar los nuevos recursos mostrando cual ha sido su fuente. Esta fuente son sus configuraciones, y se muestra en la tabla 4.3. En total se han obtenido

4.3.4.3 **Descripción del proceso de búsqueda en los índices**

La búsqueda en los índices de Amazon y DBpedia, al igual que en los anteriores experimentos, se realiza para comprobar, contra el experimento baseline, las posibilidades que tiene este enfoque. Los elementos que se utilizan para realizar las búsquedas son los recursos que se han obtenido en el paso anterior de ampliación del contexto. El número de recursos que se ha generado, teniendo en cuenta las diferentes configuraciones que se han utilizado y que están descritas en la sección 4.3.4.4 es de 96. Se ha partido de 2 recursos según los modelos utilizados y si han sido o no traducidos al lenguaje formal. Utilizando las 24 configuraciones base del experimento baseline que se muestran en la figura 4.10, pero evitando las que tienen *stopwords* ya que no son necesarias debido a que no existen estos elementos en las frases actuales, se logran 48 nuevos recursos para cada modelo, que la suma de ambos nos devuelve 96 nuevos recursos etiquetados. Estos recursos son los que han sido comparados y procesados con los índices.

Una vez realizada la búsqueda, los resultados eran almacenados y se ejecutaba el cálculo de comparación de estos resultados contra los generados por los expertos utilizando las métricas LWP, LWR, LCGM, RMSE y MAE, las cuales están detalladas en la sección 4.1.4. También se ejecutaba el cálculo para determinar si nuestro sistema, con una configuración específica, podría sustituir a un experto. Las métricas que se utilizaban para este propósito eran la distancia entre vectores, RMSE y MAE.

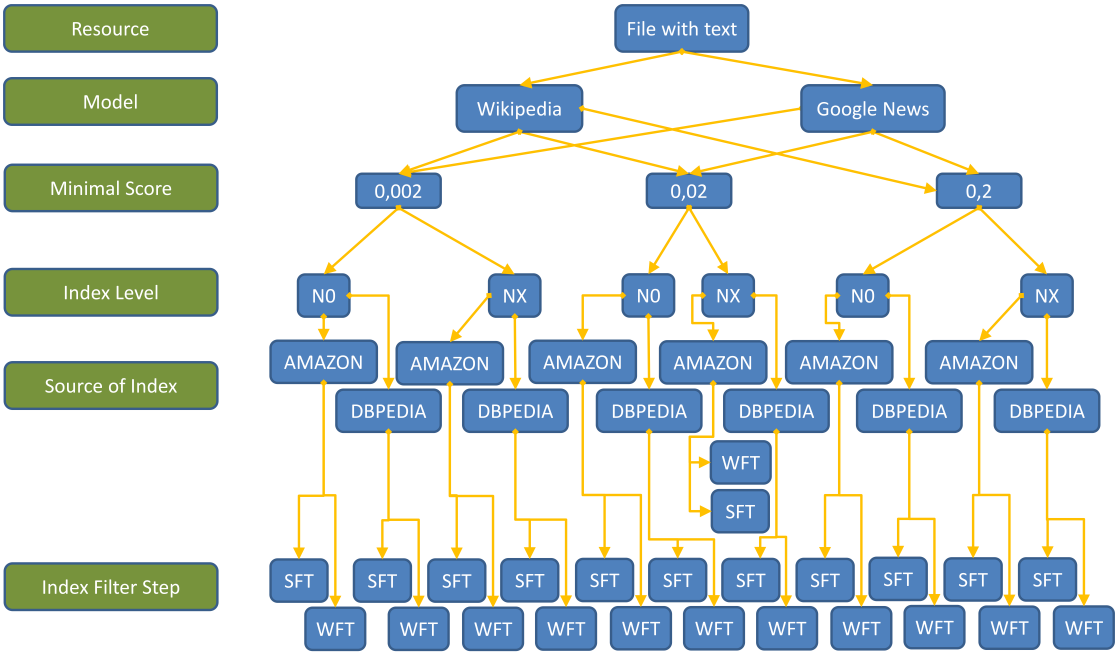


Figura 4.10: Configuraciones utilizadas en el experimento de *depplearning* para realizar las búsquedas en el indice.

4.3.4.4 Descripción de la configuración utilizada

Para poder evaluar todas las posibilidades que se contemplaban para estos experimentos, se utilizó un sistema de configuraciones para poder encontrar las variables óptimas. Cada configuración está compuesta por 6 campos que se centran en diferentes aspectos de las búsquedas.

1. **Modelo utilizado**. Este elemento toma los valores **Wikipedia** o **Google News** y determina el modelo utilizado para obtener los términos de la ampliación del contexto.

2. **Nivel de búsqueda**. Este elemento toma los valores **N0** o **NX** y determina el tipo de índice sobre el que hacer las búsquedas. Una explicación más detallada se encuentra en la sección 4.1.3.1.

3. **Filtrado del índice**. Este elemento toma los valores **WFT** o **SFT**y determina el tipo de filtrado.

4. **Score**. Este elemento toma los valores **0,2**, **0,02** o **0,002** y determina el límite mínimo para filtrar resultados de los índices.

   El filtrado completo, definido en inglés como **whole filtering type** (WFT). Este tipo está activado con el valor **false** en las configuraciones y su función

es filtrar los resultados al finalizar las búsquedas en todos los niveles. Es decir, realizar las búsquedas de clases en cada nivel y filtrar estas búsquedas una vez se hayan recuperado todos los resultados de todos los índices. El valor del filtro es el límite mínimo, definido previamente, y el valor contra el que se compara es el "*Lucene score*". Este valor se obtiene del motor de búsqueda Lucene y expresa la similitud entre la consulta realizada y la información almacenada).

Por nivel, definido en inglés como **step filtering type** (SFT). Este tipo está activado con el valor **true** en las configuraciones y su función es filtrar resultados de las búsquedas por cada búsqueda en cada nivel. Es decir, realizar una búsqueda en el nivel correspondiente y filtrar en ese mismo momento los resultados con el limite mínimo y el "*Lucene score*". Solo se seguirá buscando en los resultados que superen esta criba. De esta manera, la búsqueda se centra mucho más en áreas específicas.

5. El número máximo de posibles resultados por cada búsqueda, en cada nivel, y de manera global. Es decir, al final solo se obtienen el número de resultados que marca esta variable.

Los valores de algunas variables de la configuración se han determinado, al igual que en el experimento del baseline, en base a la experiencia obtenida en investigaciones previas.

4.4 Traducción de términos entre Amazon y DBpedia utilizando el mapa generado

4.4.1 Introducción

En anteriores experimentos, el enfoque utilizado era determinar qué métodos eran eficaces basándonos en que sus resultados no se desviase demasiado de los resultados de los expertos. La manera de comprobarlo era teniendo unos resultados semejantes a ellos para poder sustituir a un experto. Estos enfoques se han utilizado para comprobar los resultados tanto de Amazon como de DBpedia. Después de probar varios enfoques, quedaba centrarse en el objetivo principal, ofrecer categorías de productos de libros de Amazon en base a contextos, en nuestro caso, topics de DBpedia. Para realizar este experimento, hemos utilizado el mapa que hemos creado y que esta descrito en la sección 3.3. Este mapa relaciona las categorías de libros de Amazon con los topics de DBpedia. En este experimento se ha utilizado los recursos de DBpedia generados en el experimento baseline para obtener nuevos recursos de Amazon y poder compararlos contra los resultados del baseline de Amazon.

4.4.2 Definición del experimento

Este experimento tenia como objetivo utilizar el mapa de referencias generado y que se ha descrito en la sección 3.3 para ofrecer productos de Amazon en base a contextos detectados de DBpedia. A diferencia del resto de experimentos detallados hasta el momento, este ya se centra en ofrecer productos en base a contextos. Para ello se utilizaron los 24 recursos finales del experimento baseline de DBpedia para traducirlos a Amazon y realizar las evaluaciones y comparaciones. De esta manera, se podía determinar si existía una gran diferencia entre los resultados del baseline de Amazon y los resultados de este experimento. Además, también se comparaba sí la diferencia de opinión con los resultados de los expertos era lo suficientemente optima como para plantear este método como otro más de nuestro sistema para la detección de temáticas y para el ofrecimiento de productos.

Los recursos que hemos utilizado para realizar este experimento, como la base de conocimiento de contextos y productos y las frases etiquetadas, son las mismas que para el experimento baseline descrito en la sección 4.1. Partiendo de estos recursos, se han planteado 2 experimentos, todos ellos se han comparado con el baseline para comprobar los resultados:

1. En el primer experimento, se intentó validar la eficacia del método propuesto mediante la comparación de los resultados del baseline con los resultados obtenidos por el sistema automático. Concretamente los resultados obtenidos por la traducción del mapa de topics de DBpedia a categorías de Amazon, utilizando los recursos que previamente se habían obtenido en el experimento del baseline.

2. En el segundo experimento, se sustituyeron iterativamente cada experto con los resultados del sistema automático (i.e. si tenemos 5 expertos, utilizamos los resultados de 4 expertos junto con los resultados del sistema automático para compararlo con los resultados de los 5 expertos) para verificar si es posible sustituir a un experto con nuestro sistema automático. Este tipo de validación es el seleccionado para entregar el premio Loebner en Inteligencia Artificial[38]. Este premio está basado en el paper de Turing (1950) *Computing Machinery and Intelligence* .

[38] http://www.loebner.net/Prizef/loebner-prize.html

4.4.3 Metodología para traducir topics de contexto a categorías de productos

Para realizar los experimentos de este enfoque, no se han utilizado directamente los índices utilizados en el experimento baseline descrito en la sección 4.1, pero si se han utilizado sus resultados, concretamente los 24 recursos finales de DBpedia. Una ventaja a la hora de realizar este experimento ha sido que no hemos tenido que preprocesar ni postprocesar ningún elemento, ya que estos procesos ya estaban realizados y descritos en la sección 4.1.2. El único proceso que se ha realizado es el de la traducción de los resultados de DBpedia del experimento baseline a Amazon, utilizando el mapa de traducciones generado y descrito en la sección 3.3. El proceso ha seguido 3 pasos básicos para obtener los resultados utilizando los recursos heredados.

- Obtener los topics de DBpedia de cada frase.
- Calcular las traducciones para cada topic de DBpedia de cada frase en base al nivel en que se buscaron (N0 o NX).
- Obtener un número determinado de categorías de Amazon.

4.4.4 Descripción del proceso de traducción

El proceso seguido para realizar la traducción de los topics de DBpedia a las categorías de Amazon, tiene singularidades específicas debido a la naturaleza del mapa de traducción. Estas singularidades son semejantes a las que tiene un diccionario común. Tomemos, por ejemplo, la definición de un término. Dicha definición puede no estar dentro del diccionario por conjugaciones o cualquier tipo de accidente gramatical[39]. La manera de conocer su significado es obtener la raíz del término y buscarlo en el diccionario. En nuestro mapa ocurre algo semejante. Como ya se ha comentado en la definición de dicho mapa, concretamente en la sección 3.3.2, existen unos pocos términos que no tienen traducción directa. Por este motivo, hemos tenido que realizar unos ajustes semejantes a los que se hacen con los términos conjugados que no existen en los diccionarios.

El proceso se inicia con la obtención de las frases y de los topics de contexto de DBpedia. Estos elementos se obtienen de los recursos finales ya etiquetados en el experimento baseline. Una vez obtenidos, se procesan contra el mapa de traducciones. Existen 2 opciones; que la traducción directa exista o que no exista. En el caso de existir, se obtiene su traducción y se le asigna el mayor peso posible de éxito, el peso

[39] https://es.wikipedia.org/wiki/Verbo

1. En caso de que el elemento fuera el topic **Ninguno**, se asigna la misma categoría correspondiente en Amazon, es decir, categoría **Ninguno** con el peso 1 también. El caso de que no exista la traducción directa, el proceso es un poco más complicado. Entra en juego otra variable como es el nivel en el que se han buscado las etiquetas (N0 o NX). El peso también varía dividiéndose entre el número de iteraciones que se tienen que realizar, pero sin olvidar que este valor nunca superara el peso 1.

Si el nivel en el que se han buscado los topics de DBpedia es el N0, el peso se divide a la mitad por cada escalón que tenga que bajar hasta encontrar una traducción en los hijos directos del término actual. El motivo de que el valor del peso sea menor es porque, al no ser una traducción directa y utilizar la de sus hijos, los resultados serán más numerosos, lo cual dará más importancia a este elemento sobre otros que han tenido traducción directa y, en principio, son mejores traducciones. Si, en cambio, el nivel en el que se han buscado los topics de DBpedia es el NX, como varios de los elementos más pequeño, es decir, los que están denominado como hoja, no tiene traducción directa, se sube hacia arriba en el árbol de nodos padre hasta encontrar dicha traducción. El mecanismo de reducción de importancia es el mismo que en N0, solo que en vez de bajar a los hijos, se sube a los padres. El motivo es que partiendo de los nodos hojas, las posibilidades también son muy numerosas. Además, a la hora de ir subiendo niveles, los elementos se hacen más generalistas, lo que se traduce en una perdida del foco original del significado de la traducción.

Una vez terminada la obtención de todas las traducciones de los términos de DBpedia a Amazon y calculados sus pesos para cada termino (recordemos que como máximo puede ser 1), se elevan estos resultados a la categoría padre, es decir, la categoría principal. De esta manera, las categorías principales obtienen votos de las categorías que abarcan y van aumentando su importancia.

El paso final es, una vez que todos los elementos han sido elevados hasta la categoría principal, se dividen los valores de sus pesos entre el número total de términos de DBpedia buscados. Lo que se ha buscado es que el peso de cada término sea como máximo de 1. Después de esto, se ordenan de mayor a menor obteniendo el listado final. Este listado esta ordenado según la importancia que ha tenido cada categoría y además, se ha filtrado para recuperar un número finito de resultados importantes. Este valor es 10 y ha sido heredado de configuraciones anteriores. Por otra parte, las configuraciones utilizadas son las mismas que en el experimento baseline que se muestran en la figura 4.4. Al igual que en ese experimento, los resultados obtenidos son 24.

"De nada sirve rezar Flanders, yo mismo acabo de hacerlo y no vamos a ganar los dos."

Homer Jay Simpson
(1956 –)

5

Resultados

En los experimentos que se han realizado, se han generado unos resultados que muestran diferentes enfoques viables para proseguir con la investigación. En este capítulo se mostrarán todos los resultados, ordenados por experimento según la posición en los que han sido descritos en el capítulo 4.

En primer lugar se mostraran los resultados del experimento base o baseline, descrito en la sección 4.1. Este experimento es la base contra la que hemos comparado el resto de experimentos. Estos resultados están divididos en 2 apartados. El primero es el que, utilizando las formulas detalladas en la sección 4.1.4, clasifica los resultados de DBpedia y Amazon. El segundo es el que nos muestra los resultados de la sustitución de un experto por el sistema automático y los clasifica en base a las métricas que también están detalladas en la misma sección que las anteriores. En la parte de sustituir a un experto, hay que tener en cuenta que, las diferencias entre los resultados de todos los expertos contra los resultados medios de sustituir a un experto suponen la manera de calcular si el sistema realmente puede ser útil para sustituir a un experto por el sistema correspondiente con una configuración especifica.

Para el resto de los experimentos, los resultados también están divididos en 2 apartados, pero el análisis se hace comparando los resultados con el del baseline. La primera parte de los resultados es para validar la eficacia del método propuesto mediante la comparación de los resultados obtenidos y los del experimento base. El objetivo es encontrar un sistema con una configuración específica que mejore o iguale, dentro de rangos de calidad, los aciertos del sistema con los de los expertos.

Esta parte corresponde a las métricas LWP, LWR, LCGM, RMSE y MAE. Esto nos ofrece un sistema que puede comportarse como un experto.

La segunda parte de los resultados es la comparación y sustitución iterativa de cada experto con los resultados del sistema automático (i.e. si tenemos 5 expertos, utilizamos los resultados de 4 expertos junto con los resultados del sistema automático para compararlos con los resultados de los 5 expertos) para verificar si es posible sustituir a un experto por nuestro sistema automático. Seguidamente, se sigue el proceso detallado en la sección 4.1.4.4, uniendo las clases ofrecidas por los expertos y predichas por el sistema automático para generar el vector, al cual nosotros llamamos vector del "experto común".

El objetivo es verificar que los resultados que se obtienen con ese sistema y esa configuración no difieren en exceso, dentro de unos rangos de calidad, de las respuestas que pueden dar los expertos. Para ello, se calculan las distancias vectoriales entre las respuestas que han dado los expertos con las respuestas medias de sustituir a cada uno de los expertos por los resultados del sistema con una configuración especifica. Los valores de las distancias están entre 0 y 2 debido a que los vectores que se generan, como máximo pueden tomar valor 1, tanto para los expertos como para la máquina. El máximo valor posible es 2 debido a que las opciones posibles son; i) que se encuentre el topic completamente (distancia 0), ii) parcialmente (distancia entre 0 y 2) o iii) que se diga un topic completamente diferente del buscado (distancia 2). A estos valores hay que restarle 0,7 por la gestión del topic *Ninguno* debido a que es excluyente y abarcar un 35 % de las respuestas de los expertos. Finalmente, la distancia máxima, siendo el peor resultado posible, se queda en 1,3. Con este proceso se consigue verificar si el consenso que los expertos habían mostrado con los resultados sigue manteniéndose, pero habiendo sustituido a uno de ellos por un sistema y configuración especifica.

Los resultados que hemos obtenido, nos permiten soñar con que el sistema, aunque necesite mejorar y puede mejorar, es una opción y enfoque viable para la detección de la temática en frases cortas.

5.1 Resultados Baseline

En el primer experimento, el sistema automático devolvió un mínimo de 1919 votos para ambas fuentes, Amazon y DBpedia, con la configuración más restrictiva. En cambio, con la más laxa devolvió entre 14000 y 12000 votos para Amazon y DBpedia respectivamente. En la configuración más restrictiva, los votos fueron 1919 debido a que el sistema ofrecía solamente el elemento con puntuación más alta, siendo un único resultado por cada frase. En cuanto a los votos de los expertos, después de unirlos, sumaron 5405 para Amazon y 3731 para DBpedia.

Por otra parte, se analizó cada métrica para determinar cuál era la que mejor reflejaba la realidad de los algoritmos para evaluar los resultados. Este análisis consistía en comprobar la correlación entre los resultados obtenidos y el conocimiento general de qué configuraciones debían comportarse mejor según sus variables *M* y *C*, definidas en la sección 4.1.4. Una vez obtenidos los resultados y sabiendo que configuraciones daban mejor en base a la experiencia que obtuvimos en el trabajo de Laorden et al. (2014), decidimos tomar LWP como métrica principal, por la cual, ordenar los resultados. El motivo era que mantenía la coherencia esperada con las configuraciones y, además, también con las métricas RMSE, MAE y las variables *M* y *C*. Por otra parte y como puede verse en los resultados, LWP está en sintonía con el resto de métricas en mayor medida que el resto de métricas y, lo que es más importante, tanto para DBpedia como para Amazon. Del mismo modo, y teniendo en cuenta los mismos criterios de correlación entre resultados obtenidos y conocimiento adquirido, los resultados de sustitución de un experto por un sistema automático se ordenaron por la métrica MAE.

Para una mejor comprensión de los resultados, las tablas se han coloreado de verde (mejor resultado) a rojo (peor resultado), pasando por amarillo y naranja, en base al rango de resultados que se han obtenido.

En la tabla 5.1 se muestran los resultados obtenidos con la fuente Amazon y en la tabla 5.2 los resultados obtenidos con la fuente DBpedia. Estos resultados son la base contra la que se han comparado el resto de los enfoques que hemos planteado en este trabajo.

Además, este experimento también es la base contra la que se han comparado el resto de enfoques para la sustitución de un experto por un sistema automático. En la tabla 5.3 se muestran los resultados de la sustitución iterativa de un experto con la fuente de Amazon, y en la tabla 5.3 los resultados, siguiendo el mismo proceso pero para la fuente DBpedia.

5.1.1 Análisis de los Resultados de Amazon

Los resultados de la fuente Amazon son bastante homogéneos. Los algoritmos se han comportado tal y como esperábamos, siendo los mejores los más restrictivos (score 0.2). La configuración que mejores resultados ha obtenido ha sido NX-0.2-SFT-no (el tipo de indice a buscar X, 0.2 de score, filtrando a cada nivel de ejecución y sin usar stopwords). Los resultados que ha obtenido son, para LWP = 79.9928, para LWR = 27.3639, para LCGM = 62.2153, para RMSE = 0.0817 y para MAE = 0.1235. Además, y siguiendo con los resultados esperados, las configuraciones menos restrictivas (score 0.002) han obtenido los peores resultados. Siendo la mejor de estas configuraciones NX-0.002-WFT-no con LWP = 46.7756, LWR = 22.2616,

Tabla 5.1: Resultados del experimento baseline con la fuente Amazon

Configuración	M	E	C	LWP avg	LWR avg	LCGM avg	RMSE	MAE
NX-0,2-SFT-no	1922	5405	1620	79.9928	27.3639	62.2153	0.0817	0.1235
NX-0,2-WFT-no	1919	5405	1619	79.9668	27.2862	62.2565	0.0814	0.1232
N0-0,2-WFT-yes	1919	5405	1619	79.9668	27.2862	62.2565	0.0814	0.1232
N0-0,2-WFT-no	1919	5405	1619	79.9668	27.2862	62.2565	0.0814	0.1232
NX-0,2-WFT-yes	1919	5405	1619	79.9668	27.2862	62.2565	0.0814	0.1232
N0-0,2-SFT-yes	1919	5405	1619	79.9668	27.2862	62.2565	0.0814	0.1232
N0-0,2-SFT-no	1919	5405	1619	79.9668	27.2862	62.2565	0.0814	0.1232
NX-0,2-SFT-yes	1926	5405	1621	79.9398	27.4296	62.2585	0.0822	0.1239
NX-0,02-WFT-no	2172	5405	1613	77.7358	27.0394	60.6759	0.0843	0.1274
N0-0,02-WFT-no	2317	5405	1618	77.6196	26.9289	60.6291	0.0839	0.1275
N0-0,02-SFT-no	2317	5405	1618	77.6196	26.9289	60.6291	0.0839	0.1275
NX-0,02-WFT-yes	3581	5405	1651	68.0122	26.0962	54.8966	0.0934	0.1451
N0-0,02-SFT-yes	4692	5405	1719	67.2167	25.3621	54.6287	0.0907	0.1462
N0-0,02-WFT-yes	4692	5405	1719	67.2167	25.3621	54.6287	0.0907	0.1462
NX-0,02-SFT-no	5315	5405	1482	49.364	21.0499	41.3347	0.1083	0.1678
NX-0,002-WFT-no	6616	5405	1639	46.7756	22.2616	41.0003	0.1022	0.1703
N0-0,002-SFT-no	9318	5405	1879	45.6644	21.7376	41.2037	0.0965	0.1716
N0-0,002-WFT-no	9318	5405	1879	45.6644	21.7376	41.2037	0.0965	0.1716
NX-0,002-WFT-yes	9552	5405	1632	29.5793	18.8604	29.4706	0.1039	0.1872
N0-0,002-WFT-yes	14081	5405	1978	27.6465	17.2392	29.4847	0.0962	0.1895
N0-0,002-SFT-yes	14081	5405	1978	27.6465	17.2392	29.4847	0.0962	0.1895
NX-0,02-SFT-yes	9610	5405	1545	27.0886	17.4309	27.2597	0.105	0.1894
NX-0,002-SFT-yes	11977	5405	1614	20.2858	16.273	23.0868	0.0987	0.1951
NX-0,002-SFT-no	12200	5405	1595	19.9845	15.5924	22.7241	0.0989	0.1956

Tabla 5.2: Resultados del experimento baseline con la fuente DBpedia

Configuración	M	E	C	LWP avg	LWR avg	LCGM avg	RMSE	MAE
NX-0,2-SFT-yes	1919	3731	1185	60.1799	27.0033	49.262	0.1112	0.1394
N0-0,2-WFT-yes	1919	3731	1184	60.134	26.9407	49.2387	0.1112	0.1394
NX-0,2-WFT-yes	1919	3731	1184	60.134	26.9407	49.2387	0.1112	0.1394
N0-0,2-SFT-yes	1919	3731	1184	60.134	26.9407	49.2387	0.1112	0.1394
NX-0,2-WFT-no	1919	3731	1183	60.0818	26.8886	49.1866	0.1112	0.1394
NX-0,2-SFT-no	1919	3731	1183	60.0818	26.8886	49.1866	0.1112	0.1394
N0-0,2-WFT-no	1919	3731	1183	60.0818	26.8886	49.1866	0.1112	0.1394
N0-0,2-SFT-no	1919	3731	1183	60.0818	26.8886	49.1866	0.1112	0.1394
NX-0,02-SFT-no	2183	3731	1235	59.2671	28.7652	48.851	0.1145	0.1443
NX-0,02-WFT-no	2092	3731	1179	58.446	26.7239	48.1676	0.1139	0.1433
N0-0,02-WFT-no	2204	3731	1183	58.4155	26.6165	48.1703	0.1135	0.1435
N0-0,02-SFT-no	2204	3731	1183	58.4155	26.6165	48.1703	0.1135	0.1435
NX-0,02-SFT-yes	3192	3731	1467	55.9973	33.8481	49.148	0.1153	0.1521
NX-0,02-WFT-yes	2861	3731	1199	53.0463	25.9565	45.1467	0.124	0.158
N0-0,02-SFT-yes	3678	3731	1264	52.5402	25.746	45.1971	0.1213	0.1592
N0-0,02-WFT-yes	3678	3731	1264	52.5402	25.746	45.1971	0.1213	0.1592
NX-0,002-SFT-no	5505	3731	1724	43.0782	37.6167	43.7891	0.1118	0.1645
NX-0,002-WFT-no	4929	3731	1469	42.8318	29.7816	41.2949	0.124	0.1713
N0-0,002-SFT-no	8569	3731	1659	38.3266	24.8941	39.1045	0.121	0.1806
N0-0,002-WFT-no	8569	3731	1659	38.3266	24.8941	39.1045	0.121	0.1806
NX-0,002-SFT-yes	6390	3731	1807	38.1133	38.4248	41.9007	0.1074	0.1664
NX-0,002-WFT-yes	5882	3731	1615	37.6534	33.2671	39.7548	0.1201	0.1734
N0-0,002-WFT-yes	12103	3731	1842	28.9468	22.4009	34.5301	0.1184	0.1915
N0-0,002-SFT-yes	12103	3731	1842	28.9468	22.4009	34.5301	0.1184	0.1915

Tabla 5.3: Resultados experimento baseline de sustitución experto, Amazon

Configuración	RMSE	MAE	DISTANCIA VECTORIAL 0.1	0.2	0.3	0.4<
N0-0,2-SFT-yes	0.0187	0.0521	1134.8	426.4	357.8	0
N0-0,2-SFT-no	0.0187	0.0521	1134.8	426.4	357.8	0
N0-0,2-WFT-yes	0.0187	0.0521	1134.8	426.4	357.8	0
NX-0,2-WFT-yes	0.0187	0.0521	1134.8	426.4	357.8	0
NX-0,2-WFT-no	0.0187	0.0521	1134.8	426.4	357.8	0
N0-0,2-WFT-no	0.0187	0.0521	1134.8	426.4	357.8	0
NX-0,2-SFT-no	0.0187	0.0522	1131.4	427.4	360.2	0
NX-0,2-SFT-yes	0.0188	0.0524	1131.8	426.2	361	0
NX-0,02-WFT-no	0.0192	0.0541	1117.4	429.8	371.8	0
N0-0,02-SFT-no	0.0191	0.0542	1120.4	430.4	368.2	0
N0-0,02-WFT-no	0.0191	0.0542	1120.4	430.4	368.2	0
NX-0,02-WFT-yes	0.0207	0.0627	1032.4	509.2	377.4	0
N0-0,02-SFT-yes	0.0203	0.0633	1062.8	488.8	367.4	0
N0-0,02-WFT-yes	0.0203	0.0633	1062.8	488.8	367.4	0
NX-0,02-SFT-no	0.0233	0.0731	898.2	564.4	456.4	0
NX-0,002-WFT-no	0.0223	0.0745	944.8	603.8	370.4	0
N0-0,002-WFT-no	0.0213	0.0752	1023.8	554.6	340.6	0
N0-0,002-SFT-no	0.0213	0.0752	1023.8	554.6	340.6	0
NX-0,002-WFT-yes	0.0226	0.0823	939.8	674.4	304.8	0
NX-0,02-SFT-yes	0.0228	0.0831	918.4	687.4	313.2	0
N0-0,002-WFT-yes	0.0212	0.0834	1051.2	597.6	270.2	0
N0-0,002-SFT-yes	0.0212	0.0834	1051.2	597.6	270.2	0
NX-0,002-SFT-yes	0.0217	0.0859	1038.2	650	230.8	0
NX-0,002-SFT-no	0.0217	0.0861	1042.4	635.8	240.8	0

LCGM = 41.0003, RMSE = 0.1022 y MAE = 0.1703.

Los resultados de Amazon muestran que todas las métricas tienen un comportamiento similar. Se pueden utilizar cualquiera de ellas para elegir una configuración debido a que coinciden con los resultados esperados, según el tipo de restricción de las configuraciones. La única métrica que se desvía ligeramente de LWP y del resto es RMSE. El motivo se debe a su naturaleza, la cual penaliza las grandes cantidades de errores, pero únicamente penaliza con las configuraciones poco restrictivas, lo cual estaba contemplado.

Los resultados muestran que las diferencias entre la configuración más restrictiva (NX-0.2-SFT-no) y unas configuraciones más permisivas, con 0.02 de score, no penalizan en exceso los resultados, apenas 2.4 para LWR, 0.4 para LWR, 1.6 para LCGM, 0.0021 para RMSE y 0.0040 para MAE.

En los resultados del experimento de reemplazar a un experto por el sistema que utilizaba la fuente de Amazon, se puede observar que el efecto de reemplazar a un experto por el sistema automático es muy poco significante. Esto se puede observar en los resultados de las medias de las distancias vectoriales. En estos resultados, la distancia entre la mejor y la peor configuración no sobrepasa el 0,034. Utilizando este enfoque, hemos podido evaluar la desviación de los resultados finales del criterio

Tabla 5.4: Resultados experimento baseline de sustitución experto, DBpedia

Configuración	RMSE	MAE	DISTANCIA VECTORIAL 0.1	0.2	0.3	0.4<
NX-0,2-WFT-no	0.023	0.062	737	593.2	588.8	0
NX-0,2-SFT-no	0.023	0.062	737	593.2	588.8	0
N0-0,2-WFT-no	0.023	0.062	737	593.2	588.8	0
N0-0,2-SFT-no	0.023	0.062	737	593.2	588.8	0
NX-0,2-WFT-yes	0.023	0.062	737	592.4	589.6	0
N0-0,2-WFT-yes	0.023	0.062	737	592.4	589.6	0
N0-0,2-SFT-yes	0.023	0.062	737	592.4	589.6	0
NX-0,2-SFT-yes	0.023	0.062	734.6	591.6	592.8	0
NX-0,02-WFT-no	0.024	0.065	701.4	585.6	632	0
N0-0,02-WFT-no	0.024	0.065	710.2	580	628.8	0
N0-0,02-SFT-no	0.024	0.065	710.2	580	628.8	0
NX-0,02-SFT-no	0.026	0.068	656.6	561.8	700.6	0
NX-0,02-WFT-yes	0.027	0.074	602.4	573	743.6	0
N0-0,02-WFT-yes	0.026	0.074	644.8	555.6	718.6	0
N0-0,02-SFT-yes	0.026	0.074	644.8	555.6	718.6	0
NX-0,02-SFT-yes	0.029	0.08	517.4	572.6	829	0
NX-0,002-WFT-no	0.029	0.086	436.6	745.4	737	0
N0-0,002-WFT-no	0.027	0.086	588.6	642.4	688	0
N0-0,002-SFT-no	0.027	0.086	588.6	642.4	688	0
NX-0,002-SFT-no	0.031	0.091	328.8	807.8	782.4	0
NX-0,002-WFT-yes	0.031	0.092	336.2	834.6	748.2	0
N0-0,002-WFT-yes	0.027	0.092	590.2	667	661.8	0
N0-0,002-SFT-yes	0.027	0.092	590.2	667	661.8	0
NX-0,002-SFT-yes	0.031	0.094	301.8	903.8	713.4	0

de los expertos a la hora de sustituir a cada uno de ellos por el sistema automático propuesto en este enfoque. Los resultados que se obtuvieron con la fuente de Amazon eran muy prometedores. Para RMSE, la distancia entre los resultados de la mejor y la peor configuración era en torno a 0.0046, siendo el mejor resultado 0.0817. Para MAE, la diferencia se aproxima al 0.034, siendo el mejor resultado 0.0521. La conclusión que obtuvimos fue que, en principio, si era posible sustituir a un experto por el sistema automático, utilizando las fuentes de Amazon, sin penalizar significativamente los resultados finales.

5.1.2 Análisis de los Resultados de DBpedia

Los resultados de la fuente DBpedia no estaban tan ordenados como cabría esperar. Los algoritmos tuvieron el comportamiento esperado, siendo los mejores los más restrictivos (score 0.2). La configuración que mejores resultados obtuvo fue NX-0.2-SFT-yes (el tipo de índice a buscar X, 0.2 de score, filtrando a cada nivel de ejecución y utilizando stopwords). Los valores de las métricas fueron, para LWP = 60.1799, para LWR = 27.0033, para LCGM = 49.262, para RMSE = 0.1112 y para MAE = 0.1394. Siguiendo con el plan de resultados esperados, las configuraciones menos

restrictivas (score 0.002) obtuvieron los peores resultados. Siendo la mejor de estas configuraciones NX-0.002-SFT-no (el tipo de índice a buscar X, 0.002 de score, filtrando a cada nivel de ejecución y sin utilizar stopwords), teniendo como valores LWP = 43.0782, LWR = 37.6167, LCGM = 43.7891, RMSE = 0.1118 y MAE = 0.1645. Estos resultados, no seguían una tendencia específica, pero tenían consistencia y mantenían el orden en base a las restricciones de las configuraciones. Las configuraciones más restrictivas coincidían con los mejores resultados, a la vez que las menos restrictivas con los peores resultados, todo esto de manera global. De nuevo, la métrica que seguía perfectamente estos patrones de restricciones era LWP, ya que se alineaba perfectamente con las configuraciones y resultados esperados. Por otra parte, RMSE volvía a ser la nota más discordante. Los resultados, de nuevo, mostraban que las diferencias entre la mejor configuración (NX-0.2-SFT-yes), que además es de las más restrictivas, y la mejor configuración de las menos restrictivas (NX-0,002-SFT-no) no se distanciaban en exceso. Concretamente un 17.1 para LWP, un 5.4 para LCGM, 0.0006 para RMSE y un 0.025 para MAE. LWR era la métrica que resultaba diferente en este apartado, ya que justamente con esta configuración menos restrictiva obtenía el mejor resultado, diferenciándose de la configuración más restrictiva en un 10.61. El motivo es que esta métrica está pensada para calcular la proporción de resultados relevantes recuperados, sin tener en cuenta los fallos, es decir, la cobertura. Esta configuración recuperaba muchos elementos relevantes, junto con muchos elementos no tan relevantes.

De los resultado de la sustitución iterativa de cada experto por el sistema automático, podíamos inferir que el efecto de reemplazar a un experto por dicho sistema era insignificante, de acuerdo con los valores de las distancias vectoriales medias. Estas distancias no superaban en ningún momento el 0.4 de diferencia entre los resultados de los expertos y los resultados medios de cada sustitución.

Los resultados de DBpedia y Amazon fueron muy prometedores. Las diferencias entre el mejor y el peor de los resultados de las métricas RMSE y MAE, en ambos casos, no superaban una diferencia de 0.008 y 0.034 respectivamente. Las conclusiones que obtuvimos de estos resultados, fueron que era posible utilizar este sistema automático en pañales para sustituir, sin una penalización excesiva, a un experto.

5.2 Resultados Buscadores

El resultado del sistema automático utilizando las búsquedas de los motores web, devolvió muchos resultados. Algunos de ellos semejantes a los obtenidos en el experimento baseline. La gran cantidad de configuraciones utilizadas, junto con la peculiaridad de poder utilizar un prefiltrado para traducir los términos coloquiales a lenguaje formal, hizo que los resultados se disparasen en número. Es por ello que,

en este trabajo no mostramos todos los resultados, unicamente los más destacados. En todo caso, todos los resultados están disponibles en la web del autor[1]. Como dato adicional para entender las tablas de resultados, algunas configuraciones contienen un $*$ al comienzo. Esto significa que existen más configuraciones con los mismos resultados. Esta situación se debe a que los preprocesos y/o postprocesos no tienen una relevancia especial dentro de los datos con la configuración dada.

Los valores mínimos obtenidos, en referencia a los votos del sistema automático, eran iguales a los del baseline por el mismo motivo que lo fueron en el baseline. En cambio, los resultados máximos obtenidos diferían un poco. En el caso de Amazon sin preprocesamiento, eran menores (11900) y en el resto de casos, se superaba al baseline. Por ejemplo, en el caso de DBpedia, sin y con el preprocesado, aumentaban considerablemente a más de 18500. Lo mismo pasaba con Amazon utilizando preprocesamiento. El motivo para este aumento significativo se debía a que los resultados obtenidos de los buscadores aumentaban considerablemente los términos de las frases, por lo que, si la configuración no era muy restrictiva, se obtenían muchos más elementos.

Al igual que en el experimento baseline, los resultados están ordenados por franjas de colores para una mejor comprensión. En las tablas 5.5, 5.6, 5.7, 5.8, 5.9, 5.10, 5.11 y 5.12 se muestran los 24 mejores de cada score.

Tabla 5.5: Resultados experimento buscadores, sin preprocesado, Amazon

Configuración	M	E	C	LWP avg	LWR avg	LCGM avg	RMSE	MAE
PS-TF-NX-WFT-0,2	1920	5405	1625	80.211	27.423	62.4376	0.081	0.123
PS-LDA-NX-WFT-0,2	1920	5405	1625	80.2012	27.4034	62.4192	0.081	0.123
*PS-LDA-N0-WFT-0,2	1919	5405	1619	79.9668	27.2862	62.2565	0.081	0.123
*PS-BING-TF-TF-NX-WFT-0,2	1919	5405	1619	79.9554	27.3448	62.2228	0.082	0.123
PS-BING-TF-LDA-NX-WFT-0,2	1919	5405	1619	79.944	27.2992	62.1891	0.082	0.124
*PS-BING-LDA-LDA-NX-WFT-0,2	1919	5405	1618	79.9033	27.2927	62.1707	0.082	0.123
*GOOG-LDA-LDA-NX-WFT-0,2	1919	5405	1617	79.8626	27.2862	62.1523	0.082	0.123
PS-GOOG-TF-TF-NX-WFT-0,2	1919	5405	1617	79.856	27.2762	62.1446	0.082	0.124
PS-TF-NX-SFT-0,02	6218	5405	1626	47.1047	21.7931	41.6877	0.102	0.172
*PS-TF-N0-WFT-0,02	9625	5405	1940	45.6761	21.1733	42.1179	0.097	0.173
PS-LDA-NX-SFT-0,02	6791	5405	1640	43.3212	21.2538	39.2556	0.102	0.176
PS-LDA-N0-WFT-0,02	10633	5405	1999	41.6861	20.7251	39.7095	0.097	0.177
PS-TF-NX-WFT-0,02	10475	5405	1509	19.959	16.8726	21.4078	0.105	0.196
PS-LDA-NX-WFT-0,02	10615	5405	1536	19.7122	17.1963	21.4436	0.103	0.196
PS-GOOG-LDA-LDA-NX-SFT-0,02	10298	5405	1429	18.3801	15.7409	19.7677	0.103	0.198
PS-GOOG-LDA-TF-NX-SFT-0,02	10327	5405	1429	18.3382	15.7334	19.6997	0.103	0.198
PS-LDA-NX-SFT-0,002	11773	5405	1502	14.7065	16.442	17.6148	0.099	0.2
PS-TF-NX-SFT-0,002	11878	5405	1464	14.1434	15.8544	16.9867	0.1	0.201
PS-BING-LDA-TF-NX-SFT-0,002	11400	5405	1492	13.9735	16.0497	16.8729	0.101	0.201
PS-BING-LDA-LDA-NX-SFT-0,002	11421	5405	1489	13.9137	16.04	16.8194	0.101	0.201
BING-LDA-LDA-NX-SFT-0,002	11379	5405	1475	13.8445	15.9572	16.6942	0.101	0.201
BING-LDA-TF-NX-SFT-0,002	11355	5405	1472	13.8177	15.7661	16.6342	0.101	0.201
PS-BING-TF-TF-NX-SFT-0,002	11349	5405	1473	13.7298	15.768	16.6162	0.101	0.201
PS-BING-TF-LDA-NX-SFT-0,002	11380	5405	1473	13.682	15.8029	16.6037	0.101	0.201

[1] www.lugar_de_los_resultados.es

Tabla 5.6: Resultados experimento buscadores, sin preprocesado, DBpedia

Configuración	M	E	C	LWP avg	LWR avg	LCGM avg	RMSE	MAE
*PS-LDA-NX-WFT-0,2	1919	3731	1184	60.134	26.9407	49.2387	0.111	0.139
*PS-BING-LDA-LDA-NX-WFT-0,2	1919	3731	1184	60.1309	26.906	49.2167	0.111	0.139
*PS-LDA-N0-WFT-0,2	1919	3731	1183	60.0818	26.8886	49.1866	0.111	0.139
PS-TF-NX-WFT-0,02	6873	3731	1747	33.4732	34.6067	37.8948	0.123	0.185
PS-LDA-NX-WFT-0,02	7228	3731	1807	32.7853	35.4769	37.9633	0.121	0.185
PS-TF-NX-SFT-0,02	6966	3731	1660	32.7517	30.3414	36.6001	0.127	0.187
PS-LDA-NX-SFT-0,02	7417	3731	1738	32.4793	31.5673	37.1521	0.124	0.187
*PS-TF-N0-WFT-0,02	9949	3731	1826	29.1668	25.7575	34.7447	0.127	0.192
*PS-LDA-N0-WFT-0,02	10652	3731	1898	28.4955	26.3388	34.9253	0.124	0.192
PS-TF-NX-WFT-0,002	9434	3731	1795	21.5437	36.4226	29.7831	0.112	0.194
PS-LDA-NX-WFT-0,002	9578	3731	1825	21.5307	36.7772	30.0219	0.111	0.194
PS-BING-LDA-LDA-NX-SFT-0,02	9756	3731	1813	21.4742	34.9948	29.9362	0.112	0.195
PS-TF-NX-SFT-0,002	9436	3731	1787	21.4711	36.1886	29.6547	0.112	0.194
PS-BING-LDA-TF-NX-SFT-0,02	9752	3731	1808	21.4689	35.0093	29.8626	0.112	0.195
PS-LDA-NX-SFT-0,002	9574	3731	1818	21.4596	36.553	29.9042	0.112	0.194
PS-BING-TF-LDA-NX-WFT-0,002	9792	3731	1794	20.0567	35.4261	28.7404	0.112	0.196
PS-BING-TF-TF-NX-WFT-0,002	9761	3731	1791	20.0392	35.4386	28.7098	0.112	0.196
PS-BING-TF-LDA-NX-SFT-0,002	9764	3731	1790	20.0316	35.3496	28.6878	0.112	0.196
PS-BING-LDA-LDA-NX-WFT-0,002	9897	3731	1804	20.0193	35.4452	28.8134	0.112	0.196
PS-GOOG-TF-LDA-NX-SFT-0,002	10274	3731	1801	19.2689	34.6384	28.2125	0.112	0.197
PS-GOOG-LDA-LDA-N0-SFT-0,02	16963	3731	2043	14.4567	20.8473	25.2366	0.113	0.205
*PS-GOOG-LDA-TF-N0-SFT-0,02	17004	3731	2045	14.4223	20.7558	25.2398	0.112	0.205
PS-BING-LDA-LDA-N0-WFT-0,02	17676	3731	2080	14.1491	20.3015	25.2986	0.111	0.205
*GOOG-TF-TF-N0-SFT-0,002	18739	3731	2066	12.0159	18.9964	23.4705	0.11	0.207

Tabla 5.7: Resultados experimento buscadores, con preprocesado, Amazon

Configuración	M	E	C	LWP avg	LWR avg	LCGM avg	RMSE	MAE
*PS-TF-NX-WFT-0,2	1920	5405	1624	80.1605	27.3969	62.4007	0.082	0.123
*GOOG-TF-TF-N0-SFT-0,2	1919	5405	1619	79.9668	27.2862	62.2565	0.081	0.123
*PS-BING-TF-LDA-N0-SFT-0,2	1905	5405	1607	79.9558	27.5339	61.9295	0.081	0.123
*BING-TF-TF-NX-WFT-0,2	1905	5405	1607	79.9444	27.5404	61.8958	0.082	0.123
*GOOG-TF-TF-NX-WFT-0,2	1919	5405	1618	79.9147	27.2862	62.2044	0.081	0.123
*PS-BING-LDA-LDA-NX-WFT-0,2	1905	5405	1606	79.9037	27.5339	61.8774	0.082	0.123
PS-TF-NX-SFT-0,02	6520	5405	1582	44.0959	20.5192	39.4668	0.103	0.175
*PS-TF-N0-WFT-0,02	10124	5405	1924	42.8588	20.1621	40.1181	0.097	0.175
PS-LDA-NX-SFT-0,02	6989	5405	1614	41.9371	20.6708	38.259	0.102	0.177
*PS-LDA-N0-SFT-0,02	10929	5405	1977	40.4416	20.1673	38.8223	0.096	0.178
PS-TF-NX-WFT-0,02	10620	5405	1486	19.5878	16.1337	21.1521	0.104	0.197
PS-LDA-NX-WFT-0,02	10773	5405	1517	19.0192	16.4846	20.9494	0.103	0.197
PS-GOOG-LDA-TF-NX-SFT-0,02	10504	5405	1408	17.9771	16.1264	18.8829	0.103	0.199
PS-GOOG-LDA-LDA-NX-SFT-0,02	10480	5405	1406	17.8077	16.0957	18.7955	0.103	0.199
PS-BING-LDA-TF-NX-SFT-0,002	11478	5405	1457	13.9098	16.2239	16.2232	0.1	0.202
PS-BING-TF-LDA-NX-SFT-0,002	11299	5405	1456	13.902	16.2771	16.2875	0.101	0.202
PS-BING-LDA-LDA-NX-SFT-0,002	11488	5405	1458	13.8788	16.2728	16.2276	0.1	0.202
BING-LDA-LDA-NX-SFT-0,002	11455	5405	1450	13.8536	16.1529	16.1516	0.101	0.202
BING-LDA-TF-NX-SFT-0,002	11429	5405	1447	13.8391	16.1088	16.118	0.101	0.202
BING-TF-LDA-NX-SFT-0,002	11266	5405	1440	13.8297	16.1585	16.1849	0.101	0.202
PS-BING-TF-TF-NX-SFT-0,002	11297	5405	1447	13.8072	16.0114	16.1219	0.101	0.202
BING-TF-TF-NX-SFT-0,002	11234	5405	1440	13.7949	15.9424	16.0709	0.101	0.202
*BING-TF-TF-N0-SFT-0,002	18764	5405	1863	10.9138	12.2471	16.1815	0.092	0.204
*GOOG-TF-LDA-N0-SFT-0,002	18764	5405	1839	10.7886	11.8996	15.9648	0.092	0.204

Tabla 5.8: Resultados experimento buscadores, con preprocesado, DBpedia

Configuración	M	E	C	LWP avg	LWR avg	LCGM avg	RMSE	MAE
∗BING-TF-TF-NX-WFT-0,2	1919	3731	1184	60.134	26.9407	49.2387	0.111	0.139
∗GOOG-LDA-LDA-NX-WFT-0,2	1919	3731	1184	60.1278	26.899	49.2099	0.111	0.139
∗BING-TF-TF-N0-WFT-0,2	1905	3731	1175	60.0821	27.1267	48.8789	0.111	0.139
∗GOOG-TF-TF-N0-SFT-0,2	1919	3731	1183	60.0818	26.8886	49.1866	0.111	0.139
∗PS-BING-LDA-LDA-NX-WFT-0,2	1905	3731	1175	60.076	27.085	48.8501	0.111	0.139
∗PS-BING-LDA-LDA-NX-SFT-0,2	1905	3731	1174	60.03	27.0746	48.8268	0.111	0.139
PS-TF-NX-WFT-0,02	6986	3731	1730	32.3393	33.8432	36.9427	0.123	0.186
PS-TF-NX-SFT-0,02	7109	3731	1664	32.0309	30.5346	36.1475	0.126	0.188
PS-LDA-NX-WFT-0,02	7335	3731	1794	31.4983	34.8795	36.9254	0.121	0.186
PS-LDA-NX-SFT-0,02	7553	3731	1728	31.0504	31.3618	36.0472	0.124	0.188
∗PS-TF-N0-WFT-0,02	10275	3731	1832	28.1347	25.4489	34.0108	0.126	0.193
∗PS-LDA-N0-SFT-0,02	10966	3731	1885	26.816	25.4399	33.5614	0.124	0.194
PS-BING-LDA-TF-NX-SFT-0,02	9821	3731	1797	21.4555	35.2832	29.4519	0.112	0.195
PS-BING-LDA-LDA-NX-SFT-0,02	9864	3731	1806	21.4201	35.2976	29.4863	0.112	0.195
PS-TF-NX-WFT-0,002	9499	3731	1798	21.3947	36.308	29.6619	0.112	0.195
PS-LDA-NX-WFT-0,002	9630	3731	1820	21.3553	36.5313	29.804	0.111	0.195
PS-LDA-NX-SFT-0,002	9637	3731	1816	21.2939	36.3718	29.7207	0.112	0.195
PS-BING-LDA-TF-NX-WFT-0,002	9903	3731	1796	20.3609	35.8372	28.6318	0.112	0.196
PS-BING-LDA-TF-NX-SFT-0,002	9881	3731	1793	20.3379	35.7857	28.5904	0.112	0.196
PS-BING-LDA-LDA-NX-WFT-0,002	9936	3731	1805	20.3204	35.8552	28.6731	0.112	0.196
PS-BING-LDA-LDA-NX-SFT-0,002	9914	3731	1802	20.2973	35.8038	28.6317	0.112	0.196
BING-LDA-LDA-NX-WFT-0,002	9998	3731	1804	20.2064	35.6223	28.5452	0.112	0.196
∗PS-TF-N0-WFT-0,002	18488	3731	2054	12.0281	19.0319	23.3915	0.111	0.207
∗GOOG-LDA-TF-N0-SFT-0,002	18780	3731	2047	11.809	18.543	23.1586	0.11	0.207

Tabla 5.9: Resultados experimento buscadores sust. experto, sin prepro., Amazon

Configuración	RMSE	MAE	DISTANCIA VECTORIAL 0.1	0.2	0.3	0.4<
∗GOOG-TF-TF-N0-SFT-0,2	0.019	0.052	1134.8	426.4	357.8	0
PS-LDA-NX-WFT-0,2	0.019	0.052	1134.6	425.8	358.6	0
PS-TF-NX-WFT-0,2	0.019	0.052	1135.4	425	358.6	0
∗GOOG-LDA-LDA-NX-WFT-0,2	0.019	0.052	1132.8	426.4	359.8	0
∗BING-TF-TF-NX-WFT-0,2	0.019	0.052	1133.8	425.2	360	0
PS-TF-NX-SFT-0,02	0.022	0.075	889.6	688.2	341.2	0
∗PS-TF-N0-WFT-0,02	0.021	0.076	1014.2	594.2	310.6	0
PS-LDA-NX-SFT-0,02	0.022	0.077	908	678.2	332.8	0
∗PS-LDA-N0-SFT-0,02	0.021	0.078	1017.8	595.6	305.6	0
PS-TF-NX-WFT-0,02	0.023	0.086	903.2	738.8	277	0
PS-LDA-NX-WFT-0,02	0.022	0.086	931	731.4	256.6	0
PS-GOOG-LDA-LDA-NX-SFT-0,02	0.022	0.087	923	741.4	254.6	0
PS-GOOG-LDA-TF-NX-SFT-0,02	0.022	0.087	919.2	745.8	254	0
PS-LDA-NX-SFT-0,002	0.022	0.088	994.2	719	205.8	0
PS-TF-NX-SFT-0,002	0.022	0.089	989.6	710	219.4	0
PS-BING-LDA-TF-NX-SFT-0,002	0.022	0.089	976.4	733.4	209.2	0
PS-BING-LDA-LDA-NX-SFT-0,002	0.022	0.089	980.8	727.2	211	0
PS-LDA-NX-WFT-0,002	0.022	0.089	1022.8	700.4	195.8	0
BING-LDA-LDA-NX-SFT-0,002	0.022	0.089	980.4	723.4	215.2	0
BING-LDA-TF-NX-SFT-0,002	0.022	0.089	968.2	735.2	215.6	0
PS-BING-TF-TF-NX-SFT-0,002	0.022	0.089	1005.2	690.6	223.2	0
∗GOOG-LDA-TF-N0-WFT-0,002	0.02	0.09	1138	598.2	182.8	0
∗PS-GOOG-TF-TF-N0-WFT-0,002	0.02	0.09	1138	598.6	182.4	0
∗GOOG-TF-TF-N0-WFT-0,002	0.02	0.09	1140.2	593.4	185.4	0

Tabla 5.10: Resultados experimento buscadores sust. experto, sin prepro., DBpedia

Configuración	RMSE	MAE	DISTANCIA VECTORIAL 0.1	0.2	0.3	0.4<
∗PS-BING-TF-LDA-N0-SFT-0,2	0.023	0.062	737	593.2	588.8	0
GOOG-LDA-TF-N0-SFT-0,2	0.023	0.062	736.8	593.2	589	0
∗PS-BING-LDA-TF-NX-WFT-0,2	0.023	0.062	737	592.6	589.4	0
∗GOOG-TF-LDA-N0-SFT-0,2	0.023	0.062	737	592.2	589.8	0
GOOG-LDA-TF-N0-WFT-0,2	0.023	0.062	736.8	593.2	589	0
GOOG-TF-LDA-N0-WFT-0,2	0.023	0.062	737	592.2	589.8	0
GOOG-LDA-TF-NX-SFT-0,2	0.023	0.062	736.8	593.2	589	0
∗BING-LDA-TF-NX-WFT-0,2	0.023	0.062	737	592.6	589.4	0
∗PS-GOOG-LDA-TF-N0-SFT-0,002	0.026	0.099	677.2	646.6	595.2	0
∗PS-TF-N0-WFT-0,002	0.026	0.099	655	665.8	598.2	0
∗BING-LDA-TF-N0-SFT-0,02	0.026	0.098	653.6	668.2	597.2	0
∗PS-LDA-N0-WFT-0,002	0.026	0.099	654.8	665	599.2	0
∗BING-TF-TF-N0-WFT-0,02	0.026	0.098	655.2	665.6	598.2	0
∗BING-LDA-LDA-N0-SFT-0,02	0.026	0.098	651.4	669.4	598.2	0
BING-TF-LDA-N0-SFT-0,02	0.026	0.098	655.2	664.4	599.4	0
PS-BING-LDA-TF-N0-SFT-0,02	0.026	0.098	648.8	670.4	599.8	0
PS-BING-LDA-LDA-N0-WFT-0,02	0.026	0.098	648.6	670.6	599.8	0
PS-BING-TF-LDA-N0-SFT-0,02	0.026	0.098	651.4	667	600.6	0
PS-BING-TF-TF-N0-SFT-0,02	0.026	0.098	647.6	669.8	601.6	0
GOOG-TF-LDA-NX-WFT-0,002	0.028	0.099	475	845.2	598.8	0
GOOG-TF-TF-NX-WFT-0,002	0.028	0.099	473.4	846	599.6	0
PS-GOOG-TF-LDA-NX-WFT-0,002	0.028	0.099	472.2	848.8	598	0
GOOG-LDA-LDA-NX-WFT-0,002	0.028	0.099	476.8	845.6	596.6	0
GOOG-LDA-TF-NX-WFT-0,002	0.028	0.099	475.8	846.6	596.6	0

5.2.1 Análisis de los Resultados de Amazon

Para los resultados de Amazon contra los expertos con y sin preprocesamiento, mostrados en las tablas 5.5 y 5.7, analizando LWP que se basa en evaluar la "precision" de los resultados con el objetivo de proveer la probabilidad de relevancia de las clases obtenidas incluyendo una pequeña penalización, los resultados de la mejor configuración eran un poco mejores que los resultados obtenidos con la mejor configuración del baseline. Con las frases sin preprocesar, se mejoraba la detección de las temáticas en un 0,21. La diferencia era bastante baja, pero los votos que efectuaba la maquina eran 2 menos, aumentando en 5 las coincidencias con los expertos. Con las frases preprocesadas, se mejoraba la detección de la temática en un 0,17. En este caso, los votos que realizaba la maquina también se reducían en 2 y las coincidencias se aumentaban en 4.

Al mismo tiempo, los resultados de LWR para esta configuración también mejoraban. Concretamente en un 0,05 para los no preprocesados y en un 0,03 para los preprocesados. Esta métrica se centra en la evaluación del "recall" de los resultados, con el objetivo de calcular la proporción de documentos relevantes recuperados comparados con el total de documentos que son relevantes según la etiquetación de los expertos. Es decir, la cobertura del sistema automático. Además, se añade una

Tabla 5.11: Resultados experimento buscadores sust. experto, con prepro., Amazon

Configuración	RMSE	MAE	DISTANCIA VECTORIAL 0.1	0.2	0.3	0.4<
PS-LDA-NX-WFT-0,2	0.019	0.052	1135.2	425.8	358	0
∗BING-LDA-TF-N0-SFT-0,2	0.019	0.052	1134.8	426.4	357.8	0
∗GOOG-TF-LDA-NX-WFT-0,2	0.019	0.052	1133.8	426.4	358.8	0
BING-TF-LDA-NX-WFT-0,2	0.019	0.052	1134.4	425.8	358.8	0
∗PS-BING-TF-LDA-NX-WFT-0,2	0.019	0.052	1133.8	425.8	359.4	0
∗BING-LDA-LDA-NX-WFT-0,2	0.019	0.052	1133.6	426.4	359	0
PS-TF-NX-SFT-0,02	0.022	0.077	890.6	676	352.4	0
∗PS-TF-N0-SFT-0,02	0.021	0.077	1019	583.4	316.6	0
PS-LDA-NX-SFT-0,02	0.022	0.078	908.2	680.8	330	0
∗PS-LDA-N0-WFT-0,02	0.021	0.078	1030.6	592.2	296.2	0
PS-TF-NX-WFT-0,02	0.023	0.087	920.6	722.4	276	0
PS-LDA-NX-WFT-0,02	0.022	0.087	942.8	717	259.2	0
PS-GOOG-TF-LDA-NX-SFT-0,02	0.023	0.087	909.2	751.4	258.4	0
PS-GOOG-LDA-TF-NX-SFT-0,02	0.022	0.087	938.6	722.4	258	0
PS-LDA-NX-SFT-0,002	0.022	0.089	993.8	720.4	204.8	0
PS-TF-NX-SFT-0,002	0.022	0.089	997	702.2	219.8	0
PS-LDA-NX-WFT-0,002	0.022	0.089	1023.4	699.6	196	0
PS-BING-LDA-TF-NX-SFT-0,002	0.022	0.089	998.8	706.6	213.6	0
PS-BING-TF-LDA-NX-SFT-0,002	0.022	0.089	990.4	716.6	212	0
BING-TF-LDA-NX-SFT-0,002	0.022	0.089	977.4	728.4	213.2	0
BING-LDA-TF-NX-SFT-0,002	0.022	0.089	992.2	711	215.8	0
PS-BING-LDA-LDA-NX-SFT-0,002	0.022	0.089	1008.8	695.4	214.8	0
∗PS-GOOG-LDA-TF-N0-WFT-0,002	0.02	0.09	1140	598.6	180.4	0
∗PS-GOOG-LDA-LDA-N0-SFT-0,002	0.02	0.09	1141.2	598.8	179	0

pequeña penalización.

Así mismo, la métrica LCGM que es la media geométrica de la probabilidad de que una clase encontrada sea correcta y la probabilidad de que una clase de la frase sea encontrada, para las frases no preprocesadas también mejora en 0.22, siendo un 0,18 para las preprocesadas.

Los resultados de las métricas RMSE y MAE, las cuales sirven para calcular si un sistema de recomendaciones es óptimo teniendo en cuenta los aciertos y fallos y, siendo RMSE la métrica que penaliza más los grandes errores, han sido semejantes a los obtenidos con la mejor configuración del baseline. La mejora en estas métricas es realmente pequeña, pero sirve para verificar que la configuración PS-TF-NX-WFT-0,2 mejora la detección de la temática y obtiene una ventaja a la hora de detectar y filtrar los topics de las frases, tal y como lo hace un experto.

En la parte de sustituir a un experto, en las tablas 5.9 y 5.11 se puede ver que, la diferencia entre los resultados de los expertos y los resultados medios de la sustitución no difieren en exceso. Estos no superan en ningún momento una distancia mayor a 0,4, equivalente a un 30 % de no acierto completo, tanto para el sistema que utiliza preprocesamiento como para el que no. El análisis de estos resultados nos muestra que el consenso se seguía manteniendo, tal y como si se hubiera metido

Tabla 5.12: Resultados experimento buscadores sust. experto, con prepro., DBpedia

Configuración	RMSE	MAE	DISTANCIA VECTORIAL 0.1	0.2	0.3	0.4<
∗BING-LDA-LDA-NX-SFT-0,2	0.023	0.062	737	593.2	588.8	0
∗BING-TF-TF-NX-SFT-0,2	0.023	0.062	737	593.2	588.8	0
∗PS-GOOG-TF-TF-NX-WFT-0,2	0.023	0.062	736.8	593.2	589	0
∗PS-TF-NX-WFT-0,2	0.023	0.062	736.8	592.6	589.6	0
∗PS-BING-TF-LDA-N0-SFT-0,2	0.023	0.062	736.8	593.2	589	0
∗PS-BING-LDA-LDA-NX-WFT-0,2	0.023	0.062	736.4	593	589.6	0
∗PS-TF-N0-SFT-0,02	0.029	0.094	469.8	702.4	746.8	0
∗PS-LDA-N0-SFT-0,02	0.028	0.094	468.6	720.6	729.8	0
PS-TF-NX-SFT-0,02	0.03	0.094	424.6	737	757.4	0
PS-LDA-NX-SFT-0,02	0.029	0.094	422.6	758	738.4	0
PS-TF-NX-WFT-0,02	0.03	0.094	391.2	788.2	739.6	0
PS-LDA-NX-WFT-0,02	0.029	0.095	383.8	808.8	726.4	0
∗BING-LDA-TF-N0-SFT-0,02	0.026	0.099	667.6	656.2	595.2	0
∗BING-LDA-LDA-N0-SFT-0,02	0.026	0.099	667.4	655.4	596.2	0
PS-LDA-NX-SFT-0,002	0.028	0.099	448	857.6	613.4	0
PS-LDA-NX-SFT-0,002	0.028	0.099	448	857.6	613.4	0
PS-TF-NX-SFT-0,002	0.028	0.099	440.2	862.4	616.4	0
∗PS-LDA-N0-WFT-0,002	0.026	0.099	659.2	661.6	598.2	0
∗PS-TF-N0-SFT-0,002	0.026	0.099	658	662.8	598.2	0
∗PS-GOOG-TF-TF-N0-SFT-0,002	0.026	0.099	679.4	645.4	594.2	0
PS-GOOG-TF-TF-NX-SFT-0,002	0.028	0.099	481.8	837.4	599.8	0
GOOG-LDA-TF-NX-SFT-0,002	0.028	0.099	490.4	825.8	602.8	0
BING-LDA-LDA-NX-WFT-0,002	0.028	0.099	476.6	842.4	600	0
PS-BING-TF-LDA-NX-WFT-0,002	0.028	0.099	422.2	897.2	599.6	0

a un nuevo experto. En base a esto, el sistema lo dábamos por bueno para Amazon.

5.2.2 Análisis de los Resultados de DBpedia

Para los resultados de DBpedia contra los expertos, con y sin preprocesamiento, mostrados en las tablas 5.6 y 5.8, analizando LWP se veía que los resultados de la mejor configuración eran casi semejantes, aunque no igualaban ni mejoraban a los resultados del baseline. Con las frases sin preprocesamiento, los resultados obtenidos empeoraban en un 0,04. La diferencia era mínima y los votos realizados iguales, pero los votos que coincidían disminuyeron en 1. Con las frases preprocesadas, la diferencia era la misma, de 0,04, y la diferencia con los votos emitidos y las coincidencias era igual que en el anterior caso.

Al mismo tiempo, los resultados de LWR para esta configuración tampoco mejoraban los del baseline. La diferencia entre ellos era de 0,06 tanto utilizando el preprocesamiento como sin el. Así mismo, la métrica LCGM para las frases no preprocesadas y preprocesadas no alcanzaba a las del baseline, quedándose a 0.02.

Los resultados de las métricas RMSE y MAE fueron menores, pero muy semejantes a los obtenidos con la mejor configuración del baseline. La diferencia en estas

métricas era realmente pequeña, pero servia para verificar que tanto la configuración PS-TF-NX-WFT-0,2 y PS-LDA-NX-WFT-0,2 en el caso de las frases no preprocesadas y PS-TF-NX-WFT-0,2, BING-TF-TF-NX-WFT-0,2, PS-LDA-NX-WFT-0,2 y BING-TF-TF-NX-SFT-0,2 en el caso de las frases preprocesadas eran una posibilidad a tener en cuenta para la detección de la temática y filtrar los topics de las frases, tal y como lo haría un experto.

En la parte de sustituir a un experto por el sistema automático, al igual que ocurrió con los resultados de Amazon, los de DBpedia no sufrían absolutamente ninguna penalización por desviar el contexto general ni por romper el consenso que mantenían los expertos entre sí. En las tablas 5.10 y 5.12 se puede ver que, la diferencia entre los resultados de los expertos y los resultados medios de la sustitución no difieren en exceso. Estos no superan en ningún momento una distancia mayor a 0,4 tanto para el sistema que tiene preprocesa las frases como para el que no. Los resultados muestran que la sustitución de un experto por nuestro sistema sería como meter un nuevo experto. Debido a esto, en principio lo damos por bueno también para DBpedia.

5.2.3 **Conclusiones de los Resultados**

A modo de análisis y conclusión general de los resultados, hay que mencionar que este experimento estaba planteado para verificar si los resultados obtenidos por los buscadores de contenido online, como Google y Bing, servían para mejorar la capacidad de nuestro sistema y poder sustituir a un experto.

El gran número de configuraciones utilizadas con preprocesado de las frases, extracción de topics, filtrado de resultados, selección de diferentes niveles de búsqueda y diferentes filtrados para los resultados obtenidos de las búsquedas, nos han mostrado que los mejores resultados se han obtenido con técnicas de extracción de topics sin la necesidad de utilizar los buscadores. Esto se debe a que los resultados de los buscadores, aunque son muy útiles y es necesario seguir investigando en esa área, utilizando otras técnicas diferentes de extracción de topics u otros buscadores, introducen demasiado ruido en forma de contenidos que no siempre tienen que ver con el objetivo o contexto de la frase a analizar. Un elemento importante a tener en cuenta para poder continuar una investigación con buscadores es que las métricas muestran que este planteamiento es válido, pero que es necesario analizar e investigar buscadores más específicos, o aplicar otras técnicas en el filtrado de sus resultados. La búsqueda de un motor de búsqueda web es bastante importante ya que, no todos los buscadores devuelven las mismas entradas para una misma frase ni todas las fechas son las propicias para obtener resultados de ellos. Por ejemplo, teniendo en cuenta que se han estado utilizando las mismas frase, con los mismos filtros y preprocesado, el buscador Bing ha obtenido mejores resultados de las métri-

cas que el buscador Google. También, y en referencia a la fecha, si se está hablando de una noticia pasada, los resultados pueden variar según vayan apareciendo nuevos contenidos con nuevos puntos de vista sobre esa noticia. Estos enfoques pueden ser simplemente divagaciones, o nuevos puntos de vista que se aventuren a vincularlas con otras temáticas. Pero además, estos resultados son bastante semejantes para ambos datasets. En ambos casos, los mejores resultados se han obtenido sin el uso de los navegadores. Los resultados que igualan a los del baseline comprenden la utilización de ambos buscadores con diferentes filtrados y solo unas pocas configuraciones que no los utilizan. La diferencia en esta semejanza la pone DBpedia con el preprocesamiento. En estos resultados son en los únicos que entran configuraciones con buscadores dentro de las mejores, concretamente el del buscador Bing.

En el caso de sustituir a un experto, la variación ha sido mínima. Esta situación es perfecta debido a que, con estas configuraciones, el consenso que había dentro de los expertos no varía en exceso, simplemente es como si se hubiera introducido un nuevo experto dentro del sistema.

5.3 Resultados Deep Learning

Los resultados del experimento con técnicas de deeplearning nos mostraron que los motivos por los cuales realizamos estos experimentos, eran acertados. Muchos de los resultados siguieron siendo muy parecidos a los del experimento baseline. Esto era una buena señal para proseguir con este enfoque, sabiendo que era viable y que únicamente es encontrar el sistema de pre y postfiltrado adecuado para mejorarlo. Al igual que en el experimento con buscadores, para este se utilizaron preprocesamientos específicos que ya hemos descrito en la sección 4.3. En definitiva, este experimento es otra pieza viable, pensándolo como un enfoque, dentro del sistema de detección de temáticas en frases cortas.

En referencia a los resultados, los votos, tanto para Amazon como para DBpedia e independientemente del preprocesado y con las configuraciones más permisivas, se disparan hasta más de 17000. Estos votos superan a los máximos del baseline, pero en el caso de DBpedia, comparados contra los del experimento de buscadores, disminuyen notablemente. Por otra parte, los votos mínimos sí que tienen una notable disminución. En el baseline están en 1919 y en este experimento bajan hasta 1905, teniendo unos resultados semejantes.

Por último y al igual que en el experimento baseline, los resultados están ordenados por franjas de colores para una mejor comprensión y se muestran en las tablas 5.13, 5.14, 5.15, 5.16, 5.17, 5.18, 5.19 y 5.20.

Tabla 5.13: Resultados experimento deeplearning, sin preprocesado, Amazon

Configuración	M	E	C	LWP avg	LWR avg	LCGM avg	RMSE	MAE
WIKIPEDIA-N0-SFT-0,2	1905	5405	1607	79.9558	27.5339	61.9295	0.081	0.123
GOOG_NEWS-NX-SFT-0,2	1905	5405	1607	79.9558	27.5339	61.9295	0.081	0.123
WIKIPEDIA-N0-WFT-0,2	1905	5405	1607	79.9558	27.5339	61.9295	0.081	0.123
GOOG_NEWS-N0-SFT-0,2	1905	5405	1607	79.9558	27.5339	61.9295	0.081	0.123
GOOG_NEWS-N0-WFT-0,2	1905	5405	1607	79.9558	27.5339	61.9295	0.081	0.123
WIKIPEDIA-NX-SFT-0,2	1905	5405	1607	79.9558	27.5339	61.9295	0.081	0.123
GOOG_NEWS-NX-WFT-0,2	1918	5405	1609	79.8639	27.5578	61.8801	0.082	0.124
WIKIPEDIA-NX-WFT-0,2	1907	5405	1605	79.841	27.6101	61.8745	0.082	0.124
WIKIPEDIA-NX-SFT-0,02	4481	5405	1543	58.2221	24.5612	46.9995	0.097	0.157
WIKIPEDIA-N0-WFT-0,02	6574	5405	1704	57.1585	24.1712	47.0296	0.093	0.158
WIKIPEDIA-N0-SFT-0,02	6574	5405	1704	57.1585	24.1712	47.0296	0.093	0.158
GOOG_NEWS-NX-SFT-0,02	4917	5405	1592	53.7463	24.0351	44.7217	0.101	0.164
GOOG_NEWS-N0-SFT-0,02	7941	5405	1862	52.528	23.6556	44.987	0.096	0.165
GOOG_NEWS-N0-WFT-0,02	7941	5405	1862	52.528	23.6556	44.987	0.096	0.165
WIKIPEDIA-NX-WFT-0,02	8100	5405	1535	32.9345	20.9245	30.35	0.104	0.185
GOOG_NEWS-NX-WFT-0,02	8960	5405	1643	26.7726	20.9225	27.039	0.105	0.19
WIKIPEDIA-NX-SFT-0,002	10305	5405	1536	20.3257	19.2964	21.6533	0.101	0.196
GOOG_NEWS-NX-SFT-0,002	10524	5405	1611	18.6439	19.9107	20.9119	0.101	0.196
GOOG_NEWS-NX-WFT-0,002	11180	5405	1615	16.7094	20.0634	19.3754	0.098	0.198
WIKIPEDIA-NX-WFT-0,002	11310	5405	1515	16.435	18.8658	18.5494	0.098	0.199
WIKIPEDIA-N0-WFT-0,002	16989	5405	1965	16.205	15.4162	20.6282	0.094	0.199
WIKIPEDIA-N0-SFT-0,002	16989	5405	1965	16.205	15.4162	20.6282	0.094	0.199
GOOG_NEWS-N0-WFT-0,002	17698	5405	2032	14.5441	15.0764	19.6013	0.093	0.201
GOOG_NEWS-N0-SFT-0,002	17698	5405	2032	14.5441	15.0764	19.6013	0.093	0.201

Tabla 5.14: Resultados experimento deeplearning, sin preprocesado, DBpedia

Configuración	M	E	C	LWP avg	LWR avg	LCGM avg	RMSE	MAE
GOOG_NEWS-NX-WFT-0,2	1905	3731	1180	60.3288	27.2596	49.0316	0.111	0.14
GOOG_NEWS-NX-SFT-0,2	1905	3731	1176	60.1327	27.1527	48.9157	0.111	0.139
GOOG_NEWS-N0-SFT-0,2	1905	3731	1176	60.1327	27.1527	48.9157	0.111	0.139
GOOG_NEWS-N0-WFT-0,2	1905	3731	1176	60.1327	27.1527	48.9157	0.111	0.139
WIKIPEDIA-N0-SFT-0,2	1905	3731	1175	60.0806	27.1006	48.8636	0.111	0.139
WIKIPEDIA-NX-WFT-0,2	1905	3731	1175	60.0806	27.1006	48.8636	0.111	0.139
WIKIPEDIA-N0-WFT-0,2	1905	3731	1175	60.0806	27.1006	48.8636	0.111	0.139
WIKIPEDIA-NX-SFT-0,2	1905	3731	1175	60.0806	27.1006	48.8636	0.111	0.139
WIKIPEDIA-NX-WFT-0,02	4148	3731	1577	46.3648	36.6146	44.2754	0.121	0.169
WIKIPEDIA-NX-SFT-0,02	3887	3731	1353	42.7977	28.9274	39.9217	0.135	0.176
GOOG_NEWS-NX-WFT-0,02	4831	3731	1677	42.7366	38.7117	43.2316	0.12	0.174
GOOG_NEWS-NX-SFT-0,02	4719	3731	1538	40.8297	33.456	40.6215	0.129	0.178
WIKIPEDIA-N0-WFT-0,02	5496	3731	1471	40.5944	27.9473	39.3208	0.132	0.179
WIKIPEDIA-N0-SFT-0,02	5496	3731	1471	40.5944	27.9473	39.3208	0.132	0.179
GOOG_NEWS-N0-SFT-0,02	7368	3731	1679	36.5524	29.3354	38.4966	0.128	0.184
GOOG_NEWS-N0-WFT-0,02	7368	3731	1679	36.5524	29.3354	38.4966	0.128	0.184
WIKIPEDIA-NX-SFT-0,002	6267	3731	1722	31.5987	41.5125	35.7643	0.114	0.182
WIKIPEDIA-NX-WFT-0,002	6304	3731	1713	30.9525	41.709	35.1873	0.113	0.183
GOOG_NEWS-NX-SFT-0,002	6922	3731	1728	28.3875	40.7355	33.6625	0.114	0.186
GOOG_NEWS-NX-WFT-0,002	6907	3731	1722	28.0384	40.9217	33.3631	0.114	0.186
WIKIPEDIA-N0-WFT-0,002	16664	3731	2097	15.9322	22.5318	26.5316	0.112	0.204
WIKIPEDIA-N0-SFT-0,002	16664	3731	2097	15.9322	22.5318	26.5316	0.112	0.204
GOOG_NEWS-N0-WFT-0,002	17392	3731	2059	14.1486	20.6617	24.8236	0.112	0.206
GOOG_NEWS-N0-SFT-0,002	17392	3731	2059	14.1486	20.6617	24.8236	0.112	0.206

Tabla 5.15: Resultados experimento deeplearning, con preprocesado, Amazon

Configuración	M	E	C	LWP avg	LWR avg	LCGM avg	RMSE	MAE
GOOG_NEWS-N0-SFT-0,2	1905	5405	1607	79.9558	27.5339	61.9295	0.081	0.123
GOOG_NEWS-NX-SFT-0,2	1905	5405	1607	79.9558	27.5339	61.9295	0.081	0.123
WIKIPEDIA-NX-SFT-0,2	1905	5405	1607	79.9558	27.5339	61.9295	0.081	0.123
WIKIPEDIA-N0-SFT-0,2	1905	5405	1607	79.9558	27.5339	61.9295	0.081	0.123
GOOG_NEWS-N0-WFT-0,2	1905	5405	1607	79.9558	27.5339	61.9295	0.081	0.123
WIKIPEDIA-N0-WFT-0,2	1905	5405	1607	79.9558	27.5339	61.9295	0.081	0.123
WIKIPEDIA-NX-WFT-0,2	1907	5405	1607	79.9452	27.6622	61.9787	0.082	0.123
GOOG_NEWS-NX-WFT-0,2	1914	5405	1609	79.9334	27.5614	61.9141	0.082	0.124
GOOG_NEWS-NX-SFT-0,02	4888	5405	1603	54.8513	24.1433	45.5208	0.1	0.163
WIKIPEDIA-NX-SFT-0,02	4785	5405	1489	54.558	24.1735	43.9406	0.099	0.161
WIKIPEDIA-N0-WFT-0,02	7154	5405	1653	53.4062	23.5265	43.9032	0.095	0.162
WIKIPEDIA-N0-SFT-0,02	7154	5405	1653	53.4062	23.5265	43.9032	0.095	0.162
GOOG_NEWS-N0-SFT-0,02	7659	5405	1822	53.3262	23.4767	45.4913	0.095	0.165
GOOG_NEWS-N0-WFT-0,02	7659	5405	1822	53.3262	23.4767	45.4913	0.095	0.165
WIKIPEDIA-NX-WFT-0,02	8383	5405	1525	31.3621	20.8237	29.3929	0.105	0.186
GOOG_NEWS-NX-WFT-0,02	9027	5405	1632	27.3242	20.7498	27.2753	0.105	0.19
WIKIPEDIA-NX-SFT-0,002	10335	5405	1547	20.6702	19.0272	22.2057	0.1	0.195
GOOG_NEWS-NX-SFT-0,002	10634	5405	1606	19.1577	19.4002	21.4063	0.101	0.196
WIKIPEDIA-N0-SFT-0,002	17033	5405	1973	16.6504	15.1203	21.1838	0.093	0.199
WIKIPEDIA-N0-WFT-0,002	17033	5405	1973	16.6504	15.1203	21.1838	0.093	0.199
GOOG_NEWS-NX-WFT-0,002	11364	5405	1585	16.5101	19.24	19.1268	0.098	0.198
WIKIPEDIA-NX-WFT-0,002	11267	5405	1495	16.3569	18.456	18.4466	0.099	0.199
GOOG_NEWS-N0-WFT-0,002	17544	5405	2019	15.2682	14.843	20.2089	0.093	0.2
GOOG_NEWS-N0-SFT-0,002	17544	5405	2019	15.2682	14.843	20.2089	0.093	0.2

Tabla 5.16: Resultados experimento deeplearning, con preprocesado, DBpedia

Configuración	M	E	C	LWP avg	LWR avg	LCGM avg	RMSE	MAE
GOOG_NEWS-NX-WFT-0,2	1905	3731	1180	60.3288	27.2596	49.0316	0.111	0.139
GOOG_NEWS-N0-SFT-0,2	1905	3731	1176	60.1327	27.1527	48.9157	0.111	0.171
GOOG_NEWS-NX-SFT-0,2	1905	3731	1176	60.1327	27.1527	48.9157	0.111	0.139
GOOG_NEWS-N0-WFT-0,2	1905	3731	1176	60.1327	27.1527	48.9157	0.111	0.139
WIKIPEDIA-NX-SFT-0,2	1905	3731	1175	60.0806	27.1006	48.8636	0.111	0.181
WIKIPEDIA-N0-SFT-0,2	1905	3731	1175	60.0806	27.1006	48.8636	0.111	0.14
WIKIPEDIA-N0-WFT-0,2	1905	3731	1175	60.0806	27.1006	48.8636	0.111	0.205
WIKIPEDIA-NX-WFT-0,2	1905	3731	1174	60.0315	27.0759	48.8335	0.111	0.204
WIKIPEDIA-NX-WFT-0,02	4486	3731	1603	45.4531	36.177	44.1313	0.12	0.185
GOOG_NEWS-NX-WFT-0,02	4761	3731	1641	43.3647	36.9411	43.3821	0.12	0.204
WIKIPEDIA-NX-SFT-0,02	4266	3731	1391	42.496	28.862	40.3544	0.132	0.139
GOOG_NEWS-NX-SFT-0,02	4599	3731	1504	41.6804	31.8986	40.9353	0.129	0.173
WIKIPEDIA-N0-WFT-0,02	5551	3731	1472	40.5394	27.9219	39.5886	0.131	0.139
WIKIPEDIA-N0-SFT-0,02	5551	3731	1472	40.5394	27.9219	39.5886	0.131	0.181
GOOG_NEWS-N0-SFT-0,02	6601	3731	1622	38.3445	29.0464	39.3525	0.128	0.179
GOOG_NEWS-N0-WFT-0,02	6601	3731	1622	38.3445	29.0464	39.3525	0.128	0.177
WIKIPEDIA-NX-SFT-0,002	6742	3731	1717	30.1177	40.0208	34.8019	0.114	0.139
WIKIPEDIA-NX-WFT-0,002	6778	3731	1718	29.5752	40.5252	34.3496	0.113	0.139
GOOG_NEWS-NX-SFT-0,002	7189	3731	1733	28.0278	39.8994	33.5124	0.114	0.187
GOOG_NEWS-NX-WFT-0,002	7170	3731	1728	27.6769	40.2671	33.2061	0.113	0.184
WIKIPEDIA-N0-SFT-0,002	16598	3731	2068	15.8543	22.08	26.262	0.113	0.176
WIKIPEDIA-N0-WFT-0,002	16598	3731	2068	15.8543	22.08	26.262	0.113	0.187
GOOG_NEWS-N0-WFT-0,002	17203	3731	2061	14.6862	20.7816	25.2413	0.112	0.179
GOOG_NEWS-N0-SFT-0,002	17203	3731	2061	14.6862	20.7816	25.2413	0.112	0.205

Tabla 5.17: Resultados experimento deeplearning sust. experto, sin preprocesado, Amazon

Configuración	RMSE	MAE	DISTANCIA VECTORIAL			
			0.1	0.2	0.3	0.4<
WIKIPEDIA-NX-SFT-0,2	0.019	0.052	1134.8	426.4	357.8	0
GOOG_NEWS-NX-SFT-0,2	0.019	0.052	1134.8	426.4	357.8	0
GOOG_NEWS-N0-SFT-0,2	0.019	0.052	1134.8	426.4	357.8	0
WIKIPEDIA-N0-SFT-0,2	0.019	0.052	1134.8	426.4	357.8	0
GOOG_NEWS-N0-WFT-0,2	0.019	0.052	1134.8	426.4	357.8	0
WIKIPEDIA-N0-WFT-0,2	0.019	0.052	1134.8	426.4	357.8	0
WIKIPEDIA-NX-WFT-0,2	0.019	0.052	1133.2	425.2	360.6	0
GOOG_NEWS-NX-WFT-0,2	0.019	0.053	1127.2	427.2	364.6	0
WIKIPEDIA-NX-SFT-0,02	0.021	0.068	967.6	594	357.4	0
WIKIPEDIA-N0-SFT-0,02	0.021	0.069	1043	530	346	0
WIKIPEDIA-N0-WFT-0,02	0.021	0.069	1043	530	346	0
GOOG_NEWS-NX-SFT-0,02	0.022	0.072	872.2	690.8	356	0
GOOG_NEWS-N0-WFT-0,02	0.021	0.072	1036.4	545.8	336.8	0
GOOG_NEWS-N0-SFT-0,02	0.021	0.072	1036.4	545.8	336.8	0
WIKIPEDIA-NX-WFT-0,02	0.023	0.081	912.6	687.6	318.8	0
GOOG_NEWS-NX-WFT-0,02	0.023	0.084	850.6	795.2	273.2	0
WIKIPEDIA-NX-SFT-0,002	0.022	0.086	931.2	771.8	216	0
GOOG_NEWS-NX-SFT-0,002	0.022	0.087	913.8	804.4	200.8	0
GOOG_NEWS-NX-WFT-0,002	0.022	0.088	975.4	774.4	169.2	0
WIKIPEDIA-NX-WFT-0,002	0.022	0.088	1003.6	733.6	181.8	0
WIKIPEDIA-N0-SFT-0,002	0.021	0.088	1093.2	642.4	183.4	0
WIKIPEDIA-N0-WFT-0,002	0.021	0.088	1093.2	642.4	183.4	0
GOOG_NEWS-N0-SFT-0,002	0.021	0.089	1120.8	615.2	183	0
GOOG_NEWS-N0-WFT-0,002	0.021	0.089	1120.8	615.2	183	0

Tabla 5.18: Resultados experimento deeplearning sust. experto, sin preprocesado, DBpedia

Configuración	RMSE	MAE	DISTANCIA VECTORIAL			
			0.1	0.2	0.3	0.4<
WIKIPEDIA-NX-SFT-0,2	0.023	0.062	737	592.8	589.2	0
WIKIPEDIA-N0-WFT-0,2	0.023	0.062	737	592.8	589.2	0
WIKIPEDIA-N0-SFT-0,2	0.023	0.062	737	592.8	589.2	0
WIKIPEDIA-NX-WFT-0,2	0.023	0.062	736.8	592.2	590	0
GOOG_NEWS-N0-SFT-0,2	0.023	0.062	736.8	592	590.2	0
GOOG_NEWS-N0-WFT-0,2	0.023	0.062	736.8	592	590.2	0
GOOG_NEWS-NX-SFT-0,2	0.023	0.062	736.8	592	590.2	0
GOOG_NEWS-NX-WFT-0,2	0.023	0.062	734.2	588.6	596.2	0
WIKIPEDIA-N0-WFT-0,02	0.031	0.089	454	577.2	887.8	0
WIKIPEDIA-NX-SFT-0,02	0.031	0.089	396	590.6	932.4	0
WIKIPEDIA-N0-SFT-0,02	0.031	0.089	454	577.2	887.8	0
WIKIPEDIA-NX-WFT-0,02	0.031	0.09	357.4	711.8	849.8	0
GOOG_NEWS-N0-SFT-0,02	0.03	0.091	453	631.6	834.4	0
GOOG_NEWS-NX-SFT-0,02	0.031	0.091	363.2	678.8	877	0
GOOG_NEWS-N0-WFT-0,02	0.03	0.091	453	631.6	834.4	0
GOOG_NEWS-NX-WFT-0,02	0.031	0.093	323.8	765.2	830	0
WIKIPEDIA-N0-WFT-0,002	0.026	0.098	597	709.8	612.2	0
WIKIPEDIA-N0-SFT-0,002	0.026	0.098	597	709.8	612.2	0
WIKIPEDIA-NX-SFT-0,002	0.031	0.098	194.4	1019	705.6	0
GOOG_NEWS-N0-WFT-0,002	0.026	0.099	623.4	689.4	606.2	0
GOOG_NEWS-N0-SFT-0,002	0.026	0.099	623.4	689.4	606.2	0
GOOG_NEWS-NX-SFT-0,002	0.03	0.099	223.4	1013.6	682	0
WIKIPEDIA-NX-WFT-0,002	0.031	0.099	178.4	1044	696.6	0
GOOG_NEWS-NX-WFT-0,002	0.03	0.099	207.6	1034.8	676.6	0

Tabla 5.19: Resultados experimento deeplearning sust. experto, con preprocesado, Amazon

Configuración	RMSE	MAE	DISTANCIA VECTORIAL			
			0.1	0.2	0.3	0.4<
GOOG_NEWS-N0-WFT-0,2	0.019	0.052	1134.8	426.4	357.8	0
WIKIPEDIA-N0-WFT-0,2	0.019	0.052	1134.8	426.4	357.8	0
WIKIPEDIA-N0-SFT-0,2	0.019	0.052	1134.8	426.4	357.8	0
GOOG_NEWS-N0-SFT-0,2	0.019	0.052	1134.8	426.4	357.8	0
WIKIPEDIA-NX-SFT-0,2	0.019	0.052	1134.8	426.4	357.8	0
GOOG_NEWS-NX-SFT-0,2	0.019	0.052	1134.8	426.4	357.8	0
WIKIPEDIA-NX-WFT-0,2	0.019	0.052	1134.6	425.8	358.6	0
GOOG_NEWS-NX-WFT-0,2	0.019	0.052	1131.4	428.4	359.2	0
WIKIPEDIA-NX-SFT-0,02	0.022	0.07	935.6	614.4	369	0
WIKIPEDIA-N0-WFT-0,02	0.021	0.071	1031.4	534.2	353.4	0
WIKIPEDIA-N0-SFT-0,02	0.021	0.071	1031.4	534.2	353.4	0
GOOG_NEWS-NX-SFT-0,02	0.022	0.071	901.6	673.4	344	0
GOOG_NEWS-N0-SFT-0,02	0.021	0.072	1034.2	561.8	323	0
GOOG_NEWS-N0-WFT-0,02	0.021	0.072	1034.2	561.8	323	0
WIKIPEDIA-NX-WFT-0,02	0.023	0.082	921	687.6	310.4	0
GOOG_NEWS-NX-WFT-0,02	0.023	0.084	869.4	764	285.6	0
WIKIPEDIA-NX-SFT-0,002	0.022	0.086	930.8	781.4	206.8	0
GOOG_NEWS-NX-SFT-0,002	0.022	0.087	932.6	777	209.4	0
GOOG_NEWS-NX-WFT-0,002	0.022	0.088	996.2	742.8	180	0
WIKIPEDIA-NX-WFT-0,002	0.022	0.088	987.8	748.2	183	0
WIKIPEDIA-N0-SFT-0,002	0.021	0.088	1097.2	645.8	176	0
WIKIPEDIA-N0-WFT-0,002	0.021	0.088	1097.2	645.8	176	0
GOOG_NEWS-N0-WFT-0,002	0.021	0.088	1118.4	615	185.6	0
GOOG_NEWS-N0-SFT-0,002	0.021	0.088	1118.4	615	185.6	0

Tabla 5.20: Resultados experimento deeplearning sust. experto, con preprocesado, DBpedia

Configuración	RMSE	MAE	DISTANCIA VECTORIAL			
			0.1	0.2	0.3	0.4<
WIKIPEDIA-N0-SFT-0,2	0.023	0.062	737	592.8	589.2	0
WIKIPEDIA-NX-SFT-0,2	0.023	0.062	737	592.8	589.2	0
WIKIPEDIA-N0-WFT-0,2	0.023	0.062	737	592.8	589.2	0
GOOG_NEWS-N0-WFT-0,2	0.023	0.062	737	592.8	589.2	0
GOOG_NEWS-N0-SFT-0,2	0.023	0.062	737	592.8	589.2	0
GOOG_NEWS-NX-SFT-0,2	0.023	0.062	737	592.8	589.2	0
WIKIPEDIA-NX-WFT-0,2	0.023	0.062	736.2	591.8	591	0
GOOG_NEWS-NX-WFT-0,2	0.023	0.062	734.8	589	595.2	0
WIKIPEDIA-NX-SFT-0,02	0.031	0.089	430.2	605.4	883.4	0
WIKIPEDIA-N0-WFT-0,02	0.03	0.089	462.2	600.4	856.4	0
WIKIPEDIA-N0-SFT-0,02	0.03	0.089	462.2	600.4	856.4	0
GOOG_NEWS-N0-WFT-0,02	0.03	0.09	453.8	631.8	833.4	0
GOOG_NEWS-N0-SFT-0,02	0.03	0.09	453.8	631.8	833.4	0
GOOG_NEWS-NX-SFT-0,02	0.031	0.09	396.2	652.8	870	0
WIKIPEDIA-NX-WFT-0,02	0.031	0.09	388	704.8	826.2	0
GOOG_NEWS-NX-WFT-0,02	0.031	0.091	363	738.6	817.4	0
WIKIPEDIA-NX-SFT-0,002	0.03	0.098	240	983.8	695.2	0
WIKIPEDIA-N0-WFT-0,002	0.026	0.098	593.6	709.2	616.2	0
WIKIPEDIA-N0-SFT-0,002	0.026	0.098	593.6	709.2	616.2	0
GOOG_NEWS-NX-SFT-0,002	0.03	0.099	263.8	982.4	672.8	0
GOOG_NEWS-N0-WFT-0,002	0.026	0.099	617.2	695	606.8	0
GOOG_NEWS-N0-SFT-0,002	0.026	0.099	617.2	695	606.8	0
WIKIPEDIA-NX-WFT-0,002	0.03	0.099	227	1007	685	0
GOOG_NEWS-NX-WFT-0,002	0.03	0.099	238.2	1013.4	667.4	0

5.3.1 Análisis de los Resultados de Amazon

Para los resultados de Amazon contra los expertos, con y sin preprocesamiento, mostrado en las tablas 5.13 y 5.15, analizando LWPque se basa en evaluar la "precision" de los resultados con el objetivo de proveer la probabilidad de relevancia de las clases obtenidas incluyendo una pequeña penalización, los resultados de la mejor configuración están muy cerca de los mejores resultados obtenidos con la mejor configuración del baseline. La diferencia es de 0,05. Esto puede parecer que no es una mejora, pero realmente sí que lo es. El motivos es que el número de votos de la máquina para alcanzar este valor ha disminuido en referencia al baseline. Concretamente la maquina tiene 17 votos menos, con 13 coincidencias menos. La diferencia entre los resultados con el preprocesamiento y sin él, en este caso concreto, es que sean más homogéneos. Es decir, la distancia entre los resultados de diferentes configuraciones del mismo score no se distancian demasiado en valor.

Al mismo tiempo, los resultados de LWR para esta configuración si que mejoran con respecto al baseline, concretamente en un 0,18 para los no preprocesados y en un 0,23 para los preprocesados. Esta métrica se centra en la evaluación del "recall" de los resultados con el objetivo de calcular la proporción de documentos relevantes recuperados, comparados con el total de documentos que son relevantes según la etiquetación de los expertos. Es decir, la cobertura del sistema automático. Además, se añade una pequeña penalización. Cabe destacar que la mejora también lleva la reducción de votos realizados por la máquina y las coincidencias, tal y como se ha explicado en la métrica LWP.

Así mismo, la métrica LCGM que es la media geométrica de la probabilidad de que una clase encontrada sea correcta y la probabilidad de que una clase de la frase sea encontrada, se queda muy próxima a los mejores resultados del baseline para los elementos con y sin preprocesado.

Los resultados de las métricas RMSE y MAE, las cuales sirven para calcular si un sistema de recomendaciones es óptimo teniendo en cuenta los aciertos y fallos y, siendo RMSE la métrica que penaliza más los grandes errores, han sido semejantes a los obtenidos con la mejor configuración del baseline. La mejora en estas métricas es realmente pequeña, pero sirve para verificar la hipótesis de que, este tipo de enfoque no desvía el contexto que los expertos tenían, sin sustituir a uno de ellos por un sistema automático. Además, el peor resultado de este enfoque mejora al del baseline en un 0,003.

En la parte de sustituir a un experto, hay que tener en cuenta que las diferencias entre los resultados de todos los expertos contra los resultados medios de sustituir a un experto, suponen la manera de calcular si el sistema realmente puede ser útil para sustituir a un experto por el sistema correspondiente con una configuración especifica. En las tablas 5.17 y 5.19 se puede ver que, la diferencia entre los resultados

de los expertos y los resultados medios de la sustitución no difieren en exceso. Estos no superan en ningún momento una distancia mayor a 0,4, equivalente a un 30 % de no acierto completo, tanto para el sistema que tiene la traducción como para el que no. Además, se puede ver que el número de votos que se encuentran en la franja del 0.3 ha disminuido para los elementos de Amazon. Este dato es importante, ya que es el que mantiene la relación entre los resultados que ofrece el sistema automático y las respuestas de los expertos. El análisis de estos resultados nos muestra que el consenso se mantiene tal y como si se hubiera metido a un nuevo experto por lo que el sistema, en principio lo damos por bueno para Amazon.

5.3.2 Análisis de los Resultados de DBpedia

Para los resultados de DBpedia contra los expertos, con y sin preprocesamiento, mostrado en las tablas 5.14 y 5.16, analizando LWP los resultados de la mejor configuración son un poco mejor que los resultados obtenidos con la mejor configuración del baseline. Con las frases sin preprocesamiento y preprocesadas, se mejora la detección de las temáticas en 0,15. La diferencia es bastante baja, pero los votos que efectúa la maquina son 14 menos, con tan solo una disminución de 5 coincidencias con los expertos.

Al mismo tiempo, los resultados de LWR para esta configuración son los que más han mejorado con respecto a los del baseline. La diferencia entre ellos es de 3,28 para las configuraciones sin preprocesado y de un 2,1 para las que si han sido preprocesadas. Así mismo, la métrica LCGM para las frases no preprocesadas y preprocesadas no alcanza a las del baseline, quedándose a 0.23 de estas.

Los resultados de las métricas RMSE y MAE son un poco mejores, pero muy semejantes a las obtenidas con las configuraciones del baseline. La diferencia en estas métricas es realmente pequeña, pero sirve para verificar que los modelos de GOOGLE_NEWS y WIKIPEDIA con las configuraciones más restrictivas son una posibilidad a tener en cuenta para la detección de la temática y filtrar los topics de las frases, tal y como lo haría un experto. Además, por primera vez en este tipo de enfoque, los resultados entre LWP, LCGM, RMSE y MAE coinciden en cuáles son las mejores configuraciones. Evidentemente, esta sintonía se ve mejor reflejada con los recursos no preprocesados que con los que sí lo están.

En la parte de sustituir a un experto, al igual que ha ocurrido con los resultados de Amazon, los de DBpedia no sufren absolutamente ninguna penalización en desviar el contexto general ni en romper el consenso que mantienen los expertos entre sí. En las tablas 5.18 y 5.20 se puede ver que, la diferencia entre los resultados de los expertos y los resultados medios de la sustitución no difieren en exceso. Estos no superan en ningún momento una distancia mayor a 0,4 tanto para el sistema que

tiene la traducción como para el que no. Los resultados muestran que la sustitución de un experto por nuestro sistema sería como meter un nuevo experto. Debido a esto, en principio lo damos por bueno también para DBpedia.

5.3.3 Conclusiones de los Resultados

Como análisis y conclusión general de los resultados, hay que mencionar que este experimento estaba planteado para verificar si los resultados obtenidos por modelos de deeplearning centrados en las relaciones de topics según artículos, tanto de GOOGLE_NEWS como de WIKIPEDIA, servían para mejorar la capacidad de nuestro sistema y poder sustituir a un experto.

El número de configuraciones utilizadas con preprocesado de las frases, filtrado de resultados y selección de diferentes niveles de búsqueda, nos han mostrado que para obtener categorías de productos, el modelo de deeplearning que se use es independiente, siempre que tenga bastante información acerca del propio producto. Por otra parte, para la detección de contexto mediante DBpedia, el mejor modelo ha resultado ser el de GOOGLE_NEWS, tanto para los elementos preprocesados como para los que no lo están. El motivo de esta situación se debe a que el modelo de GOOGLE_NEWS está construido de noticias que se han ido publicando en Google News. Estas noticias llevan consigo contextos más amplios sobre diversos temas. Por el contrario, los modelos de Wikipedia, al ser definiciones explicitas de elementos concretos, no detectan tan bien los contextos, pero con los productos, al igual que el modelo de Google News, lo hace perfectamente. Teniendo en cuenta estos resultados, es necesario seguir investigando en esa área utilizando otros tipos de modelos y fuentes para poder encontrar los mejores y que más impulsen la detección y filtrado de los resultados.

Un elemento importante a tener en cuenta para poder continuar una investigación con modelos de deeplearning es que las métricas muestran que es válido este planteamiento, pero que es necesario buscar e investigar modelos más específicos. La búsqueda de una fuente para la generación del modelo es bastante importante porque no todas las fuentes son válidas para el objetivo de la búsqueda. Por ejemplo, para una mejor detección de productos de Amazon, un posible nuevo modelo podría ser el generado por los comentarios de los usuarios por cada producto. De esta manera se tendría mucha más información sobre el producto y, entre otra información, nuevas formas por las cuales, los usuarios denomina al propio producto, junto con elementos que están muy relacionados con él. En el caso de la detección de contexto, un posible modelo podría ser el formado por todos los comentarios generados

en la plataforma reddit[2] o, en castellano, meneame[3]. De esta manera se conseguiría mucho contexto relacionado sobre ciertas temáticas.

Por último, en el caso de sustituir a un experto, la variación ha sido mínima. Esto es algo perfecto debido a que, con estas configuraciones el consenso que había dentro de los expertos no varía en exceso, simplemente es como si hubiera entrado un nuevo experto dentro del sistema.

5.4 Resultados Mapa de Traducción

Los anteriores enfoques, tal y como hemos visto en sus resultados, han sabido responder a las expectativas que teníamos. Han demostrado que son un camino viable para poder seguir explorando e investigando. En este apartado ya solo queda mostrar los resultados que hemos obtenido con el mapa de traducciones de DBpedia a Amazon. Estos resultados, al contrario que los anteriormente mostrados, únicamente muestran los que tienen que ver con Amazon. Como en la sección 4.4 esta descrito, esto se debe a que se utilizan los contextos obtenidos de DBpedia, en el experimento baseline, para traducirlos y comprobar los productos de Amazon que nos ofrece.

Estos resultados, al igual que en los anteriores, se ordenan por la métrica LWP, que vuelve a alinearse perfectamente con las configuraciones, siendo las mejores las más restrictivas y las peores las más laxas. El número de votos mínimos es el mismo que en el experimento baseline para la fuente Amazon y el número máximo de votos es de 15688, un poco por encima de los máximos votos para la misma fuente en el baseline. Este dato, en principio que sería malo, LWP muestra que no es tan malo como cabría esperar, ya que en comparación general, los resultados por la parte media y baja son mejores y más constantes.

Por último y al igual que en el experimento baseline y el resto, los resultados están ordenados por franjas de colores para una mejor comprensión y se muestran en las tablas 5.21 y 5.22.

5.4.1 Análisis de los Resultados

Los resultados de este trabajo solo comprenden los de Amazon ya que lo que se buscaba era dar un producto en base a un contexto detectado de DBpedia. Los resultados que se han obtenido son contra los expertos de Amazon sin preprocesamiento y se encuentran en la tabla 5.21. De estos resultados, analizando LWP que se basa en evaluar la “precision” de los resultados con el objetivo de proveer la probabilidad de

[2]https://www.reddit.com/
[3]https://www.meneame.net/

Tabla 5.21: Resultados experimento traducción con mapa de DBpedia a Amazon

Configuración	M	E	C	LWP avg	LWR avg	LCGM avg	RMSE	MAE
NX-WFT-0,2-no	1919	5405	1619	79.9668	27.2862	62.2565	0.079	0.119
NX-SFT-0,2-no	1919	5405	1619	79.9668	27.2862	62.2565	0.079	0.119
N0-SFT-0,2-no	1919	5405	1619	79.9668	27.2862	62.2565	0.079	0.119
N0-WFT-0,2-no	1919	5405	1619	79.9668	27.2862	62.2565	0.079	0.119
NX-WFT-0,2-yes	1928	5405	1619	79.9213	27.2416	62.2313	0.079	0.12
N0-WFT-0,2-yes	1928	5405	1619	79.9213	27.2416	62.2313	0.079	0.12
N0-SFT-0,2-yes	1928	5405	1619	79.9213	27.2416	62.2313	0.079	0.12
NX-SFT-0,2-yes	1957	5405	1622	79.8236	27.2294	62.1759	0.079	0.12
NX-WFT-0,02-no	2854	5405	1685	76.7568	27.0979	60.4947	0.08	0.125
N0-SFT-0,02-no	2843	5405	1672	76.7416	26.983	60.4059	0.08	0.125
N0-WFT-0,02-no	2843	5405	1672	76.7416	26.983	60.4059	0.08	0.125
NX-SFT-0,02-no	3889	5405	1732	72.3278	26.798	57.8676	0.081	0.132
NX-WFT-0,02-yes	6361	5405	1849	63.6806	24.8012	53.3035	0.082	0.144
N0-WFT-0,02-yes	6320	5405	1820	63.5831	24.5393	53.1456	0.082	0.144
N0-SFT-0,02-yes	6320	5405	1820	63.5831	24.5393	53.1456	0.082	0.144
NX-SFT-0,02-yes	8341	5405	1879	54.8069	23.052	47.4465	0.084	0.156
NX-WFT-0,002-no	11827	5405	1947	42.6508	19.452	40.5911	0.084	0.169
N0-SFT-0,002-no	11773	5405	1912	42.5125	19.0542	40.4752	0.085	0.169
N0-WFT-0,002-no	11773	5405	1912	42.5125	19.0542	40.4752	0.085	0.169
NX-SFT-0,002-no	14092	5405	1861	30.8111	16.294	31.7802	0.087	0.181
NX-WFT-0,002-yes	14711	5405	1904	30.1494	16.0056	31.9021	0.086	0.181
N0-WFT-0,002-yes	14642	5405	1872	30.0127	15.7418	31.9395	0.086	0.181
N0-SFT-0,002-yes	14642	5405	1872	30.0127	15.7418	31.9395	0.086	0.181
NX-SFT-0,002-yes	15688	5405	1890	25.6223	14.6416	28.5263	0.087	0.185

Tabla 5.22: Resultados experimento traducción con mapa de DBpedia a Amazon, sustitución de experto

Configuración	RMSE	MAE	VECTORS DISTANCE			
			0.1	0.2	0.3	0.4<
NX-WFT-0,2-no	0.019	0.052	1134.8	426.4	357.8	0
N0-SFT-0,2-no	0.019	0.052	1134.8	426.4	357.8	0
NX-SFT-0,2-no	0.019	0.052	1134.8	426.4	357.8	0
N0-WFT-0,2-no	0.019	0.052	1134.8	426.4	357.8	0
N0-WFT-0,2-yes	0.019	0.052	1134.8	426.8	357.4	0
NX-WFT-0,2-yes	0.019	0.052	1134.8	426.8	357.4	0
N0-SFT-0,2-yes	0.019	0.052	1134.8	426.8	357.4	0
NX-SFT-0,2-yes	0.019	0.052	1134.6	427.4	357	0
NX-WFT-0,02-no	0.019	0.055	1128.8	444.6	345.6	0
N0-SFT-0,02-no	0.019	0.055	1127.6	444.4	347	0
N0-WFT-0,02-no	0.019	0.055	1127.6	444.4	347	0
NX-SFT-0,02-no	0.019	0.059	1130.2	451.8	337	0
NX-WFT-0,02-yes	0.019	0.065	1122	486.4	310.6	0
N0-SFT-0,02-yes	0.019	0.065	1116	492.4	310.6	0
N0-WFT-0,02-yes	0.019	0.065	1116	492.4	310.6	0
NX-SFT-0,02-yes	0.02	0.07	1120.6	503.8	294.6	0
NX-WFT-0,002-no	0.02	0.076	1137	520	262	0
N0-SFT-0,002-no	0.02	0.076	1130.4	534.8	253.8	0
N0-WFT-0,002-no	0.02	0.076	1130.4	534.8	253.8	0
NX-SFT-0,002-no	0.02	0.082	1130.8	541.8	246.4	0
NX-WFT-0,002-yes	0.02	0.082	1145.4	531.6	242	0
N0-SFT-0,002-yes	0.02	0.082	1133.8	557.2	228	0
N0-WFT-0,002-yes	0.02	0.082	1133.8	557.2	228	0
NX-SFT-0,002-yes	0.02	0.084	1148.6	537.8	232.6	0

relevancia de las clases obtenidas incluyendo una pequeña penalización, los resultados de las mejores configuración son muy semejantes a los resultados obtenidos con las mejores configuraciones del baseline. La mejora más significativa se encuentra en los resultados de las peores configuraciones. Esta mejora nos permite no mostrar, en configuraciones más laxas, unas temáticas que, según los expertos, difieren en demasía con su criterio.

Al mismo tiempo, los resultados de LWR son semejantes a los del baseline, no mostrando un empeoramiento o mejora significativa. Esta métrica se centra en la evaluación del "recall" de los resultados con el objetivo de calcular la proporción de documentos relevantes recuperados, comparados con el total de documentos que son relevantes según la etiquetación de los expertos. Es decir, la cobertura del sistema automático. Además, se añade una pequeña penalización.

Así mismo, la métrica LCGM que es la media geométrica de la probabilidad de que una clase encontrada sea correcta y la probabilidad de que una clase de la frase sea encontrada, también mejora los peores resultados en comparación con el baseline.

Los resultados de las métricas RMSE y MAE, las cuales sirven para calcular si un sistema de recomendaciones es óptimo teniendo en cuenta los aciertos y fallos y, siendo RMSE la métrica que penaliza más los grandes errores, sí que muestran una mejora significativa en comparación con los resultados del baseline. La mejora en estas métricas se muestra en que el mejor resultado de este experimento se distancia en 0,0027 del mejor resultado del baseline en el caso de RMSE y entorno a un 0,0045 en el caso de MAE. Esta mejora sirve para verificar que este enfoque, de manera global, mejora la detección de la temática y obtiene una ventaja a la hora de detectar y filtrar los topics de las frases, tal y como lo haría un experto. Además, entre los resultados de este experimento, con estas métricas, los saltos no son tan notables como en los del baseline.

En la tabla 5.22 se puede ver que, la diferencia entre los resultados de los expertos y los resultados medios de la sustitución no difieren en exceso, mejorando los resultados del baseline, ya que aumentan los aciertos que están en el umbral del 0,1 y disminuyen los que están en el de 0,3. Además, no superan en ningún momento una distancia mayor a 0,4. El análisis de estos resultados nos muestra que el consenso se mantiene tal y como si se hubiera metido a un nuevo experto por lo que el sistema, en principio lo damos por bueno.

Este primer enfoque directo para ofrecer productos en base a contextos extraídos de frases cortas nos muestra que es posible realizarlo. Nos muestra también que el recurso del mapa es una herramienta muy necesaria para realizar estos procesos. En el aspecto de partir de los datos de DBpedia obtenidos del baseline, se podía haber utilizado los generados por cualquier otro enfoque mostrado en este trabajo, pero hemos pensado que es mejor utilizar el del baseline para después poder ir mejoran-

do y tener una base contra la que compararlo. Además, los datos de sustituir a un experto, nos han mostrado que se asemejan mucho más que los anteriores enfoques al criterio de los expertos. Esto puede ser bueno o malo, según el objetivo que se persiga. Es bueno en el aspecto de tener el mismo pensamiento y preferencias que los expertos que han realizado la etiquetación, pero es malo debido a que lo que se busca no es que el sistema sea igual a un conjunto de expertos, si no que tenga cierta discordancia. Ningún Ser Humano piensa exactamente igual que otro, por lo que nuestro sistema debería de dar ciertas respuestas un poco diferentes a las del resto para parecer otro experto. En definitiva, creemos que la utilización del mapa, junto con una de las técnicas mostradas en este trabajo, son una buena opción y creemos que seguir investigando en este campo, con estos recursos y estas técnicas o nuevas puede resultar en mejores detecciones de temática en textos cortos.

"Perfectirijillo"

Margaret Evelyn Simpson
(1986 –)

6

Prueba de concepto

La realización una investigación tiene que servir para, de alguna manera, ir mejorando la vida de las personas, descubrir nuevas realidades, dar respuesta a ciertas preguntas y generar otras nuevas. Si es cierto que algunas investigaciones que se realizan se quedan un poco olvidadas debido a su naturaleza teórica, su complejidad computacional o a la imposibilidad de la tecnología actual para poder llevarla a cabo.En el caso de este trabajo, una manera de mostrar el resultado de este nuevo recurso es mediante una herramienta pública. Para ello, hemos creado un buscador, en principio de regalos, que mediante una frase o contexto, ofrece una categoría de libros de Amazon que pueden interesar al usuario. Este buscador utiliza la relación creada entre las clases de DBpedia como contexto y las categorías de libros de Amazon como producto. Para mejorar el proceso de búsqueda, se ha desarrollado un sistema que utiliza la herramienta *word2vec* descrita en la sección 4.3.2.

6.1 Descripción de la herramienta Web

La herramienta web tiene 3 elementos principales. El primero de ellos se muestra en la figura 6.1 y está centrado en los parámetros de la búsqueda. Este elemento, a su vez, está dividida en 3 sub-elementos.

El primer sub-elemento corresponde a la configuración del filtro de búsqueda ubicado en la parte superior. La configuración del filtro contiene: los elementos a filtrar para realizar las búsquedas por cualquier palabra, término o frase, el modelo de *word2vec* sobre el cual se quieren realizar las búsquedas (por el momento solo

existen 2 modelos, Wikipedia y GoogleNews) y el límite de semejanza mínimo para aceptar los resultados obtenidos del modelo seleccionado.

El segundo sub-elemento de la herramienta web corresponde a las clases de DBpedia que pueden ser seleccionadas por el usuario. Estos clases, inicialmente no están filtradas y se muestran todas, pero en el momento en el que el usuario introduce cualquier valor a buscar, este sub-elemento se filtra con los posibles resultados. El proceso de filtrado muestra los posibles resultados que, según la herramienta de distancia de *word2vec*, están más próximos a los términos de búsqueda introducidos según el modelo seleccionado y el límite de similitud elegido. La selección de cualquier topic de DBpedia afecta a las siguientes búsquedas, ya que cada selección filtra los resultados sobre los cuales se pueden volver a realizar búsquedas.

El tercer sub-elemento corresponde a los topics seleccionados de DBpedia. Estos se muestran en este área y pueden ser eliminados para que el filtro tenga en cuenta otros resultados.

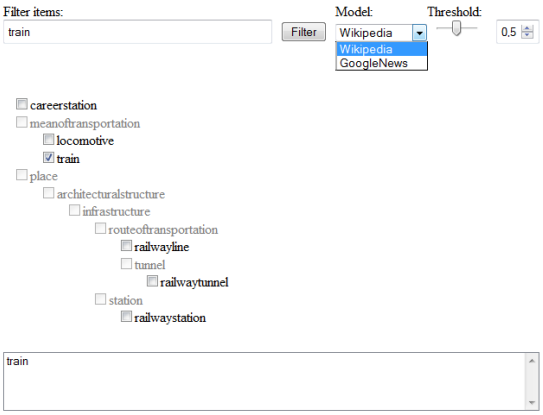


Figura 6.1: Sección de filtrado de la búsqueda de la herramienta Web.

El segundo elemento principal de la herramienta Web que puede verse en la figura 6.2, muestra las sugerencias de las categorías de libros de Amazon asociadas al topic de DBpedia seleccionado previamente. Estos resultados están enlazados con las páginas principales de dichas categorías en Amazon.

El tercer elemento principal de la herramienta web que se muestra en la figura 6.3, está focalizado en las opiniones de los usuarios que utilizan esta herramienta. En este elemento, el usuario puede valorar los resultados obtenidos y escribir sus impresiones sobre la herramienta y su funcionamiento para futuras mejoras.

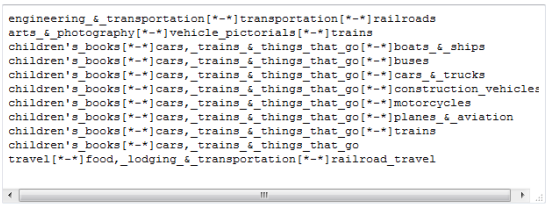


Figura 6.2: Resultados de la búsqueda de la herramienta Web.

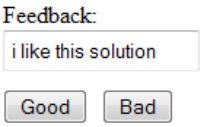


Figura 6.3: Herramienta Web.

6.2 Funcionalidad

La herramienta Web sirve para, dado una frase o contexto, se obtengan categorías de productos de la tienda de libros de Amazon. Su funcionamiento está basado en 2 pasos. El primero es utilizar el mapa de relaciones que se ha descrito en la sección 3.3.1. Y el segundo es, mediante la utilización de este mapa, obtener la relación entre los contextos que existentes en la cadena de búsqueda y en las diferentes categorías de libros de Amazon. Para solucionar el problema que surge con la necesidad de conocer todos las clases de DBpedia y sus contextos para poder realizar una búsqueda correcta, hemos decidido utilizar la herramienta *word2vec*.

Esta herramienta utiliza diversos corpus completos de texto como entrada para generar vectores de palabras. Estos vectores son los que emplea para comparar la similitud entre las palabras que ya tiene almacenadas y las nuevas palabras de las búsquedas. Para lograrlo, construye un modelo de vocabulario con los datos del texto de entrada. Estos datos los utiliza en el entrenamiento, para después generar la representación vectorial de las palabras.

Los modelos vectoriales del vocabulario que hemos utilizado han sido 2. El primero ha sido el de Wikipedia (con más de 3 billones de palabras) y el segundo ha sido el de GoogleNews (con más de 100 billones de palabras). Cada modelo lo hemos obtenido de diferentes maneras:

- El modelo de Wikipedia ha sido creado utilizando el último volcado de datos de los articulos en inglés de Wikipedia, disponible en el repositorio de Wikipedia[1] como entrada.

[1]https://dumps.wikimedia.org/enwiki/latest/

- El modelo de GoogleNews es el modelo vectorial pre-entrenado de palabras y frases de *word2vec*[2], disponible en su web[3].

Por último, utilizando la herramienta de aprendizaje profundo *word2vec* y el mapa generado, hemos resuelto la relación entre las frases aleatorias que un usuario puede introducir y los topics existentes de DBpedia, simplificando el uso de la herramienta web desarrollada como prueba de concepto.

[2]https://code.google.com/p/word2vec/
[3]https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit?usp=sharing

"Dos no discuten si uno no quiere"

Francisco García Collado
(1933 –)

7

Conclusions

Along this work we have described in detail and empirically evaluated the main contributions of the present dissertation. Now is time to measure the level of accomplishment that this work has achieved according to the established fundamental hypothesis and goals.

On the remainder of this chapter, we are going to show the objectives that we have followed to respond to the proposed hypothesis. Furthermore, we are going to detail the contributions that this work brings, what are the limitations that we have encountered and some ways, in that we believe, to made new improvements. After that, we are going to explain a general discussion of all the results and, at last, we are going to introduce some possible future work lines which are very interesting to continue to improve and evolve the system.

7.1 Main contributions

After presenting our proposed approaches and the evaluation results, it is now time to revisit once again our initial hypothesis:

> «*It is possible to generate a methodology capable of improving the topic detection in short sentences to improve targeted marketing systems within social networks and instant messaging system.*»

Moreover, reminding the proposed objectives that we have set out in section 1 to give a response to proposed hypothesis, which formulate the following tasks:

1. Consider one approach or system to get the context from sentences and other mechanism to detect the products that could be linked to that context.
2. The generated approaches or systems follow a methodology to be replicable.
3. Set out different research lines or approaches to test if there are viable for future experiments.
4. Check if any proposed research lines could have a real commercial approach.

Next we summarise the main contributions, such as new generated resources, methodologies, proof of concept and models that are the result of the study of the short sentences categorization problems applied to marketing area.

1. **A methodology to generate knowledge base information.** On chapter 3 we have shown the steps and the reasoning behind the generation of the two knowledge bases that we have used in this work. This contribution serves to have a methodological way to generate these important resources. Stating by the selection of the sources and their features, continuing by the description of the steps to convert data to information and ending by the system to keep update these knowledge bases. The final result has been one methodology, explained on section 3.2.6, that has been used to generate knowledge bases for both DBpedia and Amazon.
2. **A linked map between DBpedia topics and Amazon books categories.** On chapter 3.3 we have shown the steps and the used system to generate the relations between the Amazon and DBpedia knowledge bases. For Amazon, the classes are books categories and, for DBpedia, the classes are existing elements within the topics taxonomy that are used to label each article of Wikipedia. The final result has been a new resource that links the context from DBpedia with the products from Amazon.
3. **A baseline, datasets and future experiments.** Chapter 4 shows the different approaches used in this work, as well as used methodologies, generated datasets to test the results and generate baseline to compare the rest of approaches results. These approaches sets the future research lines in that area because, there are important and relevant approaches to detect the context in short sentences. The baseline experiment, detailed on section 4.1, provides a start point to compare the results of the rest of the new approaches using the generated

new datasets. That datasets are available on the author Web Site[1] to be used by anybody that want improve or make new approach.

4. **A proposition of three new metrics to evaluate results in this research area.** Sections 4.1.4.1, 4.1.4.2 and 4.1.4.3 detail the functioning of these new metrics, focused on the evaluation of the results of these new approaches on context detection in short sentences. These metrics are evolution of a well known metrics in information retrieval, described on section 2.2.4. The goal for these new metrics is to check and evaluate the correct operation of the proposed system, taking into account the global system restrictions and peculiarities.

5. **A viability demonstration of three different approaches using the generated resources.** Section 5 shows the obtained results with each different proposed approach. Also, on this section, it is demonstrated the viability of each approach, but with more improvements and more research. These contributions try to set the bases for future researches and experiments in this area.

6. **A new Deep Learning model.** Section 4.3.3 presents the definition of the proposed experiment to check the deep learning approach. In the same section, it is also described the followed process to create the Wikipedia model to use into the experiment. A model that has been published and is available for the scientific community.

7. **A tool, a PoC, that uses the generated resources to offer gifts.** Of an interesting area also showing open lines of future research, this *PoC*, detailed on section 6 is a system to offer gifts, based on the context extracted from short sentences. From these sentences, the tool obtains the most relevant information such as preferences or topics. With this information, the tool filters the related families of books from Amazon and offers then as products to be bought as a present.

These resources allows us to answer to the proposed hypothesis. With the described methodologies, the generated knowledge bases, with the linked map and with the different proposed research lines, we can check that it is possible offer products based on short sentences. Finally, the proof of concept system described in this work, using the resources confirms the validity of the proposed hypothesis.

[1] paginaspersonales.deusto.es/patxigg/

7.2 Limitations

There are several important points to be discussed referring to the appropriateness of our proposed methods. Next we introduce some of them. These limitations vary from physical elements, such as storage, to technical restrictions, such as the techniques that we have used in each approach.

In the physical aspect, storage has been a big problem. The amount of data that has to be processed, stored and, in general, use, has been a real challenge. In addition, to process this information we have needed powerful computers. In the section of the programming languages, we have found that the most commonly used languages, like Java, in some parts of this work are useless. The main limitations that we have found with these languages are the maximum limits that entails lists and maps. But this limitations, only have affected us in the generation of deep learning models. In the rest of the approaches and processes, their performance has been perfect.

Regarding the limitations related to existing research techniques, the major limitation has been and, currently is, the processing of natural language. Linguistic phenomena like synonymy, metonymy, homonym, words ambiguity, meaning of each word, opinion mining and sentiment analysis gives to this area a special and beautiful mean to continue working to solve each one. However, these limitations must be taken into account. In addition, new word, terms, slang or jargon that are appearing also affect to natural languages. These elements, also, make a big influence and generate some limitations to consider. Moreover, the retrieval of the necessary resources to generate the deep learning models, is quite complicated. Generating valid models for this approach requires a lot of information that, currently and unfortunately, is not available to the scientific community.

On the other hand, it could be seen as a limitation that this system only works with DBpedia contexts and Amazon products. But this is not a real limitation because we have described the methodologies that allow to port this approach to any other area. The main limitation that we see to achieve the shown results is, the necessary and costly, manual labelling of the new elements to get a new viable resource for our system.

7.3 General discussion about work

There are several important points to be discussed referring to the appropriateness of our proposed methods. Next we introduce some of them.

Targeted advertising is a marketing area that offers a certain quality and more impact in the products offered to users. Many companies expend millions to create marketing campaigns about some products and, the used approach is to flood users

with advertisements and other type of paraphernalia ads. One big problem with this approach is that the users needs and preferences are not taken into account. Recently, targeted marketing is having more relevance in the business world. In fact, companies like Google or Facebook use the user's data to learn and create specific profiles about their users to offer products that may be of interest to certain types of users. The problem is that, in the scientific community, there are not many resources that can help to make and improve the research in this area. With our work, we contribute to the scientific community by publishing knowledge bases, deep learning models, a linked map and methodologies to port this approach to any business area.

Most, if not all, of the companies that are focused on selling products to the end users, or in generating marketing campaigns, typically, use systems to classify users based on extracted features from themselves. These features can be the purchases that each user has made, the place where a user lives, age, etc. These systems impersonate the user, creating a big users groups to offer, to all of them, the same. In addition, these systems use loyalty schemes, towards a reward approaches that consist on making questions and getting user's answers about their satisfaction with the received service or/and product.

In this work we have proposed an approach to avoid, first, making users clusters and, second, to avoid the use of massive flooding or spamming techniques to try to sell to users products that they do not need, they do not like or simply they do not want. In addition, we have proposed a system that works as an expert who identifies, based on what users are talking about, the field in which they could offer a specific product. In this way, it is like if the user had a personal shopper.

It is important to remark that the system can be proposed as a person. That is, depending of the person education, environment, knowledge, society, experience, etc., each person has certain ideas and ways to answer to a specific questions. Our system works quite similarly. Depending on the source that is used to generate the knowledge base, the answers could be similar or completely different.

In conclusion, the system, taking as a reference the generated knowledge base, can adapt to different topics.

7.4 Future lines

Here we present some possible solutions to the main shortcomings discussed in the previous section, along with the future open lines of research identified in the present dissertation.

In this work, we have used different techniques in the preprocessing step to prepare the data to get the best results from knowledge bases. One step of this prepro-

cessing was the translation from colloquial language to formal language. This step of translation, could be improved using other systems improving the final outcome of the recommendation system. One possibility is the presented system by Sridhar et al. (2014). This unsupervised system uses a distributed words representation, learned by neural networks, for the SMS terms representation translation. The use of these type of systems, would simplify our proposed preprocessing step and, also, would increase the speed of the system, by avoiding an external request to a system that we do not control. Another important aspect of the proposed preprocessing step is the selection of relevant terms. In this work we have used a very common topic extraction method found in the literature: Term Frequency (TF) and LDA. But there are others that may be interesting to try. For example Wavelet Packet Transforms proposed by Mahajan y Sharmistha (2015). This method is used in short sentences, with noise and empty elements, obtaining very good results.

Is true that, for preprocessing step, there are other possible improvements, many filters and many techniques that we have not been discussed, but our goal is plant the seed, as an idea for future development of this important step within the whole architecture of the proposed approach.

Another important future line of researching is analysing approaches and areas in which this type of system could succeed. For example, one area that could be consider interesting to apply our methodologies to is in generating new marketing campaigns and the tracking along social networks. Also, it could be interesting to apply other works based on the feelings detection, like Nguyen y Shirai (2015), that can predict the markets price movements based on the feelings analysis on social networks. This information could be used to generate new marketing campaigns, more focused in each user, and check the acceptance of these campaigns by each user, making a sentimental analysis of their comments on online social networks. The aims of this approach, for example, could be the stock planning of a future product.

On the context expansion section, we use some types os approaches to achieve it, but there are others that could be tested. For example, Kenter y de Rijke (2015) developed a system to optimize the searches for similar sentences. In addition, this system could optimize these searches using semantic features or exploring the emotional aspects of the sentences to offer products based on detected feelings. The sentiment analysis on texts is a subject that, in the last years, has attracted the interest of both the scientific community and the industry. Another interesting work in the area of short sentences and feeling analysis, that could use to improve or make a new line of research with our work, is the work of dos Santos y Gatti (2014). This paper studies short sentences, using neural networks to detect the feelings.

As happens with the preprocessing step, the approaches and areas where this

work could be used is very wide.

Moreover, a vital element that carries most of the effort of this work is the knowledge base and map relationships generation and update. The process to have these resources updated is very important because, if these resources are not updated, the system could become outdated with the people criteria and fail in the obsolescence. One option to keep these resources updated could be Kim et al. topics tracking system. This system enhances the offering of products using current detected topics based on historical data. Besides, the map relations and dataset labels could be improved and expanded using the community intelligence systems such as Amazon Turk[2].

On the other hand, in the automatic generation of labels for the resources, we could used neural networks as Zeng et al. (2015) or Zhang et al. (2015b). Or, also, we could use domain adaptive systems as used by Eisenstein to discover relationships throw the adaptation from explicit to implicit elements.

In addition, instead of using only the proposed system, we could used methodologies as proposed by Hu et al. to generate compressible topics hierarchies, based on the Wikipedia article's text and contents. This methodology could help generating new knowledge bases from scratch or improving the ones we have created.

Finally, the work of Witten y Milne (2008), offers an approach to obtain the semantic relationships between different articles in Wikipedia in a very efficient way. This approach has the advantage of using the original HTML elements from the Wikipedia portal to make the unions and to be updated. But the problem arises when, for any reason of renovation, maintenance or attack, the Wikipedia site service is interrupted. Using or iterating along on-line documents, always has a penalization in time and this problem is an important point to be considered. Even so, this system uses the conceptual links, that exist in the articles to relate to each other, to find their connections. On the other hand, the Milne y Witten (2008) work uses machine learning methods to identify significant terms within unstructured text, and enrich them with appropriate links to Wikipedia articles.

The disadvantages that we have with these approaches is that, in our case, there is no documentation about the relations between Wikipedia or DBpedia articles and Amazon products. So, in order to carry out our experiments without the need of creating a request system to get these relations and, also, to have always the same resources to validate our approach and systems, we have chosen for get and use the DBpedia dump files. In the near future, it would be great to use the generated resources in this work along with tools and approaches that we have describe before such as Witten y Milne (2008) and Milne y Witten (2008), among others, to achieve a more effective system.

[2]www.mturk.com/mturk/welcome

Bibliografía

Abney, S. y Light, M. (1999). Hiding a semantic hierarchy in a Markov model. In *Proceedings of the ACL Workshop on Unsupervised Learning in Natural Language Processing*, pages 1–8.

Adams, P. H. y Martell, C. H. (2008). Topic detection and extraction in chat. In *Semantic Computing, 2008 IEEE International Conference on*, pages 581–588. IEEE.

Aggarwal, N. y Buitelaar, P. (2012). A system description of natural language query over dbpedia. *Proc. of Interacting with Linked Data (ILD 2012)[37]*, pages 96–99.

Ali, H., dÁvila Garcez, A. S., Tran, S. N., Zhou, X., y Iqbal, K. (2014). Unimodal late fusion for nist i-vector challenge on speaker detection. *Electronics Letters*, 50(15):1098–1100.

Allan, J., Carbonell, J., Doddington, G., Yamron, J., y Yang, Y. (1998). Topic detection and tracking pilot study final report.

Amato, G. y Straccia, U. (1999). User profile modeling and applications to digital libraries. In *Research and Advanced Technology for Digital Libraries*, pages 184–197. Springer.

Ammar, W., Darwish, K., El Kahki, A., y Hafez, K. (2011). Ice-tea: in-context expansion and translation of english abbreviations. In *Computational Linguistics and Intelligent Text Processing*, pages 41–54. Springer.

Anagnostopoulos, A., Broder, A. Z., Gabrilovich, E., Josifovski, V., y Riedel, L. (2007). Just-in-time contextual advertising. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 331–340. ACM.

Anagnostopoulos, A., Broder, A. Z., Gabrilovich, E., Josifovski, V., y Riedel, L. (2011). Web page summarization for just-in-time contextual advertising. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(1):14.

Andreevskaia, A. y Bergler, S. (2006). Mining WordNet for fuzzy sentiment: Sentiment tag extraction from WordNet glosses. volume 6, page 209–216.

Anglin, J. M. y Miller, G. A. (2000). *Vocabulary development: A morphological analysis*. Wiley-Blackwell.

Anklesaria, F., McCahill, M., Lindner, P., Johnson, D., Torrey, D., y Albert, B. (1993). The internet gopher protocol (a distributed document search and retrieval protocol).

Aprosio, A. P., Giuliano, C., y Lavelli, A. (2013). Automatic mapping of wikipedia templates for fast deployment of localised dbpedia datasets. In *Proceedings of the 13th International Conference on Knowledge Management and Knowledge Technologies*, page 1. ACM.

Apté, C., Damerau, F., y Weiss, S. (1994). Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems (TOIS)*, 12(3):233–251.

Argamon, S., Koppel, M., y Avneri, G. (1998). Routing documents according to style.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., y Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.

Baeza-Yates, R., Ribeiro-Neto, B., et al. (1999). *Modern information retrieval*, volume 463. ACM press New York.

Baker, J. K. (1979). Trainable grammars for speech recognition. *The Journal of the Acoustical Society of America*, 65(S1):S132–S132.

Ballmer, T. T. y Brennenstuhl, W. (1980). *Speech act classification: A study of the lexical analysis of English speech activity verbs*. Springer-Verlag.

Baum, L. E., Petrie, T., Soules, G., y Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, pages 164–171.

Becker, C. y Bizer, C. (2009). Exploring the geospatial semantic web with dbpedia mobile. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(4):278–286.

Beel, J., Gipp, B., y Wilde, E. (2010). Academic search engine optimization (aseo). *Journal of scholarly publishing*, 41(2):176–190.

Belkin, N. y Croft, W. (1992). Information filtering and information retrieval: two sides of the same coin? *Communications of the ACM*, 35(12):29–38.

Bengel, J., Gauch, S., Mittur, E., y Vijayaraghavan, R. (2004). Chattrack: Chat room topic detection using classification. In *Intelligence and Security Informatics*, pages 266–277. Springer.

Bengio, Y. (2009). Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127.

Bingham, E., Kabán, A., y Girolami, M. (2003). Topic identification in dynamical text by complexity pursuit. *Neural Processing Letters*, 17(1):69–83.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., y Hellmann, S. (2009). Dbpedia-a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154–165.

Blei, D. M., Ng, A. Y., y Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Blitzer, J., Dredze, M., y Pereira, F. (2007). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, volume 7, pages 440–447. Citeseer.

Boo, V. K. y Anthony, P. (2012). Agent for mining of significant concepts in dbpedia. In *Knowledge Technology*, pages 313–322. Springer.

Broder, A., Fontoura, M., Josifovski, V., y Riedel, L. (2007). A semantic approach to contextual advertising. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 559–566. ACM.

Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., y Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

Buckland, M. (2002). Emanuel goldberg, 1881-1970: Pioneer of information science.

Buckley, C., Salton, G., Allan, J., y Singhal, A. (1995). Automatic query expansion using smart: Trec 3. *NIST SPECIAL PUBLICATION SP*, pages 69–69.

Bush, V. (1945). As we may think.

Camacho-Collados, J., Pilehvar, M. T., y Navigli, R. (2015). A unified multilingual semantic representation of concepts. *Proceedings of ACL, Beijing, China*.

Cao, W., Hong, R., Wang, M., y Hua, X. (2014). Multifold concept relationships metrics. In *Advances in Multimedia Information Processing–PCM 2014*, pages 238–247. Springer.

Cardie, C., Farina, C., Bruce, T., y Wagner, E. (2006). Using natural language processing to improve eRulemaking.

Carrasco, J. J., Joseph, J., Daniel, C., Fain, C., Lang, K. J., y Zhukov, L. (2003). Clustering of bipartite advertiser-keyword graph.

Carreras, X., Chao, I., Padró, L., y Padró, M. (2004a). Freeling: An open-source suite of language analyzers. In *LREC*.

Carreras, X., Chao, I., Padró, L., y Padró, M. (2004b). Freeling: An open-source suite of language analyzers. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*.

Cassidy, T., Ji, H., Ratinov, L.-A., Zubiaga, A., y Huang, H. (2012). Analysis and enhancement of wikification for microblogs with context expansion. In *COLING*, volume 12, pages 441–456.

Chen, X., Liu, Z., y Sun, M. (2014). A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1025–1035.

Chen, Y., Yang, Y., Zhang, H., Zhu, H., y Tian, F. (2012). A topic detection method based on semantic dependency distance and plsa. In *Computer Supported Cooperative Work in Design (CSCWD), 2012 IEEE 16th International Conference on*, pages 703–708. IEEE.

Chen, Y.-L., Chen, Z.-C., Chiu, Y.-T., y Chang, C.-L. (2015). An annotation approach to contextual advertising for online ads. *Journal of Electronic Commerce Research*, 16(2).

Chong, A. Y. L., Ch'ng, E., Liu, M. J., y Li, B. (2015). Predicting consumer product demands via big data: the roles of online promotional marketing and online reviews. *International Journal of Production Research*, (ahead-of-print):1–15.

Cleverdon, C. (1967). The cranfield tests on index language devices. In *Aslib proceedings*, volume 19, pages 173–194. MCB UP Ltd.

Cohen, M. (2013). Semantics for mapping relations in skos. In *Web Reasoning and Rule Systems*, pages 223–228. Springer.

Collobert, R. y Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Conrad, J. G. y Schilder, F. (2007). Opinion mining in legal blogs. page 231–236.

Cormack, G. y Lynam, T. (2005). TREC 2005 spam track overview. In *The Fourteenth Text REtrieval Conference (TREC 2005) Proceedings*.

Croft, W. y Harper, D. (1979). Using probabilistic models of document retrieval without relevance information. *Journal of documentation*, 35(4):285–295.

Cullingford, R. E. (1978). Script application: computer understanding of newspaper stories. Technical report, DTIC Document.

Dagan, I., Marcus, S., y Markovitch, S. (1993). Contextual word similarity and estimation from sparse data. page 164–171.

Dave, K. S. y Varma, V. (2010). Pattern based keyword extraction for contextual advertising. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1885–1888. ACM.

Davis, P. K., Perry, W. L., Brown, R. A., Yeung, D., Roshan, P., y Voorhies, P. (2013). *Using Behavioral Indicators to Help Detect Potential Violent Acts*. RAND Corporation.

de-la Peña-Sordo, J., Santos, I., Pastor-López, I., y Bringas, P. G. (2013). Filtering trolling comments through collective classification. In *Network and System Security*, pages 707–713. Springer Berlin Heidelberg.

de-la Pena-Sordo, J., Santos, I., Pastor-López, I., y Bringas, P. G. (2014). Social news website moderation through semi-supervised troll user filtering. In *International Joint Conference SOCO'13-CISIS'13-ICEUTE'13*, pages 577–587. Springer International Publishing.

Deerwester, S., e. a. (1988). Improving information retrieval with latent semantic indexing. In *Proceedings of the 51st Annual Meeting of the American Society for Information Science 25*, pages 36–40. ASIS.

Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., y Ouellet, P. (2011). Front-end factor analysis for speaker verification. *Audio, Speech, and Language Processing, IEEE Transactions on*, 19(4):788–798.

Dempster, A. P., Laird, N. M., y Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38.

Deng, L., He, X., y Gao, J. (2013). Deep stacking networks for information retrieval. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 3153–3157. IEEE.

Dimitrova, M., Finn, A., Kushmerick, N., y Smyth, B. (2002). Web genre visualization.

Dong, H., Cheung Hui, S., y He, Y. (2006). Structural analysis of chat messages for topic detection. *Online Information Review*, 30(5):496–516.

dos Santos, C. N. y Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING), Dublin, Ireland*.

Dostert, L. E. (1955). The georgetown-ibm experiment. *1955). Machine translation of languages. John Wiley & Sons, New York*, pages 124–135.

Drucker, H., Wu, D., y Vapnik, V. (1999). Support vector machines for spam categorization. *IEEE Transactions on Neural networks*, 10(5):1048–1054.

Dumais, S., Platt, J., Heckerman, D., y Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. In *Proceedings of the seventh international conference on Information and knowledge management*, pages 148–155. ACM.

Dumais, S. T., Furnas, G. W., Landauer, T. K., Deerwester, S., y Harshman, R. (1988). Using latent semantic analysis to improve access to textual information. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 281–285. ACM.

Durbin, S. D., Richter, J. N., y Warner, D. (2003). A system for affective rating of texts. *KDD Wksp. on Operational Text Classification Systems (OTC-3)*.

EAGLES, M. (1995). Evaluation Working Group. 1996. EAGLES Evaluation of Natural Language Processing Systems: Final Report. *EAGLES Document EAGEWG-PR. 2, ISBN 87-90708-00*, 8.

Efron, M. (2004). Cultural orientation: Classifying subjective documents by cociation analysis.

Eisenstein, Y. J. G. Z. J. Closing the gap: Domain adaptation from explicit to implicit discourse relations.

Elkan, C. (2006). Clustering documents with an exponential-family approximation of the dirichlet compound multinomial distribution. In *Proceedings of the 23rd international conference on Machine learning*, pages 289–296. ACM.

Ellison, N. B., Vitak, J., Gray, R., y Lampe, C. (2014). Cultivating social resources on social network sites: Facebook relationship maintenance behaviors and their role in social capital processes. *Journal of Computer-Mediated Communication*.

Elsayed, T., Oard, D. W., y Namata, G. (2008). Resolving personal names in email using context expansion. In *ACL*, pages 941–949.

Esuli, A. y Sebastiani, F. (2006). Sentiwordnet: A publicly available lexical resource for opinion mining. volume 6, page 417–422.

Fairthorne, R. y Mooers, C. (1968). *Towards information retrieval*. Archon Books.

Fan, T.-K. y Chang, C.-H. (2011). Blogger-centric contextual advertising. *Expert Systems with Applications*, 38(3):1777–1788.

Fire, M. y Schler, J. (2015). Exploring online ad images using a deep convolutional neural network approach. *arXiv preprint arXiv:1509.00568*.

Gabrilovich, E. y Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th international joint conference on artificial intelligence*, volume 6, page 12.

Gainaru, A., Dumitrescu, S. D., y Trausan-Matu, S. (2010). Toolkit for automatic analysis of chat conversations. In *Communications (COMM), 2010 8th International Conference on*, pages 99–102. IEEE.

Galán-García, P., de la Puerta, J. G., Gómez, C. L., Santos, I., y Bringas, P. G. (2014). Supervised machine learning for the detection of troll profiles in twitter social network: Application to a real case of cyberbullying. In *International Joint Conference SOCO'13-CISIS'13-ICEUTE'13*, pages 419–428. Springer.

Gangemi, A., Nuzzolese, A. G., Presutti, V., Draicchio, F., Musetti, A., y Ciancarini, P. (2012). Automatic typing of dbpedia entities. In *The Semantic Web–ISWC 2012*, pages 65–81. Springer.

Garfield, E. y Merton, R. (1979). *Citation indexing: Its theory and application in science, technology, and humanities*, volume 8. Wiley New York.

Giovanni, A., Gangemi, A., Presutti, V., y Ciancarini, P. (2012). Type inference through the analysis of wikipedia links. *Linked Data on the Web (LDOW)*.

Goddard, C. (1998). *Semantic analysis*. Oxford University Press.

Golemati, M., Katifori, A., Vassilakis, C., Lepouras, G., y Halatsis, C. (2007). Creating an ontology for the user profile: Method and applications. In *Proceedings of the first RCIS conference*, pages 407–412.

Grefenstette, G. (1999). The world wide web as a resource for example-based machine translation tasks. In *Proceedings of the ASLIB Conference on Translating and the Computer*, volume 21.

Griffiths, A., Luckhurst, H., y Willett, P. (1997). Using interdocument similarity information in document retrieval systems. *Readings in Information Retrieval, Morgan Kaufmann Publishers, San Francisco, CA*, pages 365–373.

Habib, M. B. y Keulen, M. (2012). Unsupervised improvement of named entity extraction in short informal context using disambiguation clues.

Hadj Taieb, M. A., Ben Aouicha, M., y Ben Hamadou, A. (2013). Computing semantic relatedness using wikipedia features. *Knowledge-Based Systems*.

Harman, D. (1993). Overview of the first trec conference. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 36–47. ACM.

Haslhofer, B., Martins, F., y Magalhães, J. (2013). Using skos vocabularies for improving web search. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 1253–1258. International World Wide Web Conferences Steering Committee.

Hatzivassiloglou, V. y McKeown, K. R. (1997). Predicting the semantic orientation of adjectives. page 174–181.

Hearst, M. (1999). Untangling text data mining. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 3–10. Association for Computational Linguistics.

Heck, L. P., Konig, Y., Sönmez, M. K., y Weintraub, M. (2000). Robustness to telephone handset distortion in speaker recognition by discriminative feature design. *Speech Communication*, 31(2):181–192.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., y Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53.

Hidalgo, J. M. G. y Díaz, A. A. C. (2012). Combining predation heuristics and chat-like features in sexual predator identification. In *CLEF (Online Working Notes/-Labs/Workshop)*.

Hillard, D., Ostendorf, M., y Shriberg, E. (2003). Detection of agreement vs. disagreement in meetings: Training with unlabeled data. page 34–36.

Hirschman, L., Yeh, A., Blaschke, C., y Valencia, A. (2005). Overview of BioCreAtIvE: critical assessment of information extraction for biology. *BMC bioinformatics*, 6(Suppl 1):S1.

Hoffart, J., Fabian, M., Berberich, S. K., Weikum, G., Berberich, K., y Suchanek, F. M. (2010). Yago2: a spatially and temporally enhanced knowledge base from wikipedia. In *Commun. ACM*. Citeseer.

Hoffart, J., Suchanek, F. M., Berberich, K., Lewis-Kelham, E., De Melo, G., y Weikum, G. (2011). Yago2: exploring and querying world knowledge in time, space, context, and many languages. In *Proceedings of the 20th international conference companion on World wide web*, pages 229–232. ACM.

Hoffart, J., Suchanek, F. M., Berberich, K., y Weikum, G. (2013). Yago2: a spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61.

Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1):177–196.

Hong, L., Dom, B., Gurumurthy, S., y Tsioutsiouliklis, K. (2011). A time-dependent topic model for multiple text streams. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 832–840. ACM.

Hopkins, D. y King, G. (2007). Extracting systematic social science meaning from text. *Manuscript available at http://gking. harvard. edu/files/words. pdf*.

Hu, L., Wang, X., Zhang, M., Li, J., Li, X., Shao, C., Tang, J., y Liu, Y. Learning topic hierarchies for wikipedia categories. *Volume 2: Short Papers*, page 346.

Huang, P.-S., He, X., Gao, J., Deng, L., Acero, A., y Heck, L. (2013). Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 2333–2338. ACM.

Huang, X. D., Ariki, Y., y Jack, M. A. (1990). *Hidden Markov models for speech recognition*, volume 2004. Edinburgh university press Edinburgh.

Hui, S. C., He, Y., y Dong, H. (2008). Text mining for chat message analysis. In *Cybernetics and Intelligent Systems, 2008 IEEE Conference on*, pages 411–416. IEEE.

Jang, Y., Lee, T., Kim, K., Lee, W., Ann, D., y Chung, S. (2007). Keyword management system based on ontology for contextual advertising. In *Advanced Language*

Processing and Web Information Technology, 2007. ALPIT 2007. Sixth International Conference on, pages 440–445. IEEE.

Jansen, B. J. y Schuster, S. (2011). Bidding on the buying funnel for sponsored search and keyword advertising. *Journal of Electronic Commerce Research*, 12(1):1–18.

Jin, X., Li, Y., Mah, T., y Tong, J. (2007). Sensitive webpage classification for content advertising. page 28–33.

Jindal, V., Bawa, S., y Batra, S. (2015). A query-context oriented approach to semantic search on web. *International Journal of Artificial Intelligence and Knowledge Discovery*, 1(1):14–20.

Joachims, T. (1997). A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pages 143–151.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *Machine Learning: ECML-98*, pages 137–142.

Jones, G., Foote, J., Jones, K., y Young, S. (1996). Retrieving spoken documents by combining multiple index sources. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 30–38. ACM.

Kent, A. (1966). *Textbook on mechanized information retrieval*. Interscience.

Kent, A., Berry, M., Luehrs, F., y Perry, J. (2007). Machine literature searching viii. operational criteria for designing information retrieval systems. *American documentation*, 6(2):93–101.

Kenter, T. y de Rijke, M. (2015). Short text similarity with word embeddings. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1411–1420. ACM.

Kessler, B., Numberg, G., y Schütze, H. (1997). Automatic detection of text genre. page 32–38.

Khalili, A., Auer, S., y Ngomo, A.-C. N. (2014). context–lightweight text analytics using linked data. In *The Semantic Web: Trends and Challenges*, pages 628–643. Springer.

Khan, S., Ilyas, Q. M., y Anwar, W. (2009). Contextual advertising using keyword extraction through collocation. In *Proceedings of the 7th International Conference on Frontiers of Information Technology, FIT*, volume 9, page 69.

Khemmarat, S., Saha, S., Song, H. H., Baldi, M., y Gao, L. (2014). On understanding user interests through heterogeneous data sources. In *Passive and Active Measurement*, pages 272–274. Springer.

Kim, S., Banchs, R. E., y Li, H. A composite kernel approach for dialog topic tracking with structured domain knowledge from wikipedia.

Kim, S.-M. y Hovy, E. (2004). Determining the sentiment of opinions. In *Proceedings of the 20th international conference on Computational Linguistics*, page 1367. Association for Computational Linguistics.

Knoblock, C. A. (1997). Searching the world wide web. *IEEE EXPERT*, pages 8–14.

Kobayashi, N., Inui, T., y Inui, K. (2001). Dictionary-Based acquisition of the lexical knowledge for p/n analysis. *SIG SLUD*, 33:45–50.

Kobilarov, G., Scott, T., Raimond, Y., Oliver, S., Sizemore, C., Smethurst, M., Bizer, C., y Lee, R. (2009). Media meets semantic web–how the bbc uses dbpedia and linked data to make connections. In *The semantic web: research and applications*, pages 723–737. Springer.

Kolenda, T., Hansen, L. K., y Larsen, J. (2001). Signal detection using ica: application to chat room topic spotting. In *Third International Conference on Independent Component Analysis and Blind Source Separation*, pages 540–545.

Konig, Y., Heck, L., Weintraub, M., Sonmez, K., et al. (1998). Nonlinear discriminant feature extraction for robust text-independent speaker recognition. In *Proc. RLA2C, ESCA workshop on Speaker Recognition and its Commercial and Forensic Applications*, pages 72–75.

Koppel, M. y Schler, J. (2005a). Using neutral examples for learning polarity. In *Proceedings of International Joint Conferences on Artificial Intelligence (IJCAI)*, volume 2005, pages 1616–1616.

Koppel, M. y Schler, J. (2005b). Using neutral examples for learning polarity. volume 19, page 1616.

Koppel, M. y Schler, J. (2006a). The importance of neutral examples for learning sentiment. *Computational Intelligence*, 22(2):100–109.

Koppel, M. y Schler, J. (2006b). The importance of neutral examples for learning sentiment. *Computational Intelligence*, 22(2):100–109.

Kumar, A. S. y Singh, S. (2013). Detection of user cluster with suspicious activity in online social networking sites. In *Advanced Computing, Networking and Security (ADCONS), 2013 2nd International Conference on*, pages 220–225. IEEE.

Lacasta, J., Nogueras-Iso, J., López-Pellicer, F. J., Muro-Medrano, P. R., y Zarazaga-Soria, F. J. (2013). Thmanager: An open source tool for creating and visualizing skos. *Information Technology and Libraries*, 26(3):39–51.

Landauer, T. K., Foltz, P. W., y Laham, D. (1998). An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284.

Langheinrich, M., Nakamura, A., Abe, N., Kamba, T., y Koseki, Y. (1999). Unintrusive customization techniques for web advertising. *Computer Networks*, 31(11):1259–1272.

Laorden, C., Galán-García, P., Santos, I., Sanz, B., Nieves, J., Bringas, P. G., y Hidalgo, J. M. G. (2014). Negobot: Detecting paedophile activity with a conversational agent based on game theory. *Logic Journal of IGPL*, page jzu034.

Larkey, L. (1998). Automatic essay grading using text categorization techniques. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 90–95. ACM.

Laver, M., Benoit, K., y Garry, J. (2003). Extracting policy positions from political texts using words as data. *American Political Science Review*, 97(02):311–331.

Lee, J.-H., Ha, J., Jung, J.-Y., y Lee, S. (2013). Semantic contextual advertising based on the open directory project. *ACM Transactions on the Web (TWEB)*, 7(4):24.

Lee, L. (2003). ï'm sorry dave, i'm afraid i can't do that": Linguistics, statistics, and natural language processing circa 2001. *Arxiv preprint cs/0304027*.

Lehmann, J., Schüppel, J., y Auer, S. (2007). Discovering unknown connections-the dbpedia relationship finder. *CSSW*, 113:99–110.

Lehnert, W. G. (1977). A conceptual theory of question answering. In *Proceedings of the 5th international joint conference on Artificial intelligence-Volume 1*, pages 158–164. Morgan Kaufmann Publishers Inc.

Lenhart, A., Simon, M., y Graziano, M. (2001). The internet and education: Findings of the pew internet & american life project.

Lewis, D. (1992). *Representation and learning in information retrieval*. PhD thesis, University of Massachusetts.

Lewis, D. (1997). Reuters-21578 text categorization test collection. *AT&T Labs Research*.

Lewis, D., Schapire, R., Callan, J., y Papka, R. (1996). Training algorithms for linear text classifiers. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 298–306. ACM.

Lewis, D., Yang, Y., Rose, T., y Li, F. (2004). Rcv1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research*, 5:361–397.

Li, Y. y Jain, A. (1998). Classification of text documents. *The Computer Journal*, 41(8):537.

Lim, E.-P., Nguyen, V.-A., Jindal, N., Liu, B., y Lauw, H. W. (2010). Detecting product review spammers using rating behaviors. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 939–948. ACM.

Lin, D. (1998). Automatic retrieval and clustering of similar words. page 768–774.

Lin, F.-H. y Hung, Y.-F. (2009). The value of and attitude toward sponsored links for internet information searchers. *Journal of Electronic Commerce Research*, 10(4):235–251.

Liu, H., Lieberman, H., y Selker, T. (2003). A model of textual affect sensing using real-world knowledge. page 125–132.

Liu, J., Wang, C., Liu, Z., y Yao, W. (2010). Advertising keywords extraction from web pages. In *Web Information Systems and Mining*, pages 336–343. Springer.

Liu, P., Azimi, J., y Zhang, R. (2014a). Automatic keywords generation for contextual advertising. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pages 345–346. International World Wide Web Conferences Steering Committee.

Liu, T. y Guan, S. (2014). Factor analysis method for text-independent speaker identification. *Journal of Software*, 9(11):2851–2860.

Liu, X., Yu, Y., Guo, C., Sun, Y., y Gao, L. (2014b). Full-text based context-rich heterogeneous network mining approach for citation recommendation. In *Digital Libraries (JCDL), 2014 IEEE/ACM Joint Conference on*, pages 361–370. IEEE.

Luhn, H. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of research and development*, 1(4):309–317.

Ma, X., Carranza, E. J. M., Wu, C., Van Der Meer, F. D., y Liu, G. (2011). A skos-based multilingual thesaurus of geological time scale for interoperability of online geological maps. *Computers & Geosciences*, 37(10):1602–1615.

Mahajan, A. y Sharmistha, S. R. (2015). Feature selection for short text classification using wavelet packet transform. *CoNLL 2015*, page 321.

Mahdian, M. y Tomak, K. (2007). Pay-per-action model for online advertising. In *Proceedings of the 1st international workshop on Data mining and audience intelligence for advertising*, pages 1–6. ACM.

Maña López, M., de Buenaga Rodríguez, M., y Gómez Hidalgo, J. (2010). Using and evaluating user directed summaries to improve information access. *Research and Advanced Technology for Digital Libraries*, pages 852–852.

Marcus, M. P., Marcinkiewicz, M. A., y Santorini, B. (1993). Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Maron, M. y Kuhns, J. (1960). On relevance, probabilistic indexing and information retrieval. *Journal of the ACM (JACM)*, 7(3):216–244.

Mauldin, M. I. (1997). Lycos: Design choices in an internet search service. *IEEE Expert*, 12(1):8–11.

Mayr, P., Zapilko, B., y Sure, Y. (2010). Establishing a multi-thesauri-scenario based on skos and cross-concordances. *arXiv preprint arXiv:1009.5352*.

McCallumzy, A. y Nigamy, K. (1998). A comparison of event models for naive bayes text classification.

Mendes, P. N., Jakob, M., García-Silva, A., y Bizer, C. (2011). Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 1–8. ACM.

Mesnil, G., He, X., Deng, L., y Bengio, Y. (2013). Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *INTERSPEECH*, pages 3771–3775.

Mihalcea, R. y Csomai, A. (2007). Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242. ACM.

Mikolov, T., Chen, K., Corrado, G., y Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., y Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Mikolov, T., Yih, W.-t., y Zweig, G. (2013c). Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751.

Miles, A. y Bechhofer, S. (2009). Skos simple knowledge organization system reference. Technical report, Technical report, W3C.

Miller, G. A. y Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28.

Milne, D. y Witten, I. H. (2008). Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518. ACM.

Mishne, G. y Glance, N. (2006). Predicting movie sales from blogger sentiment.

Momjian, B. (2001). *PostgreSQL: introduction and concepts*, volume 192. Addison-Wesley.

Mooney, R. J. y Roy, L. (2000). Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 195–204. ACM.

Moro, A. y Navigli, R. (2012). Wisenet: building a wikipedia-based semantic network with ontologized relations. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1672–1676. ACM.

Moro, A., Raganato, A., y Navigli, R. (2014). Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

Morshed, A., Keizer, J., Johannsen, G., Stellato, A., y Baker, T. (2010). From agrovoc owl model towards agrovoc skos model.

Nandhini, B. y Sheeba, J. (2015a). Cyberbullying detection and classification using information retrieval algorithm. In *Proceedings of the 2015 International Conference on Advanced Research in Computer Science Engineering & Technology (ICARCSET 2015)*, page 20. ACM.

Nandhini, B. S. y Sheeba, J. (2015b). Online social network bullying detection using intelligence techniques. *Procedia Computer Science*, 45:485–492.

Nazerzadeh, H., Saberi, A., y Vohra, R. (2008). Dynamic cost-per-action mechanisms and applications to online advertising. In *Proceedings of the 17th international conference on World Wide Web*, pages 179–188. ACM.

Nguyen, T. H. y Shirai, K. (2015). Topic modeling based sentiment analysis on social media for stock market prediction. In *Proceedings of the 53rd Annural Meeting of the Association for Computational Linguistics*.

Nieto-Nieto, L. (2012). Evolución de los computadores: Textos para una exposición.

Nosrati, M., Karimi, R., Mohammadi, M., y Malekian, K. (2013). Internet marketing or modern advertising! how? why? *International Journal of Economy, Management and Social Sciences*, 2(3):56–63.

Özyurt, Ö. y Köse, C. (2010). Chat mining: Automatically determination of chat conversations' topic in turkish text based chat mediums. *Expert Systems with Applications*, 37(12):8705–8710.

Page, L., Brin, S., Motwani, R., y Winograd, T. (1999). The pagerank citation ranking: bringing order to the web.

Pak, A. N. y Chung, C.-W. (2010). A wikipedia matching approach to contextual advertising. *World Wide Web*, 13(3):251–274.

Pal, D., Mitra, M., y Bhattacharya, S. (2015). Exploring query categorisation for query expansion: A study. *arXiv preprint arXiv:1509.05567*.

Pang, B. y Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. page 271.

Pang, B., Lee, L., y Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

Pasca, M. y Harabagiu, S. (2001). High performance question/answering. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 366–374. ACM.

Pereira, F., Tishby, N., y Lee, L. (1993). Distributional clustering of english words. page 183–190.

Pierce, J. R. y Carroll, J. B. (1966). Language and machines: Computers in translation and linguistics.

Pons-Porrata, A., Berlanga-Llavori, R., y Ruiz-Shulcloper, J. (2007). Topic discovery based on text mining techniques. *Information processing & management*, 43(3):752–768.

Prussakov, E. (2007). A practical guide to affiliate marketing.

Puertas Sanz, E., Gómez Hidalgo, J., y Cortizo Pérez, J. (2008). Email Spam Filtering. *Advances in Computers*, 74:45–114.

Rabiner, L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Raghunathan, K. y Krawczyk, S. (2009). Cs224n: Investigating sms text normalization using statistical machine translation. *Department of Computer Science, Stanford University*.

RahmanMiah, M. W., Yearwood, J., y Kulkarni, S. (2011). Detection of child exploiting chats from a mixed chat dataset as a text classification task. In *Proceedings of the Australian Language Technology Association Workshop*, pages 157–165. Citeseer.

Rajaraman, A. y Ullman, J. D. (2011). *Mining of massive datasets*. Cambridge University Press.

Rauber, A. y Müller-Kögler, A. (2001). Integrating automatic genre analysis into digital libraries. page 1–10.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. *Arxiv preprint cmp-lg/9511007*.

Ribeiro-Neto, B., Cristo, M., Golgher, P. B., y Silva de Moura, E. (2005). Impedance coupling in content-targeted advertising. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 496–503. ACM.

Riloff, E., Wiebe, J., y Wilson, T. (2003). Learning subjective nouns using extraction pattern bootstrapping. page 25–32.

Robertson, S. (1977). The probability ranking principle in ir. *Journal of documentation*, 33(4):294–304.

Rocchio, J. (1971). Relevance feedback in information retrieval.

Rogers, E. M. (1995). Diffusion of innovations. *New York*.

Rojas Lopez, F., Lopez Arevalo, I., Pinto, D., y Sosa Sosa, V. J. (2015). Context expansion for domain-specific word sense disambiguation. *Latin America Transactions, IEEE (Revista IEEE America Latina)*, 13(3):784–789.

Rowley, J. (2008). Understanding digital content marketing. *Journal of marketing management*, 24(5-6):517–540.

Salton, G. (1971). The smart retrieval system—experiments in automatic document processing.

Salton, G. y McGill, M. (1983). Introduction to moderm information retrieval.

Salton, G. y Smith, M. (1989). On the application of syntactic methodologies in automatic text analysis. volume 23, page 137–150.

Salton, G., Wong, A., y Yang, C. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Sánchez, J. A. P. (2013). Iso-thes: ampliando skos a partir de la norma de tesauros iso 25964. *Anuario ThinkEPI*, (1):189–193.

Sanderson, M. (1994). Word sense disambiguation and information retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 142–151. Springer-Verlag New York, Inc.

Santos, I., De-La-PeñA-Sordo, J., Pastor-López, I., Galán-García, P., y Bringas, P. G. (2012). Automatic categorisation of comments in social news websites. *Expert Systems with Applications*, 39(18):13417–13425.

Santos, I., Laorden, C., Sanz, B., y Bringas, P. G. (2011). Enhanced topic-based vector space model for semantics-aware spam filtering. *Expert Systems With Applications*.

Santos, I., Miñambres-Marcos, I., Laorden, C., Galán-García, P., Santamaría-Ibirika, A., y Bringas, P. G. (2014). Twitter content-based spam filtering. In *International Joint Conference SOCO'13-CISIS'13-ICEUTE'13*, pages 449–458. Springer.

Santos, J., Martins, B., y Batista, D. S. (2013). Document analytics through entity resolution. In *Web Information Systems Engineering–WISE 2013*, pages 531–534. Springer.

Schandl, T. y Blumauer, A. (2010). Poolparty: Skos thesaurus management utilizing linked data. In *The Semantic Web: Research and Applications*, pages 421–425. Springer.

Schank, R. C., Goldman, N. M., Rieger III, C. J., y Riesbeck, C. (1973). Margie: Memory analysis response generation, and inference on english. In *IJCAI*, pages 255–261.

Schmitt, P., Skiera, B., y Van den Bulte, C. (2011). Referral programs and customer value. *Journal of Marketing*, 75(1):46–59.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

Shams, R. (2014). Semi-supervised classification for natural language processing. *arXiv preprint arXiv:1409.7612*.

Sherman, C. (2007). The state of search engine marketing 2006. *Retrieved October*, 25.

Shulman, S., Callan, J., Hovy, E., y Zavestoski, S. (2005). Language processing technologies for electronic rulemaking: A project highlight. page 87–88.

Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., y Chanona-Hernández, L. (2014). Syntactic n-grams as machine learning features for natural language processing. *Expert Systems with Applications*, 41(3):853–860.

Smiley, D. y Pugh, E. (2009). *Solr 1.4 Enterprise Search Server*. Packt Pub Limited.

Smith, L. (1976). Review of information retrieval and processing by lauren b. doyle; los angeles, melville publishing company, 1975. In *ACM SIGIR Forum*, volume 11, pages 7–9. ACM.

Sparck Jones, K. (1971). *Automatic keyword classification for information retrieval*, volume 253. London: Butterworths.

Spertus, E. (1997). Smokey: Automatic recognition of hostile messages. page 1058–1065.

Sridhar, V. K. R., Chen, J., Bangalore, S., y Shacham, R. (2014). A framework for translating sms messages. In *Proceedings of COLING*, pages 974–983.

Strube, M. y Ponzetto, S. P. (2006). Wikirelate! computing semantic relatedness using wikipedia. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 1419. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Strzalkowski, T., Guthrie, L., Karlgren, J., Leistensnider, J., Lin, F., Perez-Carballo, J., Straszheim, T., Wang, J., y Wilding, J. (1996). Natural language information retrieval: Trec-5 report. In *Proceedings of the 5th Text Retrieval Conference (TREC-5)*.

Subasic, P. y Huettner, A. (2001). Affect analysis of text using fuzzy semantic typing. *Fuzzy Systems, IEEE Transactions on*, 9(4):483–496.

Suchanek, F. M., Kasneci, G., y Weikum, G. (2007). Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM.

Sugiyama, K., Hatano, K., y Yoshikawa, M. (2004). Adaptive web search based on user profile constructed without any effort from users. In *Proceedings of the 13th international conference on World Wide Web*, pages 675–684. ACM.

Sullivan, D. (2010). Does sem= seo+ cpc still add up? *Online unter URL: http://searchengineland. com/does-sem-seo-cpc-still-add-up-37297 Abfrage*, 22:2013.

Sussna, M. (1993). Word sense disambiguation for free-text indexing using a massive semantic network. In *Proceedings of the second international conference on Information and knowledge management*, pages 67–74. ACM.

Tacoa, F., Bollegala, D., y Ishizuka, M. (2012). A context expansion method for supervised word sense disambiguation. In *Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on*, pages 339–341. IEEE.

Takamura, H., Inui, T., y Okumura, M. (2005). Extracting semantic orientations of words using spin model. page 133–140.

Tatemura, J. (2000). Virtual reviewers for collaborative exploration of movie reviews. page 272–275.

Terveen, L., Hill, W., Amento, B., McDonald, D., y Creter, J. (1997). PHOAKS: a system for sharing recommendations. *Communications of the ACM*, 40(3):59–62.

Trattner, C. y Kappe, F. (2013). Social stream marketing on facebook: a case study. *International Journal of Social and Humanistic Computing*, 2(1):86–103.

Trefethen, L. N. y Bau III, D. (1997). *Numerical linear algebra*. Number 50. Society for Industrial Mathematics.

Tsur, O., Davidov, D., y Rappoport, A. (2010). Icwsm-a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *ICWSM*.

Tuominen, J., Frosterus, M., Viljanen, K., y Hyvönen, E. (2009). Onki skos server for publishing and utilizing skos vocabularies and ontologies as services. In *The Semantic Web: Research and Applications*, pages 768–780. Springer.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, pages 433–460.

Turney, P. (2002). Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews.

Turney, P. y Littman, M. L. (2010). Unsupervised learning of semantic orientation from a hundred-billion-word corpus.

Turney, P. D. (2000). Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4):303–336.

Turtle, H. y Croft, W. (1989). Inference networks for document retrieval. In *Proceedings of the 13th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 1–24. ACM.

Uzuner, O., Szolovits, P., y Kohane, I. (2006). i2b2 workshop on natural language processing challenges for clinical records. In *Proceedings of the Fall Symposium of the American Medical Informatics Association*.

Vaquero Pulido, J. R. (1997). Recuperación de la información en internet: motores y otros agentes de búsqueda. *Scire: representación y organización del conocimiento*, 3(2):85–100.

Völkel, M., Krötzsch, M., Vrandecic, D., Haller, H., y Studer, R. (2006). Semantic wikipedia. In *Proceedings of the 15th international conference on World Wide Web*, pages 585–594. ACM.

Walmsley, W. S., Snelgrove, W. X., y Truong, K. N. (2014). Disambiguation of imprecise input with one-dimensional rotational text entry. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 21(1):4.

Wang, R., Sang, N., y Gao, C. (2015). Text detection approach based on confidence map and context information. *Neurocomputing*.

Weizenbaum, J. (1966). ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45.

Widenius, M. y Axmark, D. (2002). *MySQL reference manual: documentation from the source*. O'Reilly Media, Inc.

Wiebe, J. M. (1994a). Tracking point of view in narrative. *Computational Linguistics*, 20(2):233–287.

Wiebe, J. M. (1994b). Tracking point of view in narrative. *Computational Linguistics*, 20(2):233–287.

Wilbur, W. y Sirotkin, K. (1992). The automatic identification of stop words. *Journal of information science*, 18(1):45–55.

Wilensky, R. (1978). Understanding goal-based stories. Technical report, DTIC Document.

Willett, P. (2006). The porter stemming algorithm: then and now. *Program*, 40(3):219–223.

Wilson, J. (1990). *Politically speaking: The pragmatic analysis of political language*. Basil Blackwell.

Winograd, T. (1980). What does it mean to understand language? *Cognitive science*, 4(3):209–241.

Witten, I. y Milne, D. (2008). An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA*, pages 25–30.

Xu, J. y Croft, W. B. (1996). Query expansion using local and global document analysis. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 4–11. ACM.

Yang, Y. y Liu, X. (1999). A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49. ACM.

Yang, Y. y Pedersen, J. (1997). A comparative study on feature selection in text categorization. In *Machine Learning-International Workshop then Conference*, pages 412–420.

Yih, W.-t., Goodman, J., y Carvalho, V. R. (2006). Finding advertising keywords on web pages. In *Proceedings of the 15th international conference on World Wide Web*, pages 213–222. ACM.

Yu, H. y Hatzivassiloglou, V. (2003). Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. page 129–136.

Yu, Z., Zhou, X., Hao, Y., y Gu, J. (2006). Tv program recommendation for multiple viewers based on user profile merging. *User Modeling and User-Adapted Interaction*, 16(1):63–82.

Zeng, D., Liu, K., Chen, Y., y Zhao, J. (2015). Distant supervision for relation extraction via piecewise convolutional neural networks. EMNLP.

Zhang, D., Xu, H., Su, Z., y Xu, Y. (2015a). Chinese comments sentiment classification based on word2vec and svm perf. *Expert Systems with Applications*, 42(4):1857–1863.

Zhang, H., Wang, C.-D., y Lai, J.-H. (2014). Topic detection in instant messages. In *Machine Learning and Applications (ICMLA), 2014 13th International Conference on*, pages 219–224. IEEE.

Zhang, M., Zhang, Y., y Vo, D.-T. (2015b). Neural networks for open domain targeted sentiment. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*.

Zhang, Y., Xu, W., y Callan, J. (2002). Exact maximum likelihood estimation for word mixtures. In *Text Learning Workshop in International Conference on Machine Learning (ICML2002)*. Citeseer.

Zhao, S., Li, C., Ma, S., Ma, T., y Ma, D. (2013). Combining pos tagging, lucene search and similarity metrics for entity linking. In *Web Information Systems Engineering–WISE 2013*, pages 503–509. Springer.