Proyecto Fin de Carrera

Ingeniería en Informática

Mining Semantic Relations from Product Features

Autor

Juan José Turmo Gutiérrez

Abril 2012

Director (Tutor)

Wolf-Tilo Balke (**Silviu Homoceanu**)

Technische Universität Braunschweig

Ponente

Eduardo Mena Nieto

Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Nuestra recompensa se encuentra
en el esfuerzo y no en el resultado.
Un esfuerzo total es una victoria completa
Mahatma Gandhi

Cuando un hombre planta árboles
bajo los cuales sabe muy bien que nunca se sentará,
ha empezado a descubrir el significado de la vida.
Elton Trueblood

Agradecimientos

A mis *tutores*, Tilo, quien me dio la oportunidad de colaborar en su departamento con sus investigaciones; Silviu, más que tutor, compañero de investigación que me orientó de la mejor forma que pudo mientras yo lo entendía también de la manera más acertada posible; y Eduardo, quien aunque desde su rol de ponente, siempre ha estado y espero que esté ahí para cualquier cosa, como yo lo espero estar por él y no perder la relación.

A mis amigos del Erasmus vivido Braunschweig (Alemania), donde quien más quien menos, formó parte de este proyecto, sólo por el hecho de estar ahí, de escucharme cuando lo necesitaba, de ayudarme a comprender mi propio proyecto y sobre todo de darme ánimos y de convertirse en mi familia durante el maravilloso año que disfruté junto a ellos.

A mis amigos de Barbastro, Zaragoza y alrededores, a mis compañeros de trabajo y a mi familia, que aunque no pudieron estar en el *terreno de juego* durante mi estancia Erasmus, pudieron apoyarme y animarme desde el *banquillo* a través de Internet, y durante la *prórroga* que he vivido en los últimos meses en España, en los que más de una vez he necesitado un empujón para poder salir adelante con el trabajo.

Y en especial, a María José y Mariano, mis padres, quienes aunque no lo quiera creer, me conocen mejor que nadie, y cuyo vínculo con mi ánimo y sufrimiento estuvo siempre ahí, a las duras y a las maduras, de cerca y de lejos, durante todos estos años de carrera que tan rápido han pasado, y que finalmente concluyen con el desarrollo de este proyecto, que bien saben que no ha sido un trabajo fácil, y que sin su apoyo incondicional, no habría salido adelante. Más que agradecérselo, a ellos se lo dedico.

Resumen

Mientras Internet se colaba en los hogares de millones de personas y el comercio electrónico se convertía en una práctica habitual en nuestra sociedad, surgió la capacidad de comentar acerca de los servicios o productos adquiridos para contrastar diversas opiniones antes de realizar una compra. Motivados por este hecho, los mismos portales de venta on-line, foros, comunidades, etc., se dedicaron a recoger esos comentarios, opiniones o revisiones, y agruparlos por productos, tipos, modelos, etc. Sin embargo, tal avalancha de datos no servía de nada sin un orden que los hiciera útiles. De esta situación surgió la necesidad de tratar de alguna manera estas grandes cantidades de información para que no se fueran acumulando sin un objetivo concreto. Desde el punto de vista de los consumidores, es necesaria la extracción de información relevante, la cual les ayude a decidirse por un producto u otro; mientras que para los vendedores, lo es la realización de un seguimiento de las opiniones relativas a sus productos, el cual puede hacerles mejorar su posición en el mercado. En base a éste escenario, se han planteado diversos retos en la comunidad investigadora, donde ha surgido todo tipo de corrientes para obtener cualquier tipo de beneficio a partir de los comentarios de entre ellos la minería de opiniones, la construcción de ontologías, el aprendizaje automático, etc.

Es aquí donde se enmarca la investigación realizada en este proyecto fin de carrera, desarrollado dentro del Departamento de Sistemas de Información (IfIS[1]) de la Universidad Técnica de Braunschweig[2] (Alemania) bajo la dirección de Silviu Homoceanu y la supervisión de Wolf-Tilo Balke. En concreto, la investigación del proyecto se basa en las revisiones o los comentarios elaborados, extraídos de un portal web real acerca de teléfonos inteligentes, los cuales suelen estar acompañados de sus respectivas especificaciones técnicas. Son éstas el comienzo a partir del cual, se extraen los términos iniciales que suelen girar en torno a un aspecto específico de dichos teléfonos, como puede ser la batería, que forman los grupos semánticos iniciales. La clave del proceso consiste en utilizar técnicas clásicas de recuperación de información ayudadas de estos dos tipos de recursos, como son los comentarios y las especificaciones técnicas. Este proyecto fin de carrera tiene como objetivo la extracción de relaciones semánticas entre los términos iniciales y los que aparecen en los comentarios mencionados. El comportamiento de éste tipo de técnicas en diferentes escenarios generados a partir de la colección de comentarios utilizada es analizado con todo tipo de detalle, donde el método de Latent Dirichlet Allocation (LDA) se presenta como la técnica que mejor se adapta al objetivo propuesto, demostrando la madurez estas técnicas siendo una actual fuente de inspiración para nuevas investigaciones.

[1] www.ifis.cs.tu-bs.de (Institut für Informationssysteme)
[2] www.tu-braunschweig.de (TU Braunschweig)

Contenido

1. Introducción

Este documento describe el Proyecto Fin de Carrera titulado "Mining semantic relations from product features", desarrollado en el Departamento de Sistemas de Información (IfIS) de la Universidad Técnica de Braunschweig (Alemania). En este Capítulo se describen los conceptos principales que hacen referencia a la descripción del problema y la motivación, el estado del arte, y la distribución de los Capítulos de este proyecto. En el **Anexo 1** se detalla en mayor grado la versión inglesa del proyecto.

1.1. Descripción del problema y motivación

Debido a que la popularidad del comercio electrónico se está incrementando, el número de comentarios o revisiones, en sitios web, blogs, comunidades y foros, que recibe un producto o servicio, también lo hace. Mientras algunos productos reciben algunos comentarios, otros reciben cientos. "Cuanta más información, mejor", ha sido el eslogan de la era de la información [1]. No obstante, si se quiere obtener algún beneficio de esta información de manera manual, la tarea es casi irrealizable. Esto dificulta la posible decisión de compra por parte de un cliente, al que le es casi imposible obtener alguna conclusión. Además, dificulta el seguimiento de las opiniones de los clientes y su acogida en el mercado por parte de los vendedores. Este problema se ve agravado dado que muchos portales suelen vender el mismo producto, además de que un vendedor suele suministrar cantidad de productos [2].

Por otro lado, los consumidores no aportan sus comentarios únicamente dando su visión objetiva del producto, sino que manifiestan sus sentimientos e impresiones, alabando o criticando las características del mismo. Escribir acerca de un producto suele ser muy distinto tanto para vendedores como consumidores, y entre ellos también, dado que los vendedores habitualmente aportan detalles técnicos, mientras que son los consumidores los que emplean un lenguaje más natural, incluso hasta crean su propio vocabulario.

Basándose en este escenario, muchos investigadores han centrado sus esfuerzos en diferentes tareas como la minería de opiniones, la extracción y clasificación de características de productos, el resumen de comentarios, la medida de la similitud semántica entre características o la construcción de ontologías. Sin embargo, no hay muchos esfuerzos centrados en el agrupamiento de términos o características semánticamente relacionadas, por lo que ésa es la motivación de este proyecto. Ésto ha sido posible gracias a la labor previa desarrollada por el Departamento de Sistemas de Información (IfIS[3]) de la Universidad Técnica

[3] www.ifis.cs.tu-bs.de (Institut für Informationssysteme)

de Braunschweig[4], donde durante la estancia Erasmus en Alemania y bajo la tutela de Silviu Homoceanu y la supervisión de Wolf-Tilo Balke, se facilitó el acceso a una colección de comentarios reales previamente extraídos del portal Phonearena[5]. Dicha colección ha constituido el marco del desarrollo de este proyecto, el cual se ha centrado en los comentarios que consumidores y expertos han realizado sobre los teléfonos inteligentes.

Principalmente, se han considerado los hechos tales como a) que las especificaciones técnicas, en este caso de los teléfonos inteligentes y en general de cualquier producto, vienen agrupadas por características principales de producto que engloban a una serie de términos que se suponen semánticamente relacionados; y b) que en la mayoría de los casos los comentarios o revisiones vienen acompañados por estas especificaciones técnicas, como así lo hacen los proporcionados por este departamento. Por esto, las especificaciones técnicas son consideradas como recursos para este proyecto, al igual que los comentarios. Entonces, se propone en este proyecto que, a partir de los grupos semánticos formados por las especificaciones técnicas, se proceda a la obtención de nuevas relaciones semánticas entre los términos iniciales y otros términos nuevos de los comentarios semánticamente relacionados con los primeros. Para llevar a cabo el objetivo, se ha diseñado un proceso manual y otro automático, con la intención de poder comprobar las diferencias entre ambos, a pesar de que el proceso manual se hace sobre una parte simbólica del entorno. Dentro del proceso automático, se pretende utilizar las técnicas clásicas de recuperación de información, como son *Latent Semantic Indexing* (LSI) [3], *Probabilistic Latent Semantic Indexing* (PLSI) [4] y *Latent Dirichlet Allocation* (LDA) [5], que muestran en sí mismas la evolución de este tipo de métodos.

Por ejemplo, en la parte izquierda de la **Figura 1-1**, *multimedia*, *music player*, *video playback* y *mpeg4* son algunos de esos términos que se suponen semánticamente relacionados *a priori* y que juntos forman un grupo semántico de términos semánticamente relacionados entre sí, mientras que *multimedia* es una de las características principales antes mencionadas. A partir de estos términos que se denominan como los términos iniciales relativos a *multimedia*, se procedería a buscar los términos semánticamente relacionados dentro de los comentarios, como en el fragmento que se muestra en la parte derecha de la **Figura 1-1**. Ahí se puede observar cómo tanto los sustantivos y los sustantivos compuestos están subrayados, lo que son términos que pueden estar semánticamente relacionados con los términos de *multimedia*. Una vez extraídos estos términos, se procedería a evaluar su correcta extracción, es decir, si están o no semánticamente relacionados, de manera que se puede que términos como *concept*, *performance* o *surprise* tienen una relación muy débil o nula con el campo semántico descrito que abarcaría *multimedia*, mientras que *resolution*, *pixel* o *codec* son términos directa y semánticamente relacionados con los términos iniciales que se muestran en las especificaciones técnicas. Son estos últimos términos los que completarían la red semántica formada por los términos iniciales. No se tiene en cuenta el tipo de relación o la nomenclatura de la misma, como se haría en una ontología, sino que se trata de ampliar la familia semántica inicial descubriendo nuevas relaciones entre términos dentro de los comentarios.

[4] www.tu-braunschweig.de (TU Braunschweig)
[5] www.phonearena.com

[...] The LG Optimus GT540 is a strange concept when it comes to its multimedia playback capabilities. You can flawlessly play Xvid files with resolution of up to 640x360 pixels, and you can even go beyond that resolution, but the performance will be getting worse with each additional pixel.

Now for the surprise: you cannot play MPEG-4, DivX or H.264 videos on the Optimus. At least we couldn't run our tests files, which have always worked on other Android devices (we mean, the MPEG-4 ones). So there you have it - a nice Xvid-capable Android smartphone, crippled by its inability to play other popular codecs. [...]

Figura 1-1. Términos de *multimedia* encontrados en especificaciones técnicas y nuevos términos descubiertos en un fragmento de un comentario.

Resultados como los obtenidos en este proyecto pueden ayudar y mejorar a los procesos de minería de datos u opiniones, la representación del conocimiento, los procesos de aprendizaje automático, la mejora de los procesos de recuperación de información a través de la expansión de la consulta, etc. Mientras que el ser humano adquiere el conocimiento ayudado por las experiencias, emociones y sentimientos, un proceso automático no posee ninguno de estos contextos, sólo el texto plano, a partir del cual se consiguen impresionantes resultados, pero todavía hay un largo camino que recorrer. En este proyecto explicamos cómo la misma tarea puede ser llevada a cabo tanto manualmente como a través de herramientas automáticas, como son las técnicas de recuperación de información. Éstas han sido utilizadas durante años para otro tipo de propósitos, pero que encajan perfectamente en el marco anteriormente descrito, debido a que en cualquier motor de búsqueda actual se puede comprobar el exitoso tratamiento de enormes cantidades documentos en lenguaje natural.

1.2. Trabajo relacionado

Hay muchos estudios relacionados con las distintas partes del proceso automático. La mayoría de ellos están relacionados con la extracción de términos o características de producto. Sin embargo, existen diferentes investigaciones que se centran tanto en la obtención de la similitud semántica entre términos u ontologías, como en la agrupación y clasificación de características de producto, entre otras. A continuación se comenta una breve descripción de las mismas, que se detalla en el Capítulo **Related work** (ver **Anexo 1**).

1.2.1. Extracción de las características de producto

En primer lugar, es necesario aclarar qué se entiende por característica de producto o término. En trabajos anteriores [6], la palabra *característica* (*feature*) ha sido usada para referirse a los aspectos de un producto sobre los que los usuarios expresan su opinión, también llamadas *características de opinión* (*opinion features*) [7]. En este caso, una *característica de producto* (*product feature*) es un atributo, componente u otro aspecto del producto representado por un *nombre simple* o *compuesto*, también llamado *término*.

El proceso de extracción de características ha sido incluido habitualmente en el proceso de minería de opiniones, incluso cuando han sido tratados como procesos separados, éste último se ha abastecido de los resultados del primero. Para llevar a cabo el proceso de extracción, comúnmente se han utilizado las técnicas de etiquetado gramatical (*part-of-speech tagging*) como en los trabajos [2, 6-15], necesario para identificar y extraer los nombres simples y compuestos, como en las investigaciones de [10-12], donde además se establece un umbral de ocurrencia también utilizado en este proyecto. Además, en los estudios [2, 6, 8, 10, 11] se ha utilizado el proceso de lematización, de la misma forma que se aplica en nuestro proceso. Teniendo en cuenta qué técnicas han utilizado diversos investigadores para la extracción de características de producto, éstas están divididas en diferentes puntos de vista: minería de datos, que es la más usada, pero también algoritmos basados en probabilidades, modelos de entropía máxima, métodos de similitud de características y re-ocurrencia. Sin embargo, en este proyecto no se requiere tanta complejidad para extraer los nombres de los comentarios, entonces, se ha optado por el etiquetado gramatical y la lematización.

1.2.2. Similitud semántica

En principio, es necesario describir qué se entiende por relación semántica. En [16] se afirma que los conceptos y relaciones son el fundamento del conocimiento y del pensamiento. La variedad de las relaciones semánticas y sus propiedades juegan un rol importante en la comprensión y el razonamiento. Además de en palabras, las relaciones semánticas pueden ocurrir en altos niveles del texto – entre expresiones, frases, oraciones y fragmentos de texto mayores, así como entre documentos y conjuntos de documentos. Sin embargo, es necesario aclarar que este proyecto se centra la extracción de relaciones semánticas entre palabras o expresiones (nombres compuestos, términos), donde se dan lugar una gran cantidad de tipos de relación semántica más propias de ontologías, como son los pares hiponimia-hiperonimia (gato- animal), meronimia-holonimia (dedo-mano), y las más comunes de sinonimia y antonimia, a excepción de la troponimia, dado que no se tratan los verbos. No obstante, el objetivo de este estudio consiste en la extracción de relaciones semánticas, sin distinciones, por eso no se tiene en cuenta la clasificación que pueda existir. Los psicólogos consideran la importancia de las relaciones semánticas para explicar la coherencia y la estructura de los conceptos en categorías, donde cada categoría no es un conjunto aleatorio de entidades, sino que éstas pertenecen a la misma categoría porque juntas tienen sentido [16].

El método más común para saber cómo están relacionados dos términos consiste en utilizar una medida que determine la semejanza entre los significados de ambos. Los motores de búsqueda disponen de eficientes herramientas como son el recuento de páginas y la recuperación de fragmentos de texto. Estos fragmentos que devuelve un motor de búsqueda han sido utilizados en [17] para realizar comparaciones entre términos, mientras que [18] utiliza los fragmentos devueltos por Wikipedia concretamente. Además, los trabajos como [19] combinan los fragmentos con una serie de coeficientes como los de Jaccard, Dice y Overlap, así como el algoritmo de Información Mutua introducido previamente en [20], que calculan la similitud semántica entre dos términos basándose en el recuento de páginas. En [18] también se utiliza de la misma manera la similitud del coseno. De otra manera, en [21] se proponen dos medidas basadas en el contexto de los términos, ancho y estrecho, tomando el documento

completo o los términos que rodean a la palabra en cuestión, todo ello aludiendo a la suposición de que contextos similares implican significados similares. También existen otros tipos de medida, como la similitud de Google [22] basada en la complejidad de Kolmogorov, o como una visión probabilística desarrollada en [23]. En este proyecto se tienen en cuenta conceptos de medidas, contextos y en mayor medida modelos probabilísticos. Dado que sólo se tiene una colección de comentarios y sus especificaciones técnicas, el uso de cualquier medida relacionada con motores de búsqueda, Wikipedia, Google, etc., queda fuera de nuestro alcance, pero no lo están aquellos coeficientes que pueden medir la similitud entre vectores y que son empleados en uno de los métodos propios propuestos (ver Sección **3.6**).

Sin embargo, también existen métodos para definir la similitud semántica que consisten en agrupar términos o construir ontologías, aunque éstas no se utilizan aquí, se tiene en cuenta su existencia. Algunos investigadores han observado que los usuarios comentan sobre las diferentes características de productos, pero llega un punto en el que las palabras usadas convergen, entre ellos los métodos PLSI [4] y LDA [5], que introdujeron el modelado de temas (explicado en el Capítulo **3**) y han servido de inspiración para muchos otros. En [24] se propuso un método que aboga por la búsqueda de temas globales y locales. En [14] se propone una categorización sin supervisión de características de productos con múltiples niveles de asociación semántica. En [25] se combinan la agrupación de términos con la utilización de las medidas de similitud semántica. Finalmente, existen trabajos relacionados [26-32] donde se utilizan herramientas como WordNet [33] o ConceptNet [34], y se trata de encontrar relaciones semánticas, pero en el con el contexto de las ontologías. En conclusión, lo que se ha utilizado para llevar a cabo el objetivo consiste en utilizar técnicas de recuperación de información como son LSI, PLSI, y LDA, que no necesitan de recursos externos que ayuden a medir la semántica, y que modelan los datos distribuyéndolos en categorías, dimensiones o temas, a través de vectores, donde los términos son puntuados de una determinada forma.

1.3. Distribución de los capítulos

La distribución de los Capítulos es la siguiente: el Capítulo 2 detalla el contexto del análisis; el Capítulo 3 describe los conceptos teóricos de los métodos analizados; el Capítulo 4 agrupa todos los análisis realizados sobre los métodos en los distintos escenarios; el Capítulo 5 resume los resultados conseguidos; el Capítulo 6 presenta las conclusiones obtenidas; y en los capítulos siguientes se recogen la Bibliografía utilizada y el Anexo 1, que incluye el proyecto desarrollado en la TU Braunschweig, en cuyos Anexos se muestran tablas y gráficas que aportan más detalle al análisis desarrollado en este proyecto fin de carrera.

Comentarios iniciales, especificaciones técnicas, tablas, gráficas y resultados se almacenan en el DVD adjunto a este proyecto. Implementaciones de métodos utilizadas (LSI, PLSI y LDA) y desarrolladas (similitud de contextos y coocurrencia y CLDA), y programas utilizados y desarrollados necesarios para el procesado de los comentarios se recogen también en el DVD. Para poder ejecutarlos correctamente todos ellos están distribuidos ordenadamente en carpetas con su consiguiente descripción y manual de uso.

2. Contexto del análisis

En este Capítulo se describe el entorno en el que se desarrolla el presente proyecto fin de carrera. El Capítulo está dividido en secciones que hacen referencia a los tipos de comentarios, a los tipos de especificaciones técnicas y a cómo se pasa de éstos dos recursos a la construcción de una matriz de contextos.

El principal problema consiste en que ningún método matemático trabaja directamente sobre comentarios escritos en lenguaje natural, o sobre fragmentos de los mismos, sino que este tipo de datos necesita de un procesamiento previo [35]. En el Capítulo **Context of analysis** (ver **Anexo 1**) se detalla más ampliamente el contexto del análisis.

2.1. Tipo de comentarios

En los últimos años, la cantidad de comentarios se ha incrementado muy rápidamente [14], sin embargo, no todos ellos se analizan aquí, sino que se ha tomado una muestra real representativa para poder llevar a cabo los objetivos del proyecto. El estudio está limitado a la colección de 542 comentarios acerca de teléfonos móviles inteligentes, previamente extraídos del portal web de Phonearena por el IfIS de la TU Braunschweig (Alemania).

En primer lugar, hace falta determinar que lo que se entiende por comentarios o revisiones son aquellas evaluaciones escritas en lenguaje natural donde se expresa la opinión o el punto de vista que un consumidor se ha formado acerca de un producto o servicio específico. En función de dónde esté localizado un comentario, en un foro, sitio web, etc., puede venir acompañado de vídeos, fotos, especificaciones, etc., pero teniendo en cuenta el texto escrito, los comentarios pueden ser de tres tipos [8]. El primer formato (1) – Pros y Contras: Al usuario se le pide que resuma los pros y los contras por separado. El segundo formato (2) – Pros, contras y una revisión detallada: Al usuario se le pide que resuma los pros y contras por separado, y además que incluya una descripción detallada de sus impresiones. Por último, el tercer formato (3) – Formato libre: El usuario puede escribir libremente sin ninguna restricción de pros y contras. Además, dentro de esta libertad, los usuarios suelen delimitar secciones como “Introducción y diseño”, “Cámara y multimedia” y “Rendimiento”, entre otras. Es por esto que se ha tomado la decisión de no sólo tratar los comentarios como documentos completos, sino dividirlos en estas secciones y también en párrafos. Decisión basada en las etiquetas HTML o XML que componen las páginas que alojan estos comentarios y que un proceso automático podría recolectar, y que fue gratamente acogida por el equipo de investigación. Además de la inclusión de *comentarios extra* que se detallan en la sección de **2.3.3**.

2.2. Datos estructurados

Acompañando a los comentarios es común ver vídeos, fotos, enlaces, etc., pero normalmente, cuando se quiere concretar el producto o servicio que se quiere comentar para que no haya ambigüedades, éste se describe a través de las especificaciones que detallan a bajo nivel qué se está comentando realmente. En este caso en el que se han tomado como muestra los comentarios relativos a los teléfonos inteligentes, las especificaciones son técnicas, y en ellas se detallan en mayor o menor grado las características que hacen a ese dispositivo más diferente o parecido a los demás. De esta manera, se tiene que las especificaciones técnicas pueden no ser siempre las mismas, pero en muchos casos sí que comparten contenido entre ellas.

Figura 2-1. Ejemplo de especificaciones técnicas relativas a m*ultimedia*.

Se da por hecho que cada teléfono inteligente tiene sus propias especificaciones, como es el caso del portal web Phonearena, no se ha procedido a la extracción de todas y cada una de ellas, sino que la investigación se ha centrado en torno a tres características principales sobre las que se realiza todo el estudio, como son *battery*, *organizer* y *multimedia*. De la colección de 542 comentarios, se han seleccionado aleatoriamente aquellas especificaciones técnicas asociadas a éstos de las que se extraen manualmente los términos iniciales, dado que en la mayoría de los casos se compartían estos términos. En la **Figura 2-1** se observa un pequeño ejemplo de lo que se considera como característica principal, en este caso *multimedia*, alrededor de la cual se pueden observar términos como *music player*, *mp3* o *audio*, entre otros, que forman el grupo semántico relativo a *multimedia.* En este caso, se han tratado como nombres compuestos los términos como *music player*, sin tener en cuenta *music* o *player* por separado, porque muchas veces el significado del nombre compuesto no es la suma de los dos o más que lo componen, como sucede en este caso, dado que la palabra *player* puede tener muchos significados diferentes distintos completamente.

Las especificaciones técnicas, junto con los comentarios, forman los puntos de partida de esta investigación, porque la mezcla de ambos y la aplicación de técnicas de recuperación de la información, es la base de la búsqueda de nuevos términos semánticamente relacionados con los primeros.

2.3. Procesamiento de Lenguaje Natural

El procesamiento de lenguaje natural es el campo de la informática, la inteligencia artificial y la lingüística que concierne aquellas interacciones entre los ordenadores y los lenguajes, en este caso en inglés, en resumen, la forma que tiene un ordenador de *entender* el lenguaje escrito expresado por las personas. En esta investigación, se han empleado dos técnicas de procesamiento: el etiquetado gramatical (*POS tagging*), para poder seleccionar los nombres simples y compuestos; y la lematización (*stemming*) o extracción de la raíz de las palabras, para poder contabilizar conjuntamente los plurales, géneros y desinencias que hacen referencia a un único concepto semántico. Todo este procesamiento es necesario para *traducir* los comentarios y especificaciones técnicas, y hacerlas legibles para sistemas automáticos como son las técnicas de recuperación de información probadas.

2.3.1. Etiquetado gramatical

En primer lugar el etiquetado gramatical consiste en asignar el rol gramático, verbo, sustantivo, pronombre, adjetivo, adverbio, etc., que juega cada palabra dentro de cada oración [15]. En la Sección **1.2** se comenta el uso de esta técnica en diversos estudios. En [2, 6, 8, 12] se utiliza el analizador lingüístico NLProcessor[6] cuyo resultado se muestra en un fichero XML. En [10, 11] se utiliza MontyLingua[7] mientras que en [7] se opta por TreeTager[8]. El Standford POS Tagger[9], también usado en [15], es uno de los más utilizados en la actualidad y por ello se ha probado su funcionamiento en la colección de comentarios y especificaciones. Sin embargo, pese a que tiene varias modalidades de ejecución, presentaba problemas al identificar *mp3* como adjetivo o al reducir *aac+* en *aac* y tratarlos como adjetivos también. CRFTagger[10] también fue testado, pero se encontraron anomalías similares, como el tratamiento de *eaac* como adjetivo, *3g2* como número cardinal o *3g* como interjección. Finalmente, el etiquetador de OpenNLP[11] fue el escogido debido a que su rango de criba era más amplio y detectaba como sustantivos aquellas palabras que por su posición parecían serlo, aunque no las tuviera registradas previamente como tales. El paquete ofrece la posibilidad de combinar el detector de oraciones, el separador de símbolos (*tokenizer*) y el etiquetador gramatical para obtener una óptima etiquetación.

2.3.2. Lematización

El proceso de lematización es utilizado a menudo en el campo de la recuperación de información. En este caso, lo que se busca es saber cuántas veces se repite una misma raíz semántica en un mismo documento o contexto, es decir, saber cuántas veces aparece el término *battery* o *batteries* en el mismo contexto, para así obtener la frecuencia de aparición de cada término. Para ello, los trabajos mencionados en la sección de **Trabajo relacionado**,

[6] http://www.infogistics.com/textanalysis.html
[7] http://web.media.mit.edu/~hugo/montylingua
[8] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger
[9] http://nlp.stanford.edu/software/tagger.shtml
[10] crftagger.sourceforge.net
[11] incubator.apache.org/opennlp

tales como [2, 6, 8] en los que además de eliminar las palabras de parada (*stop words*) como son los signos de puntuación, etc., se aplicaba un lematizador. En [10, 11] también se utilizaba un lematizador, pero éste está embebido en el propio etiquetador gramatical. Dos de los lematizadores más utilizados en técnicas de recuperación de información son los algoritmos de Lovins [36] y Porter [37], basados en la eliminación de sufijos. Lovins trata de encontrar la terminación que mejor se adapta, mientras que Porter usa un algoritmo iterativo con un pequeño número de sufijos y unas reglas de recodificación [38]. Después de probar varias implementaciones en Java, tales como el Paice/Husk Stemmer[12], una implementación[13] del algoritmo de Lovins, y una implementación[14] del algoritmo de Porter, se han encontrado algunos defectos sutiles. Sin embargo, el mismo Porter ha creado lo que hoy en día es Snowball[15], un entorno para desarrollar lematizadores, que es también el nombre del lematizador resultante escrito en Java. Es éste el que ha sido elegido dada su cuasi-perfecta actuación sobre los nombres extraídos de los comentarios y especificaciones técnicas.

2.3.3. Construcción de la matriz de términos-contextos

El hecho de trabajar únicamente con términos contenidos en contextos sin tener en cuenta el orden se conoce como *bolsa de palabras* (explicada en el capítulo de **Conceptos teóricos**), dado que cada contexto es representado en una matriz de términos y contextos, donde cada celda representa la frecuencia de aparición de ese término (de ahí la necesidad del proceso de lematización) en ese contexto. Además, dado que se ha decidido analizar la actuación de los métodos mencionados en los distintos contextos descritos, documentos, secciones y párrafos, se obtienen tres escenarios para los que se han de preparar los comentarios. Es este el trabajo propio desarrollado complementado por los programas citados en este capítulo.

Como uno de los factores implícitos es la coocurrencia de los términos, también se ha llevado a cabo la construcción de tres *comentarios extra*, en los que se agrupan todas las especificaciones técnicas obtenidas, agrupadas por su característica principal, es decir, para *battery*, *organizer*, y *multimedia*. De esta forma, el objetivo del análisis se centra en observar el comportamiento de las diferentes técnicas de recuperación de información dentro de documentos, secciones y párrafos de la misma colección de comentarios, adjuntando además tres *comentarios extra* que contienen las especificaciones técnicas de cada una de las características principales a analizar.

El proceso llevado a cabo es el de almacenar en ficheros cada uno de los 542 comentarios y a través del etiquetador obtener los ficheros con las etiquetas asignadas a cada una de las palabras. Estos ficheros son procesados por un programa propio desarrollado en Java que realiza acciones como: identificar los sustantivos simples y compuestos; obtener su raíz con el lematizador; establecer la frecuencia por documento, sección y párrafo, cuyo valor sea mayor que 1 (umbral de frecuencia); y construir la matriz de términos-contextos para cada escenario posible, con y sin *comentarios extra*.

[12] www.comp.lancs.ac.uk/computing/research/stemming/paice/article.htm
[13] www.cs.waikato.ac.nz/~eibe/stemmers
[14] tartarus.org/~martin/PorterStemmer
[15] snowball.tartarus.org

3. Conceptos teóricos

En este capítulo se recogen los conceptos teóricos sobre los cuales se fundamenta la investigación realizada en este proyecto fin de carrera. La descripción de los métodos utilizados y los procesos seguidos, así como los fundamentos teóricos de las técnicas de recuperación de información utilizadas. Se detalla con mayor precisión en el Capítulo **Theoretical grounds** (ver **Anexo 1**).

3.1. Proceso manual

Se ha diseñado y llevado a cabo un proceso de búsqueda de términos semánticamente relacionados a partir de los términos extraídos de las especificaciones técnicas relativas a *battery*, *organizer*, y *multimedia* de manera manual, como lo haría un proceso automático, pero sin la ayuda de ningún método ni técnica que facilitara el desarrollo del mismo. El punto de partida es el mismo, las especificaciones y comentarios, pero no han sido analizados todos los 542 comentarios, sino que se ha seleccionado aleatoriamente una pequeña muestra del corpus de 16 comentarios sobre la cual se ha desarrollado el proceso. Considerando el mismo corpus sería tedioso y casi imposible de llevar a cabo ningún análisis, dado que los resultados sólo se quieren tener como referencia para demostrar la necesidad de métodos automáticos.

La base de este proceso consiste en calcular el porcentaje de veces que un término inicial coocurre en el mismo contexto con otro cualquiera en relación a las veces que aparece en toda la muestra analizada. Entonces, la fortaleza de la relación semántica entre ambos términos se mide mediante la superación de un porcentaje umbral, en ambos sentidos, desde el término inicial y desde el nuevo término potencialmente semánticamente relacionado. Finalmente, se pretende evaluar si la relación resultante tiene sentido o no, de manera que no se obtengan resultados erróneos. Estos nuevos términos incrementan el grupo inicial semánticamente relacionado dentro de cada grupo de términos de cada una de las características principales.

3.2. Proceso automático

El diseño del proceso automático completo se ha resumido en la figura **4-1** (ver **Anexo 1**), en la que se puede ver la división claramente identificada de los dos procesos principales: el primero consiste en la extracción de términos de las especificaciones técnicas y comentarios; y el segundo trata de medir la similitud semántica de las relaciones entre los términos

encontrados y los iniciales. Aunque el proceso ha sido ampliamente descrito con anterioridad, no se ha hecho hincapié en la aplicación de los métodos de recuperación de información, por eso se detalla en esta sección. Aunque existen muchas variaciones y alternativas disponibles, se han seleccionado aquellos métodos clásicos que han determinado de alguna manera la evolución de este tipo de técnicas en los últimos años, como se muestra en la figura **4-2** (ver **Anexo 1**), en la que se puede ver la evolución desde el uso de palabras clave (*keywords*), pasando por la frecuencia de términos junto con la frecuencia inversa de documentos (*tf-idf*) y la coocurrencia, hasta llegar al actual modelado de temas (*topic modeling*).

Los métodos seleccionados son *Latent Semantic Indexing* (LSI) [3], *Probabilistic Latent Semantic Indexing* (PLSI) [4] y *Latent Dirichlet Allocation* (LDA) [5]. Todos emplean el modelo de *bolsa de palabras* (*bag of words*) o unigrama, que es el caso particular del modelo n-grama de tamaño 1. Este tipo de modelos se emplean para el procesamiento estadístico del lenguaje natural, como el diseño de algoritmos automáticos que extraen datos de cadenas de texto. En el caso de bigramas y trigramas, el tamaño se incrementa hasta 2 y 3 términos respectivamente, pero en nuestro caso no se considera ningún orden, así que el modelo de unigrama es perfecto. Además, lo que se busca es que el proceso se lleve a cabo sin conocimiento extra, o lo que es lo mismo, sin supervisión, y los métodos escogidos no la tienen, ya que no emplean ningún recurso externo del tipo WordNet [33], ConceptNet [34] o Wikipedia[16]. En general, las técnicas que emplean algún tipo de ayuda externa o supervisión suelen ser más efectivas, pero éstas dependen de un mantenimiento externo que no es controlable. De ahí que el objetivo sea el de conseguir los mejores resultados con las técnicas que no dependen de conocimiento extra. Como conclusión, en la **Tabla 3-1** se muestra un ejemplo del análisis de LSI para *battery*. Estos pasos son seguidos en cada uno de los análisis simplemente cambiando los parámetros LSI, *battery* y documentos.

1. Lanzar LSI en el escenario de documentos con todas las posibilidades de sus parámetros, obteniendo como resultado las matrices, con las puntuaciones de los términos organizadas en dimensiones (vectores de términos).
2. Elegir cuál o cuáles dimensiones tienen los términos iniciales relativos a *battery* mejor puntuados dentro de la dimensión ordenada en orden descendente, considerando especialmente los primero 30 términos.
3. Aplicar el método de descarte sobre esos potenciales nuevos términos, que elimina todos los términos que aparecen en más de 5 dimensiones distintas de la que se está analizando con puntuaciones mayores que la actual.
4. Juzgar como persona los potenciales nuevos términos clasificándolos en *positivo verdadero (tp),* resultado correcto*, falso positivo (fp),* resultado inesperado*, negativo verdadero (tn),* una correcta eliminación de resultado*,* y *falso negativo (fn)*, una eliminación errónea, omitiendo los términos iniciales, los cuales siempre son resultados correctos.
5. Calcular y mostrar gráficamente la precisión, exactitud y recall[17] de estos potenciales nuevos términos dentro de las dimensiones seleccionadas.
6. Extraer los nuevos términos relativos a la característica *battery* que mantengan como mínimo al 80% de precisión.

Tabla 3-1. Algoritmo de extracción de términos semánticamente relacionados.

[16] http://es.wikipedia.org

[17] Porcentaje de términos bien devueltos y no descartados, del total de términos bien devueltos.

3.3. *Latent Semantic Indexing* (LSI)

En esta sección se presenta una breve descripción [39] de los conceptos matemáticos de LSI, que se detallan en mayor grado en [3]. La clave de la innovación de LSI fue el uso de la Descomposición en Valores Simples (SVD) para descomponer la matriz original de términos-contextos. Si se tiene que A_0 tiene t filas, una para cada término, y d columnas, una para cada contexto, la aplicación de la SVD nos da como resultado $A_0 = U_0 S_0 V_0^T$, cuyas matrices resultantes son: una matriz $t\ x\ m$, U_0, de columnas ortonormales (dimensiones de términos) también llamadas vectores singulares izquierdos; una matriz $m\ x\ m$ diagonal, S_0, con los valores singulares ordenados decrecientemente; y una matriz m $x\ d$, V_0^T, de columnas ortonormales (dimensiones de documentos) también llamadas vectores singulares derechos. El valor de m es el rango de la matriz A_0, como se puede ver en la figura **4-3** (ver **Anexo 1**). Sin embargo, no sólo el uso de la SVD es lo que distingue a LSI, si no que éste se queda con los k valores singulares más altos de la matriz S_0 y pone a cero el resto, reduciendo todas las demás matrices a k como se ve en la **Figura 3-1**.

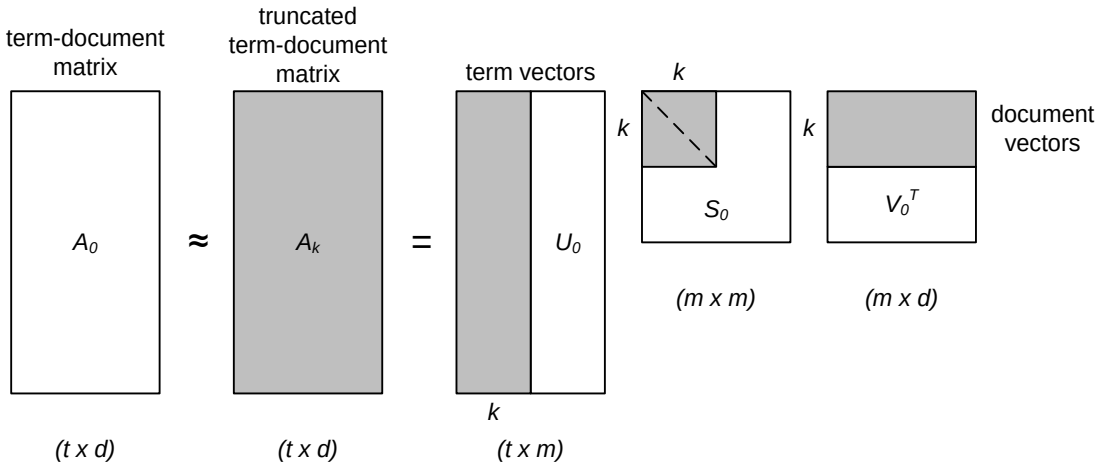

Figura 3-1. SVD de la matriz de términos-contextos A_0 truncada a k valores singulares.

El rendimiento de LSI puede verse incrementado considerablemente a partir de la selección del parámetro k de 10 ó 20, consiguiendo picos cuando éste vale entre 70 y 100, y decreciendo a partir de ese valor. Sin embargo, cuando se trabaja con grandes valores de D, todavía está abierto el debate, pero lo experimentos que se han realizado ajenos a esta investigación indican que con valores de k entre 100 y 300 se consiguen los mejores resultados [40]. Debido a ello, los valores a analizar en esta investigación han sido 50, 100, 150, y 200.

La implementación utilizada ha sido proporcionada por el Departamento de Sistemas de Información (IfIS[18]), en la cual además se utiliza una función de pesado, donde los elementos a_{ij} son calculados por el producto de la función de pesado de entropía global y local por una función logarítmica como se puede ver en la tabla **4-2** (ver **Anexo 1**). Se ha modificado casi por completo para poderla ajustar a los datos utilizados aquí y mostrar los resultados de manera clara y precisa.

[18] www.ifis.cs.tu-bs.de (Institut für Informationssysteme, TU Braunschweig)

3.4. *Probabilistic Latent Semantic Indexing* (PLSI)

En esta sección se presenta una breve descripción de PLSI [4]. En comparación con LSI, PLSI se basa en una mezcla de descomposición, derivada de un modelo de clases latentes, pero con unos fundamentos estadísticos sólidos [41]. Principalmente está basado en un modelo estadístico llamado *modelado de aspecto* (*aspect model*), que es un modelo de latencia variable para datos concurrentes, que asocia variables de clases latentes z_k, $k \in \{1, 2, \dots, K\}$ con cada *observación*, donde K es el número de clases latentes. El número de clases latentes, es similar al número de dimensiones para LSI, un parámetro que tiene que ser seleccionado previamente. La *observación* consiste en la aparición de una palabra w_j, $j \in \{1, 2, \dots, N\}$en un particular documento d_i, $i \in \{1, 2, \dots, M\}$ donde N es el número de palabras y M el número de documentos. Estas clases latentes pueden ser entendidas como los temas o categorías (*topics*) que componen el texto. Las distribuciones de probabilidad que asocian las variables latentes con las palabras y documentos describen cuánto están relacionadas estas con los temas. El modelo generativo para la *observación* se muestra gráficamente en **Figura 3-2** y se define a continuación [42]: 1) Obtener un documento d_i en el que la aparición de una palabra será observada con probabilidad $P(d_i)$; 2) cuando el documento d_i es conocido, seleccionar el tema z_k de la palabra con probabilidad $P(z_k|d_i)$. Esta distribución de probabilidad es también una medida del grado en que el documento es relevante para cada tema; y 3) cuando el tema es conocido, seleccionar la palabra w_j cuya aparición es observada con probabilidad $P(w_j|z_k)$.

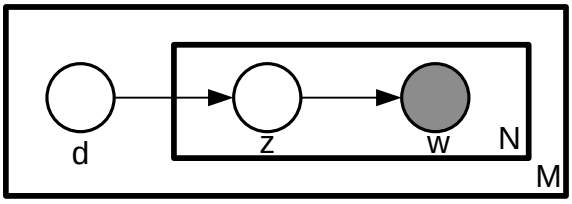

Figura 3-2. Modelo de PLSI.

En vez de utilizar el modelo convencional del algoritmo de Maximización de la Expectativa (*Expectation Maximization*), Hoffman [4] propuso un algoritmo de EM Templado (*Tempered EM*) para calcular las distribuciones de probabilidad a partir del los datos de entrenamiento (en este caso los comentarios). El algoritmo alterna entre estos dos pasos: un paso de *expectativa* (*E*) donde las probabilidades posteriores son calculadas en base a las actuales estimaciones de los parámetros; y un paso de *maximización* (*M*) en el que los parámetros son actualizados en función de la minimización de los criterios y la dependencia de las probabilidades posteriores calculadas en el paso *E* (ver **Ecuación 3-1** y **Ecuación 3-2**). En las ecuaciones, el término $n(d_i|w_j)$ muestra la frecuencia de la palabra w_j en el documento d_i.

$$\boldsymbol{P(w_j|z_k) = \frac{\sum_{i=1}^{N} n(d_i,w_j) P(z_k|d_i,w_j)}{\sum_{m=1}^{M}\sum_{i=1}^{N} n(d_i,w_j) P(z_k|d_i,w_m)}} \textbf{(5.2)} \quad \boldsymbol{P(z_k|d_i) = \frac{\sum_{j=1}^{N} n(d_i,w_j) P(z_k|d_i,w_j)}{\sum_{m=1}^{M} n(d_i,w_m)}} \textbf{(5.3)}$$

Los mismos valores de k utilizados para LSI, 50, 100, 150 y 200, han sido tomados para PLSI porque en [43] se obtuvieron los mejores resultados para los valores de 48 y 128. Después de probar con multitud de implementaciones en Matlab y C, se ha escogido una versión en C++

de Search Art[19] que funciona correctamente con los datos utilizados en esta investigación. La única modificación reseñable ha sido el cambio en el tipo de dato dado que el uso de memoria se disparaba con la división en secciones y párrafos.

El resultado de la aplicación de PLSI se puede traducir en los mismos términos de LSI, entendiendo las matrices resultantes como $\hat{U}_k = \left(P(w_j|z_k)\right)_{i,k}$, $\hat{V}_k = \left(P(d_i|z_k)\right)_{j,k}$ y $\hat{S}_k = diag\left(P(z_k)\right)_k$, se puede escribir el modelo de probabilidad P como un producto de matrices $P = \hat{U}\hat{S}\hat{V}^T$, donde la matriz generada por $P(w_j|z_k)$ es la que interesa analizar.

3.5. *Latent Dirichlet Allocation* (LDA)

De la misma forma que se explica PLSI en el apartado anterior, LDA [5] es un modelo generativo de probabilidades para colecciones de textos. Utiliza un modelo Bayesiano jerárquico de tres niveles, en los que cada elemento de la colección está modelado como una mezcla finita sobre el grupo de temas o categorías (*topics*) latentes. Blei et al. [5] consideran que el trabajo de Hoffman [4] es incompleto porque no da un modelo probabilístico a nivel de documento y esto genera una serie de problemas como la falta de claridad a la hora de asignar una probabilidad a los documentos. La diferencia más sustancial consiste en que no hace falta estimar la probabilidad de obtener un documento. En su lugar, esto se consigue cambiando el modelo generativo separando el proceso para cada documento, y utilizando la distribución de clases latentes de palabras para determinar la distribución de clases latentes de documentos.

Para comprender mejor la diferencia, en la **Figura 3-3** se muestra cómo ambos elementos conocidos, documentos y palabras, están generados por una distribución de Dirichlet, cuyos parámetros, α y β respectivamente, son los vectores de parámetros seleccionados para generar el modelo. Además, Blei et al. utilizan un algoritmo basado en el clásico algoritmo de Maximización de la Expectativa (*EM*), para calcular los distintos valores en las sucesivas iteraciones hasta lograr que converja.

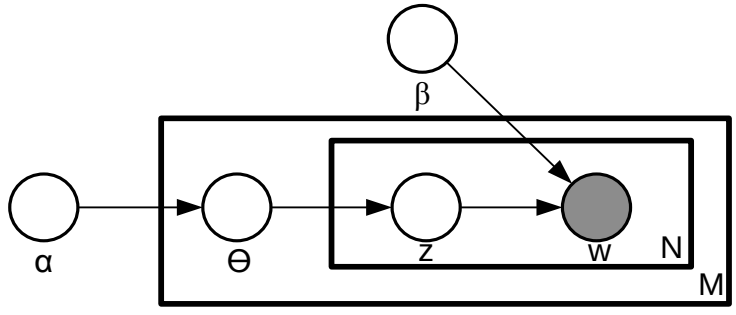

Figura 3-3. Modelo de LDA.

La implementación utilizada corresponde al paquete[20] LDA desarrollado por Daichi Mochihashi, que incluye versiones en C++ y MatLab, de las cuales, aunque es más cómodo

[19] http://www.semanticsearchart.com/researchpLSA.html
[20] http://chasen.org/~daiti-m/dist/lda

trabajar con grandes matrices en Matlab, se ha utilizado la versión en C++ dado que es 8 veces más rápida que la otra, y a pesar de que el análisis posterior se ha tratado en MatLab como se hiciera para LSI y PLSI.

3.6. Propuestas de métodos automáticos

Dado que en la propuesta del presente proyecto se mencionó la posibilidad de desarrollar un método propio en función de la bondad extraída de los tres métodos automáticos analizados, esta sección resume las propuestas de métodos automáticos escritos en MatLab, que no han sido analizados debido a que los resultados no son los esperados.

3.6.1. Coocurrencia y similitud de contextos

Basado en las técnicas analizadas en esta investigación se ha intentado desarrollar una herramienta que permitiera establecer unas determinadas puntuaciones a los distintos términos extraídos de los comentarios. Se fundamenta en las mismas matrices de términos-contextos construidas anteriormente para poder ser aplicado, pero difiere en los resultados obtenidos, ya que el objetivo es que devuelva una matriz simétrica de términos-términos donde en cada fila o columna se tuviera la categoría o tema (*topic*) de cada término, siendo este definido por las relaciones con los términos mejor puntuados dentro de su fila o columna como en [44], que se ha usado y analizado la matriz de términos-términos obtenida del resultado de LSI.

El método propio tiene como fundamento dos situaciones: que cada término se encuentra en uno o diferentes contextos y éstos pueden ser comparados a través de medidas como la distancia del coseno o de Euclides, suponiendo que aquellos términos que aparecen en contextos similares están semánticamente relacionados; y que, de la misma forma, aquellos términos que aparezcan en el mismo contexto, es decir, que concurran, también se puede calcular su porcentaje de coocurrencia (como se hace para el análisis manual), y calcular su grado de relación con los otros términos. En un entorno mínimo, como el de [44], se obtuvieron resultados favorables, pero cuando se probó con los comentarios reales, el tiempo de cálculo tendía a infinito, dado que existen millones de comparaciones de grandes vectores.

3.6.2. *Cascade Latent Dirichlet Allocation* (CLDA)

O mejor dicho LDA en Cascada, porque se ha supuesto que, dado que LDA devuelve vectores donde aparecen los términos agrupados por categorías, éstas categorías pueden ser tratadas como contextos igualmente, y aplicar una segunda vez LDA sobre estos primeros resultados. El problema reside en la naturaleza de los datos, ya que las matrices de términos-contextos expresan la frecuencia en números enteros, mientras que el resultado es una probabilidad expresada en números reales. Esto causa problemas en la segunda ejecución de LDA y nunca se obtienen resultados contrastables.

4. Análisis

En este capítulo se describe brevemente el análisis llevado desarrollado en detalle en el Capítulo **Analysis** (ver **Anexo 1**), sobre los distintos métodos y técnicas anteriormente expuestos ejecutados en los escenarios construidos para tratar de demostrar el objetivo planteado en este proyecto fin de carrera.

4.1. Análisis manual

Se ha considerado como premisa inicial lo que un ser humano entiende por contexto, ni muy amplio como un documento entero ni muy reducido como una frase, sino que la entidad más común es el párrafo, dado que suele venir recogida toda aquella información relativa al tema que se trata en ese momento. No se han tenido en cuenta las diferentes situaciones de análisis con y sin los 3 *comentarios extra* explicados en el Capítulo **3**, dado que su adhesión a los 16 comentarios aleatorios seleccionados supondría más del 15% de la muestra representativa, por lo tanto se procede a analizar esta muestra sin alteraciones.

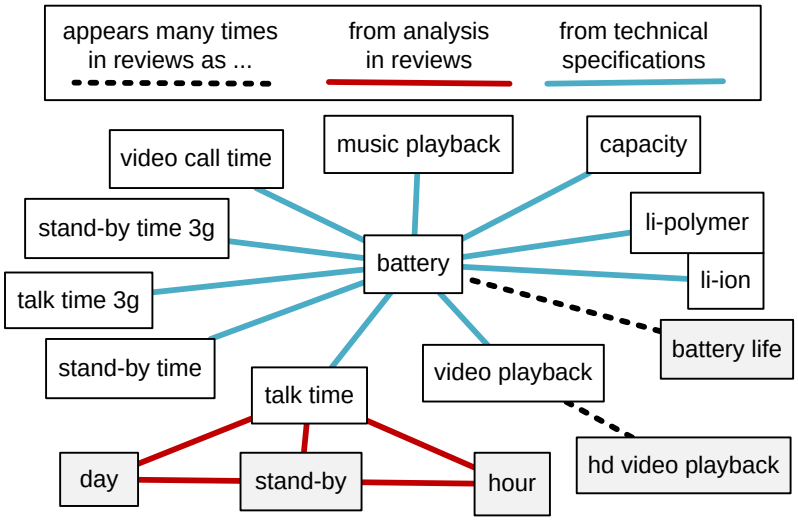

Figura 4-1. Red semántica resultante de la extracción manual.

Para comenzar con el análisis, se han extraído los términos encontrados en las especificaciones técnicas, relativos a la característica principal de *battery*, obteniendo así el primer grupo semántico en torno a este término. A partir de estos términos, se ha procedido a su búsqueda en la pequeña colección de comentarios, en la que se han localizado los párrafos donde aparecen y se ha procedido al conteo de las veces que cada uno de ellos coocurre con otro término o entre ellos. Una vez realizado el conteo, se ha calculado el porcentaje de coocurrencia desde el punto de vista de los términos iniciales, y si éste alcanza el 50%,

entonces, esa relación en ese sentido se supone que es fuerte, pero tiene que serlo también en sentido contrario. Una vez se tiene que ambos términos coocurren lo suficiente como para considerar esa relación semántica fuerte, se deduce que están semánticamente relacionados y el nuevo término se extrae y se añade al grupo semántico. Además, se han tenido en cuenta aquellos términos compuestos que poseen parte de algún término inicial, pero no han sido contabilizados para ningún resultado aunque se exponen en la misma red semántica. Las operaciones de conteo y cálculo de porcentajes se recogen en la tabla **5-1** (ver **Anexo 1**) y se puede observar el grupo semántico resultante en la **Figura 4-1**. La misma operación se realiza con todos los términos de *battery*, *organizer*, y *multimedia*.

4.2. Análisis de LSI

Como se explica anteriormente, LSI en esencia se basa en la Descomposición en Valores Singulares (SVD), la cual es truncada a los k valores singulares mayores. De la SVD truncada sólo se considera la matriz U_k, ya que en ésta se almacenan los términos distribuidos en los k vectores, llamados dimensiones, como se observa en la figura **5-4** (ver **Anexo 1**). Estas dimensiones almacenan la puntuación otorgada por LSI a cada término en cada dimensión, cuya ordenación en orden descendiente muestra al principio los términos más representativos, entendiendo que son estos términos los que definen el significado de la dimensión. Es por esto que siguiendo los pasos enunciados en la **Tabla 3-1**, se pretende encontrar aquella dimensión que mejor defina el significado de *battery*, *organizer*, y multimedia, suponiendo que aquella o aquellas dimensiones que contengan más términos iniciales o estos estén mejor valorados, serán más susceptibles de definir su propio grupo semántico. Además, se ha realizado ésta búsqueda en los distintos escenarios y con los diferentes valores del parámetro k.

Dada la imposibilidad de mostrar todas las dimensiones, éstas han sido almacenadas en el DVD que se adjunta con este documento, pero se incluyen las más relevantes en el **Annex A: LSI's running tables** (ver **Anexo 1**). En primer lugar se ha buscado aquellas dimensiones relativas a *battery* en los distintos escenarios. Con la aplicación de LSI con los distintos valores del parámetro k, 50, 100, 150, y 200, sucede que si observamos como en el escenario de los documentos aparece la *dimensión 1*, como una de las dimensiones destacadas con el valor de del parámetro k igual a 50 sin *comentarios extra*, ésta misma dimensión repite las mismas puntuaciones para los mismos términos en los valores de 100, 150, y 200. Esta peculiaridad de LSI lleva a pensar que el incremento del parámetro k no mejora las dimensiones ya existentes, sino que aporta más posibles dimensiones susceptibles de definir la característica de *battery*. El problema, si se le puede llamar así, es que no siempre sucede esto, sino que en ciertos casos, como es el de la *dimensión 5* también sin *comentarios extra*, se puede ver como se detalla que para los valores de 50 y 150 ésta mantiene las puntuaciones de sus términos relevantes, mientras que para el resto, 100 y 200, sus términos iniciales están puntuados con el mismo valor, pero con signo negativo. Se puede entender que LSI puntúa alto (positivo) o bajo (negativo) en función de si la concurrencia de ese término implica una relación de sinónimos o de antónimos.

A pesar de todo, LSI no funciona tan bien como se pretendía para esta primera característica, *battery*, dado que no hay dimensiones que contengan algún término inicial de ésta entre sus 30 primeros términos cuando la dimensión está ordenada, ni en el escenario de documentos. Sin embargo, para los escenarios de secciones y párrafos LSI puntúa términos como *battery* y *talk time* dentro de los mencionados top 30 términos de alguna dimensión. Éste es caso de la *dimensión 56* en el escenario de secciones, que aparece casi igual puntuada con y sin *comentarios extra*, y sobre la que se puede: aplicar el método de descarte sobre los 30 primeros términos, cuyos resultados pueden verse en el **Annex D: Discarding method applied to LSI dimensions** (ver **Anexo 1**); y aplicar el juicio personal sobre los términos bien o mal devueltos y bien o mal descartados, cuyos resultados se localizan en las **5-5** y **5-6** (ver **Anexo 1**), relativas a secciones y párrafos. Sin embargo, sólo de la ejecución de LSI en sobre párrafos se pueden extraer conclusiones relevantes, porque si se atiende a las gráficas **5-3** y **5-4** (ver **Anexo 1**), que cruzan los valores de precisión y recall, se puede ver que las dimensiones seleccionadas en secciones, empiezan en el 0% de precisión y aunque luego remontan, ni siquiera alcanzan el 80%, pero de todas formas no mantienen más del 80% de precisión en ningún momento, a diferencia de la *dimensión 17*, que en la ejecución de LSI en párrafos sin *comentarios extra*, consigue extraer los términos tales como *hour*, *talk*, *gsm*, *network* y *day*, semánticamente relacionados con al menos alguno de los términos iniciales de *battery*, manteniendo una precisión y exactitud del 100%, pero tan sólo un recall del 41.18%. Las gráficas relativas a éstas medidas pueden verse en **Annex G: Precision, recall and accuracy graphs of LSI** (ver **Anexo 1**). Cabe destacara que aunque no se alcance el recall completo en ninguna de las dimensiones analizadas, esto significa que de todos los términos correctamente devueltos, algunos de ellos han sido descartados por el método de descarte.

Éste es un pequeño ejemplo de lo que más adelante se ha ejecutado sobre las características de *organizer* y *multimedia*, en las cuales se han obtenido mucho mejores resultados, cuyos procesos se detallan ampliamente en la Sección **LSI analysis** del Capítulo **Analysis** (ver **Anexo 1**). En los tres anexos citados antes se recogen: las tablas de las dimensiones más representativas de cada característica y la selección hecha dentro de éstas; las gráficas relativas a la aplicación del método de descarte; y las gráficas de las medidas calculadas a partir del juicio personal sobre los 30 primeros términos de cada dimensión seleccionada.

La obtención de las dimensiones donde mejor puntuados y mejor posicionados están los términos iniciales se ha desarrollado en MatLab, con el cual se han obtenido las distintas gráficas y comparaciones. La ejecución de cada paso está concretamente detallada en cada carpeta de cada método, siendo reunidos todos los resultados en la misma carpeta de resultados. Todo este material se adjunta en el DVD junto con las explicaciones oportunas para su prueba y utilización.

4.3. Análisis de PLSI

La diferencia con respecto a LSI es la naturaleza de los datos, ya que cada implementación de cada método devuelve los datos de una forma, pero éstos siempre son

tratados finalmente por los mimos procesos de análisis desarrollados en MatLab, para unificar y contrastar todas las medidas obtenidas. A diferencia de LSI, donde los valores de las puntuaciones iban desde valores reales negativos hasta positivos sin ningún tipo de límite ni superior ni inferior, en PLSI se determinan distribuciones de probabilidad, lo que hace que se reparta el valor 1 en todos los términos que tengan algo de relación con el tema o categoría (llamada dimensión en LSI), siendo lo valores otorgados a los términos entre 0.0 y 1.0. De la misma forma que en LSI se analiza la matriz U_k, aquí se analiza la matriz $\widehat{U}$, de la cual se decía que era similar a la de LSI, pero que contenía los valores relativos a $P(w_j|z_k)$, que indican la probabilidad de aparecer una palabra w_j en un tema z_k.

El análisis se ejecuta de la misma forma que con LSI, donde para comprobar el funcionamiento y el rendimiento de PLSI se siguen las mismas directrices enunciadas en la **Tabla 3-1**. Se sigue el mismo orden, empezando por la característica principal *battery*, analizando la aparición de sus términos iniciales en los escenarios de documentos, secciones y párrafos, en los que, a diferencia de LSI, PLSI solo actúa bien en el primero, es decir, que mientras LSI no obtenía ningún resultado en el escenario de documentos, PLSI sólo encuentra potenciales temas que definan el grupo semántico de *battery* ahí, y no haya nada en las secciones y en los párrafos. Éste hecho puede darse debido a que la implementación utilizada, o en general el algoritmo de PLSI, consigue mejores resultados en contextos más amplios, ya que se repite para las demás características, *organizer* y *multimedia*.

En general, PLSI obtiene unos resultados más pobres que LSI, a diferencia de lo que se suponía, dado que debería haber una evolución reflejada en la consecución de los objetivos expuestos con anterioridad, al igual que sucede con la aplicación de las técnicas de recuperación de información. En los anexos **Annex B: PLSI's running tables**, **Annex E: Discarding method applied to LSI topics** y **Annex H: Precision, recall and accuracy graphs of PLSI** (ver **Anexo 1**) pueden verse respectivamente las tablas de los temas en las distintas ejecuciones de PLSI, los distintos resultados de la aplicación del método de descarte, y las gráficas resultantes de la aplicación del juicio personal sobre lo bien o mal devueltos y lo bien o mal descartados que son los 30 primeros términos de cada tema analizado.

4.4. Análisis de LDA

Para terminar con el análisis, LDA también es analizado en el mismo marco que los dos anteriores, pero éste, a diferencia de PLSI, sí que cumple las expectativas evolutivas planteadas como base de esta investigación, obteniendo los mejores resultados en casi todos los pares analizados de escenario-característica principal. De igual manera, aunque su ejecución y obtención de resultados se realiza con la versión el C++, el análisis se realiza en el marco de MatLab, desde el cual se miden las mismas magnitudes y se trata a los datos de la misma manera para poderlos contrastar con certeza. Siguiendo las directrices probabilísticas del método, éste puntúa los términos dentro de cada categoría o tema con valores entre 0.0 y 1.0, modelando una distribución de probabilidad por cada tema obtenido. Se analiza aquella matriz U_0, similar a la matriz devuelta por LSI U_k, pero que contiene los valores relativos a las

distribuciones de probabilidades similares a las de PSLI en $P(w_j|z_k)$, que se muestran en la **Figura 4-2**.

A nivel de cada característica analizada en los distintos escenarios, LDA consigue algún resultado relevante (algún tema del cuál obtener algún término) para todas las posibles combinaciones de escenario y característica principal, teniendo su actuación más débil con la característica *battery* en el escenario de documentos, donde no supera la actuación de PLSI, pero sí la iguala. En cuanto a las demás situaciones, mantiene una media extracción de términos bien semánticamente relacionados muy alta en comparación con los otros métodos. En cuanto a los términos iniciales encontrados en los temas que potencialmente son más susceptibles de definir a las características principales, se han encontrado casos en los que hasta 11 términos de los 30 analizados pertenecen al grupo inicial semánticamente relacionado extraído de las especificaciones técnicas, lo que indica que el método actúa de una manera muy eficiente con los distintos contextos planteados.

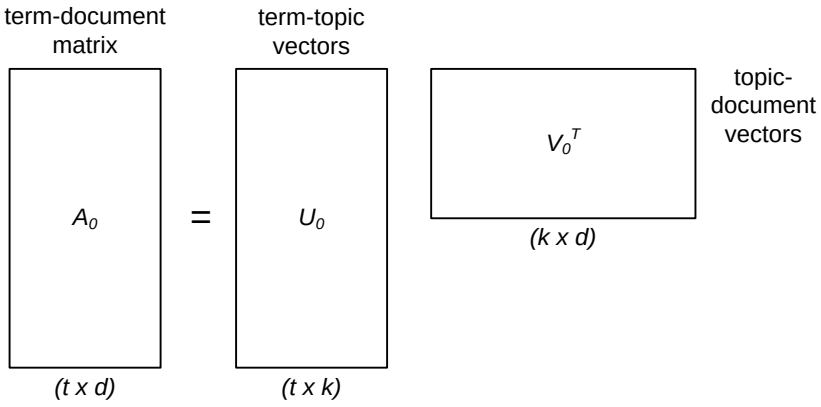

Figura 4-2. Representación matricial de las probabilidades de términos-temas y temas-documentos de LDA.

En general, LDA obtiene los más regulares y mejores resultados, de acuerdo a lo que se suponía, demostrando la evolución descrita en cuanto a las técnicas de recuperación de información utilizadas en esta investigación. En los anexos **Annex C: LDA's running tables**, **Annex F: Discarding method applied to LSI topics** y **Annex I: Precision, recall and accuracy graphs of LDA** (ver **Anexo 1**) pueden verse respectivamente las tablas de los temas en las distintas ejecuciones de LDA, los distintos resultados de la aplicación del método de descarte, y las gráficas resultantes de la aplicación del juicio personal sobre lo bien o mal devueltos y lo bien o mal descartados que son los 30 primeros términos de cada tema analizado.

5. Resultados

En este capítulo se resumen los resultados obtenidos en el análisis de los métodos manuales y automáticos, presentados anteriormente, y se comparan unos con otros. El mismo resumen con mayor grado de detalle puede verse en el Capítulo **Results** (ver **Anexo 1**).

El análisis se ha desarrollado en un ordenador portátil Packard Bell, modelo *Easynote TJ76*. Este ordenador posee un procesador de Intel Core i5 con dos núcleos M430 de 2.27GHz, complementado con de 4GB de memoria RAM. Dentro de este ordenador, los métodos automáticos han sido ejecutados alcanzando unos tiempos de ejecución del orden de segundos (LSI), hasta varias horas (PLSI y LDA) en función de la convergencia. Sin embargo, estos datos no son considerados como relevantes porque las técnicas aquí utilizadas se suelen ejecutar en servidores dedicados y no en ordenadores personales.

5.1. Battery

En especial, para la característica principal *battery*, se puede ver en la tabla **6-1** (ver **Anexo 1**) cómo la influencia de ser el grupo semántico con menor número de términos iniciales, y quizá menos populares (que aparecen menos en los comentarios) salvo alguna excepción, este hecho ha llevado a que los métodos de recuperación de información analizados obtengan unos resultados mediocres en comparación con las otras dos características principales. No obstante, con LDA se han conseguido unos resultados considerablemente alentadores, ya que en los escenarios de secciones y párrafos, en este último sobre todo, ha conseguido constatar la evolución supuesta de los métodos utilizados, obteniendo a partir de 3 y 2 términos iniciales en los *temas 48* y *35*, hasta 16 y 15 nuevos términos semánticamente relacionados con los iniciales, lo que suponen 19 y 17 términos de los 30 términos por tema analizados.

En relación al método manual, aunque puede entenderse que no es comparable con ningún escenario dado que se ha utilizado una colección de comentarios reducida, ha sido comparado con todos para demostrar la mejora al aplicar técnicas automáticas. Además, se entiende que no podría ser escalable, es decir, que se obtuvieran mejores resultados si se incrementara el número de comentarios analizados, porque esto haría descender también el porcentaje obtenido en el análisis descrito en el Capítulo 4. Sólo en el escenario de documentos son mejores los resultados del método manual, pero se reitera que la base de la comparación es meramente orientativa. En resumen, LDA es el método que mejor funciona para la búsqueda de una categoría que defina mejor al grupo semántico relativo a *battery*.

5.2. Organizer

A diferencia de lo que ocurre con *battery*, *organizer* posee un grupo de términos semánticamente relacionados mayor y además estos términos tiene significados más amplios y son más populares dentro de la colección de documentos. En la tabla **6-2** (ver **Anexo 1**), se puede observar cómo cualquier método analizado suele obtener buenos resultados en algún escenario con el fin de encontrar una dimensión o tema que defina esta característica principal a partir de los términos iniciales. La única situación donde no se ha extraído resultado alguno es el caso en el que LSI se ejecuta sobre el escenario de documentos, pero en el resto hay al menos una dimensión o tema que se acerca en mayor o menor medida al significado de *organizer*.

Como se menciona anteriormente, es en los documentos donde, a pesar de que LSI no obtenga resultados, PLSI y LDA obtienen unos resultados muy ajustados, como se puede ver en el **Gráfico 5-1**.

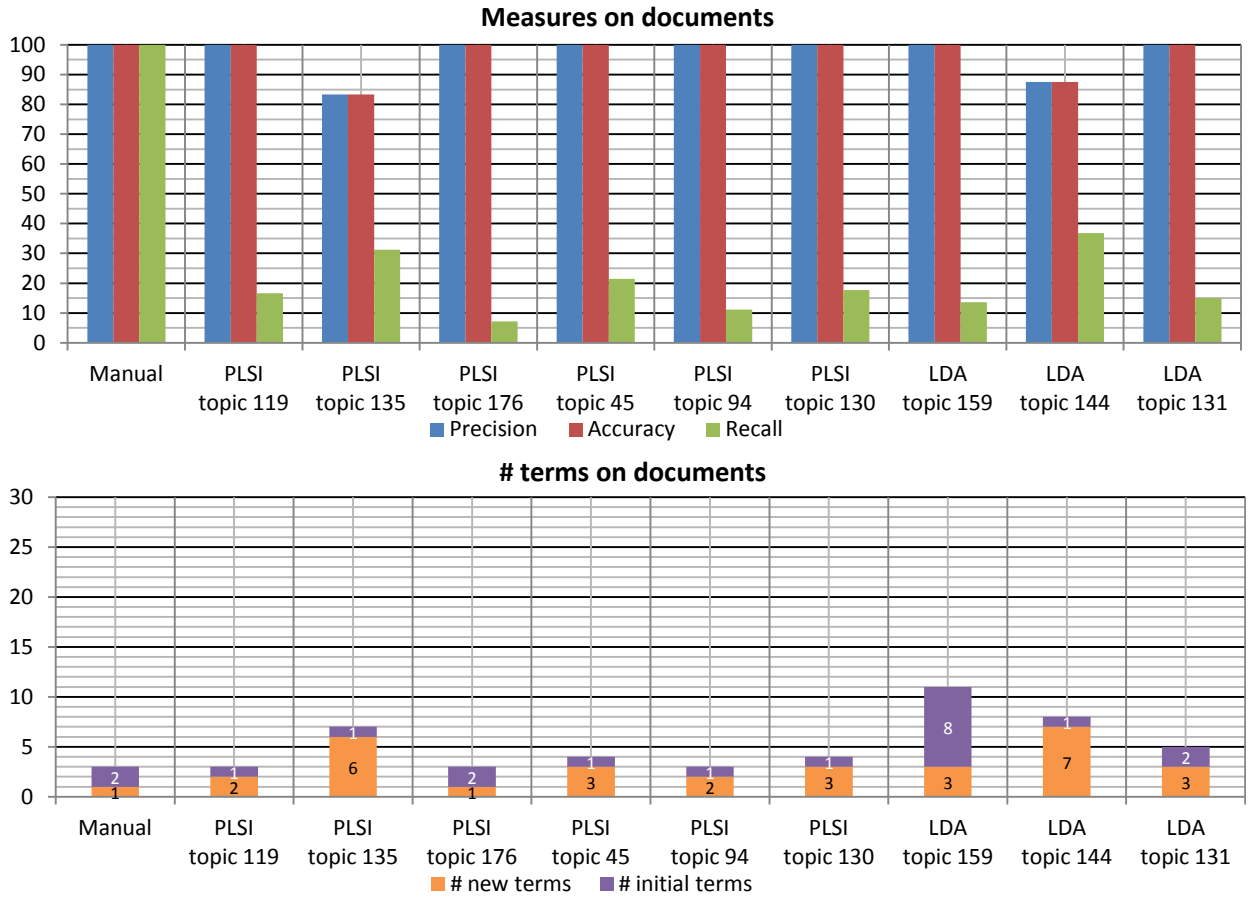

Gráfico 5-1. Resumen de las medidas y términos extraídos para *organizer* en documentos.

Si se consideran los *temas 135* de PLSI y *159* y *144* de LDA, se puede observar como el primero no tiene una precisión y una exactitud muy elevadas, pero por lo menos la precisión es mayor o igual que el 80%, mientras que el recall no supera el 35%, y se obtienen 6 nuevos términos a partir de 1 que aparece en los 30 primeros términos; el segundo es un caso particular, ya que aglutina hasta 8 términos iniciales, pero sólo extrae 3 nuevos términos; y por

último, el tercero es el mejor, dado que a partir de 1 término inicial extrae 7 nuevos, y obtiene un recall mayor del 35%, y una precisión y exactitud por encima del 85%. Otra peculiaridad sucede en el escenario de secciones, donde contra todo pronóstico LSI es el que obtiene mejores resultados hasta en dos ocasiones por encima de LDA, en la *dimensión 4*, con y sin *comentarios extra*, ambos resultados alcanzan casi la situación ideal en la que todos los términos devueltos son correctos y no son descartados, pero en la situación sin *comentarios extra* consigue 27, mientras que con ellos llega hasta 29 en la suma de los iniciales más los nuevos. LDA obtiene muy buenos resultados también, pero no llegan a ser tan altos como los de LSI. Sin embargo, en el escenario de párrafos, es LDA el método que mejor se adapta al contexto y a los términos iniciales.

5.3. *Multimedia*

Por último, la característica principal *multimedia* aglutina la mayor cantidad de términos iniciales, pero no por ello obtiene mejores resultados que *organizer*, porque la mayoría de sus términos no son tan comunes como para la anterior. En la tabla **6-3** (ver **Anexo 1**), se puede observar de manera global cómo la única situación en la que no se obtiene nada relevante es la misma que para *organizer*, cuando LSI es ejecutado sobre los documentos. Pero es en éstos donde LDA consigue el mayor número de nuevos términos extraídos, 8, junto con 2 términos iniciales suponen un 33.33% de los términos devueltos en los 30 primeros términos.

Para las secciones, aunque LSI obtiene muy buenos resultados con hasta 19 términos extraídos por la *dimensión 11* sin *comentarios extra*, pero es LDA nuevamente el que con 20 nuevos términos y 6 iniciales dentro de los 30 primeros, se entiende como mejor resultado dentro de los datos recogidos. Por último, LDA también supera las extracciones de LSI, a pesar de la supuesta evolución de LSI a PLSI y de PLSI a LDA. En general, PLSI obtiene unos resultados pobres en relación a la supuesta mejora que debía implicar su uso de modelos probabilísticos que se supone que modelan los temas de una forma más eficiente que LSI. En cuanto a los resultados del análisis manual, a pesar de haber encontrado más términos iniciales, tampoco se mejoran los resultados de LDA y LSI, demostrando otra vez la necesidad de éstos.

5.4. El parámetro k y la inclusión de *comentarios extra*

El análisis se ha visto condicionado a la utilización del parámetro k, de tal forma que se pueden comparar los resultados en función de qué valor se le ha dado. Si se tienen en cuenta la tabla **6-4** y la gráfica **6-10** (ver **Anexo 1**), se puede observar cómo, en general, conforme se incrementa el valor del parámetro k, se incrementa la precisión de los métodos en el escenario de documentos, es decir, que hay más dimensiones y temas seleccionados. No sucede lo mismo para los párrafos, cuyo número de dimensiones y temas seleccionados alcanza su máximo en parámetro $k = 100$, y luego decrece. En cambio, para las secciones, el número de

dimensiones y temas seleccionados se mantiene más regular, alcanzando pequeños picos en valores del parámetro $k = 100, 200$.

Finalmente, la inclusión o no de los *comentarios extra* no ha supuesto un cambio tan drástico como se suponía, dado que en algunos casos mejora muy poco, en otros mantiene el mismo número de dimensiones y temas seleccionados, y en algún caso (PLSI principalmente y total) es peor que sin ellos. Por lo tanto, se puede deducir que, salvo en casos puntuales, la inclusión de los *comentarios extra*, no ha supuesto la mejora esperada.

6. Conclusiones y futuro trabajo

En este capítulo se detallan las conclusiones obtenidas en este proyecto fin de carrera, tanto a nivel analítico como personal, en relación a la consecución de objetivos planteados, el los problemas encontrados en el desarrollo de la misma, el desarrollo temporal del proyecto y trabajo futuro a partir esta investigación. Además, se incluye la opinión personal de la elaboración de este proyecto en un marco investigador y en el extranjero.

El incesante crecimiento de los comentarios acerca de productos que se venden hoy en día a través de Internet, produce la masificación de este tipo de recurso que puede ser útil tanto para clientes como para vendedores de los mismos. Sin embargo, sin un tratamiento eficaz de estas cantidades enormes de comentarios, éstos se vuelven casi inservibles. Basándonos en este escenario, este proyecto fin de carrera, el cual se ha realizado en la Universidad Técnica de Braunschweig (Alemania), trató de colaborar con una de las tareas propias de las técnicas que extraen información de estas ingentes cantidades de datos. En concreto, se persiguieron los objetivos del descubrimiento de nuevas relaciones semánticas entre términos, extraídas de los comentarios acerca de teléfonos inteligentes y de las especificaciones técnicas de los mismos. Para ello, se propuso un proceso manual, con el que se demostró que es prácticamente irrealizable la tarea a mano. Además de un proceso automático, para el que se adaptaron al problema las técnicas clásicas de recuperación de información, tales como LSI, PLSI, y LDA. Este tipo de tareas puede ayudar y mejorar a los procesos de minería de datos u opiniones, la representación del conocimiento, los procesos de aprendizaje automático, la mejora de los procesos de recuperación de información a través de la expansión de la consulta, etc.

En primer lugar, corroborando lo que se preveía en cuanto al análisis manual y a la propuesta del proceso automático, de inmediato se comprobó que es de esta segunda manera como mejor se llevan a cabo las tareas a la hora de tratar una ingente cantidad de datos, como ha sido nuestro caso. Es por eso que el objetivo latente de mejora de prestaciones está cumplido. Además, cabe destacar que en cuanto a los métodos automáticos analizados en los diferentes marcos propuestos, han satisfecho ampliamente las expectativas, unos más que otros. Éste es el caso de LDA, cuyo análisis demostró que por algo es una de las actuales bases de líneas de investigación de este tipo y fundamento de nuevas técnicas de recuperación de información basadas en su algoritmo. El método LSI, aunque desarrollado hace más tiempo, sigue conservando su carácter pionero, dado que cuando funciona bien, lo hace hasta mejor que LDA en casos muy puntuales. Por el contrario, queda constancia de que, o bien la elección de la implementación de PLSI no ha sido la adecuada (aunque fuera ésta la única que funcionaba en todos los escenarios), o bien la técnica no se ajusta a las predicciones iniciales, que preveían una evolución en la obtención de resultados, proporcional a la época de desarrollo de las técnicas y dado que PLSI se hizo con la base de LSI y así sucesivamente.

En cuanto a la decisión tomada en este proyecto fin de carrera de dividir los comentarios en secciones y párrafos, además de considerar las indicaciones iniciales propuestas por el director de trabajar con los comentarios completos (documentos), ha sido el mayor logro. Aunque sí que se cumpla la supuesta evolución para el escenario de los documentos, se demostró claramente que es en este escenario donde los métodos obtienen los peores resultados, tanto es así que LSI ni siquiera obtiene resultados reseñables en este escenario. Además, en cuanto a la opción de incluir o no los *comentarios extra* donde se agruparan los términos extraídos de las especificaciones técnicas, ha sido lamentablemente baldía, dado que si existe mejora alguna en algunos casos, en otros es completamente inapreciable y a veces hasta perjudicial, ya que altera la concurrencia de los términos y esto provoca algún resultado atípico.

La elección del parámetro k ha sido acertada, porque se ha podido comprobar la eficiencia en distintos supuestos, todos cercanos a las mejores prestaciones de cada método. Teniendo en cuenta que es imposible hacer un análisis de esta magnitud en todos y cada uno de los valores posibles de este parámetro (desde 1 hasta 200), la decisión de tomar 50 como menor valor, donde quizá se requiera una mayor subdivisión, y 200 como cota máxima, donde quizá el refinamiento ya no mejora nada en cuanto a los resultados, ha resultado correcta.

Por último, dado que la selección de las características principales fue acorde al número de términos iniciales que posee cada una, éste es directamente proporcional al número de términos extraídos en casi cualquier escenario, siendo *battery* la que peores resultados obtiene. Sin embargo, la influencia de la popularidad de los términos, es decir, lo mucho o poco que aparecen en la colección de comentarios, como sucede en el caso de *organizer*, cuyos términos tienen una frecuencia muy elevada, también ha sido relevante, dado que *multimedia* posee muchos más y no ha extraído un número significativo mayor de términos, de hecho en la mayoría de los casos ha sido un poco menor.

6.1. Problemas encontrados

El problema inicial encontrado fue la difícil comprensión inicial del problema, dado que sin mucha idea de lo que eran y cómo funcionaban las técnicas de recuperación de información, en concreto LSI, cuyos resultados tampoco eran muy descriptivos, retrasó las expectativas de una pronta finalización.

Como se ha reiterado en varias ocasiones, se trata de una investigación, y muchas veces ni siquiera el investigador sabe hacia dónde se dirige si no tiene claro el marco en el que realiza la investigación. Por eso, la parte más dura fue la de recabar información, técnicas, estudios, métodos, procedimientos o análisis similares a lo que se estaba investigando y desarrollando, porque muchas veces se decidía “sobre la marcha” la dirección de la investigación, dado que ni siquiera el director tenía muy claro cuál era camino a seguir, pero que satisfactoriamente se centró en lo que recoge este documento.

Esto sumado al hecho de realizar el proyecto en el extranjero, con los problemas de comunicación que ello implica, a pesar de que se haya desarrollado enteramente en inglés,

hasta que uno no está familiarizado con los términos empleados para describir y nombrar a los elementos, cuando la traducción literal no existe o no es lo suficientemente intuitiva

Por último, otro de los grandes problemas sufridos apareció cuando, después de haber propuesto, seleccionado y probado en entornos reducidos la implementación del método PLSI, éste no funcionaba en grandes cantidades de datos, lo que suponía la pérdida de gran cantidad de información, por lo que se optó por modificar el código mínimamente para que éste se adaptara a la ingente cantidad de datos existente al realizar divisiones en secciones y párrafos.

6.2. Cronograma

En esta sección se presenta el tiempo aproximado de las ejecuciones de cada tarea, como se aprecia en la **Tabla 6-1**.

	2010	2011												2012			
	dic	ene	feb	mar	abr	may	jun	jul	ago	sep	oct	nov	dic	ene	feb	mar	abr
documentación	■	■	■	■	■		■				■			■			
análisis manual		■	■														
test LSI				■													
análisis LSI					■		■	■	■								
test PLSI										■							
análisis PLSI											■	■					
test LDA													■				
análisis LDA														■	■		
test métodos propios																■	
implementación y construcción de matrices		■	■	■	■				■	■			■			■	
escritura PFC				■	■			■	■			■			■	■	■

Tabla 6-1. Cronograma de la realización del proyecto fin de carrera.

Se puede observar como aproximadamente en mayo de 2011, debido a problemas técnicos con el equipo, se produjo un parón en el desarrollo. Además, no se incluyen aquí todas las tareas de aprendizaje de programas, protocolos, reuniones, organización, etc., que se compatibilizaron con la estancia de Erasmus en Alemania. En general, existe una progresión desde 2010 hasta 2012 en la que el esfuerzo y el trabajo dedicado aumentan de manera proporcional al avance del proyecto, por eso los primero análisis han tenido una mayor dedicación.

6.3. Trabajo futuro

Esta investigación abre un abanico de posibilidades de desarrollo, tanto de implementaciones basadas en estos métodos como en otros, pero que busquen el objetivo planteado aquí. Aunque lamentablemente las propuestas planteadas en este proyecto no han podido ser probadas de manera satisfactoria, queda abierta la posibilidad de un trabajo futuro basado en las propuestas aquí realizadas, como son los casos de la implementación en un entorno paralelo, ya no solo del método de cálculo de la concurrencia y similitud de contextos, sino de cualquier método que trabaje con datos y técnicas como los propuestos aquí. La paralelización a nivel de código de este tipo de métodos puede suponer un incremento tanto en las prestaciones como en el tiempo empleado en sus respectivas ejecuciones.

Por otro lado, se ha llegado a la obtención de la mayor cantidad de datos reseñables para el valor 100 del parámetro k. Esto hace pensar que alrededor de este valor podrían hacerse nuevas pruebas y aproximaciones para acotar con mayor grado de nitidez dónde actúan mejor los métodos presentados. Además, se podría probar en otro tipo de entorno para comprobar si los resultados se mantenían en la misma línea. La inclusión o uso de ontologías podría dar lugar a mejores resultados, siempre manteniéndose en la línea de métodos sin supervisión, que no necesitan de recursos externos más allá de los presentados.

Además de que directamente con los resultados obtenidos aquí, que se deduce de ellos que se pueden emplear estas técnicas con la finalidad planteada, la inclusión del proceso de obtención de términos semánticamente relacionados aquí descrito en sistemas de aprendizaje automático, minería de opiniones o expansión de consultas, supondría una alternativa a los métodos que actualmente se utilizan para llevar a cabo estas tareas.

6.4. Opinión personal

Este proyecto no puede entenderse si no es en el contexto de Erasmus, en el que he aprendido que el factor tiempo es algo esencial. A la par que me instruía en idiomas conocía gente maravillosa, lugares increíbles, culturas autóctonas y vivía experiencias inolvidables.

Se puede decir que un trabajo de este tipo depende en muy alto grado de los datos iniciales, los cuales si se modifican pueden dar lugar a resultados muy distintos, además de que se sabe que la investigación está viva, donde una investigación similar puede llevarse a cabo en cualquier parte del mundo, y hacer efímero el sudor y lágrimas invertidos en el desarrollo de la misma, pero que da la gratificación de intentar mejorar el presente para luchar por un futuro mejor. En general, me siento orgulloso de haberme metido en la piel de un investigador, principalmente porque a mis ojos, el proyecto desarrollado “tiene pinta” de investigación, y porque he sufrido en mi piel las horas invertidas en hacer algo muy específico, concreto y apenas abordado antes. De la misma forma que se trata sobre un tema de inminente actualidad, como es la extracción de semántica a partir de grandes cantidades de información, como se supone que sucederá en la futura Web Semántica, para la que investigaciones como ésta pueden ser de gran utilidad, aunque todavía queda mucho camino por explorar.

Bibliografía

1. Ohler, J., *The semantic web in education.* EDUCAUSE Quarterly, 2008. **31**(4): p. 7-9.
2. Hu, M. and B. Liu, *Mining and summarizing customer reviews*, in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2004, ACM: Seattle, WA, USA. p. 168-177.
3. Deerwester, S.C., et al., *Indexing by Latent Semantic Analysis.* Journal of the American Society of Information Science, 1990. **41**(6): p. 391-407.
4. Hofmann, T., *Probabilistic latent semantic indexing*, in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. 1999, ACM: Berkeley, California, United States. p. 50-57.
5. Blei, D.M., A.Y. Ng, and M.I. Jordan, *Latent dirichlet allocation.* J. Mach. Learn. Res., 2003. **3**: p. 993-1022.
6. Hu, M. and B. Liu, *Mining opinion features in customer reviews*, in *Proceedings of the 19th national conference on Artifical intelligence*. 2004, AAAI Press: San Jose, California. p. 755-760.
7. Lopez-Fernandez, A., T. Veale, and P. Majumder, *Feature Extraction from Product Reviews using Feature Similarity and Polarity*. 2009.
8. Liu, B., M. Hu, and J. Cheng, *Opinion observer: analyzing and comparing opinions on the Web*, in *Proceedings of the 14th international conference on World Wide Web*. 2005, ACM: Chiba, Japan. p. 342-351.
9. Popescu, A.-M. and O. Etzioni, *Extracting product features and opinions from reviews*, in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. 2005, Association for Computational Linguistics: Vancouver, British Columbia, Canada. p. 339-346.
10. Scaffidi, C., *Application of a probability-based algorithm to extraction of product features from online reviews*. 2006.
11. Scaffidi, C., et al., *Red Opal: product-feature scoring from reviews*, in *Proceedings of the 8th ACM conference on Electronic commerce*. 2007, ACM: San Diego, California, USA. p. 182-191.
12. Miao, Q., Q. Li, and R. Dai, *An integration strategy for mining product features and opinions*, in *Proceeding of the 17th ACM conference on Information and knowledge management*. 2008, ACM: Napa Valley, California, USA. p. 1369-1370.
13. Somprasertsri, G. and P. Lalitrojwong, *Automatic product feature extraction from online product reviews using maximum entropy with lexical and syntactic features*, in *Information Reuse and Integration, 2008. IRI 2008. IEEE International Conference*. 2008. p. 250 -255.
14. Guo, H., et al., *Product feature categorization with multilevel latent semantic association*, in *Proceeding of the 18th ACM conference on Information and knowledge management*. 2009, ACM: Hong Kong, China. p. 1087-1096.
15. Kim, W.Y., et al., *A method for opinion mining of product reviews using association rules*, in *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*. 2009, ACM: Seoul, Korea. p. 270-274.
16. Khoo, C.S.G. and J.-C. Na, *Semantic relations in information science.* Annual Review of Information Science and Technology, 2006. **40**(1): p. 157-228.

17. Sahami, M. and T.D. Heilman, *A web-based kernel function for measuring the similarity of short text snippets*, in *Proceedings of the 15th international conference on World Wide Web*. 2006, ACM: Edinburgh, Scotland. p. 377-386.
18. Takale, S.A. and S.S. Nandgaonkar, *Measuring semantic similarity between words using web search engines*, in *Proceedings of the 16th international conference on World Wide Web*. 2007, ACM: Banff, Alberta, Canada. p. 757-766.
19. Bollegala, D., Y. Matsuo, and M. Ishizuka, *Measuring semantic similarity between words using web search engines*, in *Proceedings of the 16th international conference on World Wide Web*. 2007, ACM: Banff, Alberta, Canada. p. 757-766.
20. Church, K.W. and P. Hanks, *Word association norms, mutual information, and lexicography.* Comput. Linguist., 1990. **16**(1): p. 22-29.
21. Iosif, E. and A. Potamianos, *Unsupervised Semantic Similarity Computation using Web Search Engines*, in *Web Intelligence, IEEE/WIC/ACM International Conference*. 2007. p. 381 -387.
22. Cilibrasi, R.L. and P.M.B. Vitanyi, *The Google Similarity Distance.* IEEE Trans. on Knowl. and Data Eng., 2007. **19**(3): p. 370-383.
23. Lin, D., *An Information-Theoretic Definition of Similarity*, in *Proceedings of the Fifteenth International Conference on Machine Learning*. 1998, Morgan Kaufmann Publishers Inc. p. 296-304.
24. Titov, I. and R. McDonald, *Modeling online reviews with multi-grain topic models*, in *Proceeding of the 17th international conference on World Wide Web*. 2008, ACM: Beijing, China. p. 111-120.
25. Pangos, A., et al., *Combining Statistical Similarity Measures for Automatic Induction of Semantic Classes*. 2005.
26. Tous, R. and J. Delgado, *A vector space model for semantic similarity calculation and OWL ontology alignment*, in *IN DEXA 2006*. 2006.
27. Budanitsky, A. and G. Hirst, *Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures*, in *IN WORKSHOP ON WORDNET AND OTHER LEXICAL RESOURCES, SECOND MEETING OF THE NORTH AMERICAN CHAPTER OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*. 2001.
28. Stoutenburg, S., J. Kalita, and S. Hawthorne, *Extracting semantic relationships between Wikipedia articles*, in *In Proc. 35th International Conference on Current Trends in Theory and Practice of Computer Science*. 2009: Spindelruv Mlyn, Czech Republic.
29. King, J. and V. Satuluri, *Extracting Semantic Relations Using Dependency Paths.* Unpublished.
30. Spagnola, S. and C. Lagoze, *Edge dependent pathway scoring for calculating semantic similarity in ConceptNet*, in *Proceedings of the Ninth International Conference on Computational Semantics*. 2011, Association for Computational Linguistics: Oxford, United Kingdom. p. 385-389.
31. Li, Y., Z.A. Bandar, and D. McLean, *An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources.* IEEE Trans. on Knowl. and Data Eng., 2003. **15**(4): p. 871-882.
32. Han, L., et al., *ADSS: an approach to determining semantic similarity.* Adv. Eng. Softw., 2006. **37**(2): p. 129-132.
33. Fellbaum, C., *WordNet: An Electronic Lexical Database*. 1998: Bradford Books.
34. Liu, H. and P. Singh, *ConceptNet: A Practical Commonsense Reasoning Tool-Kit.* BT Technology Journal, 2004. **22**(4): p. 211-226.
35. Mahinovs, A. and A. Tiwari, *Text classification method review*, R. Roy and D. Baxter, Editors. 2007, Cranfield University.
36. Lovins, J.B., *Development of a Stemming Algorithm.* Mechanical Translation and Computational Linguistics, 1968. **11**: p. 22-31.

37. Porter, M.F., *An algorithm for suffix stripping*, in *Readings in information retrieval*, J. Karen Sparck and W. Peter, Editors. 1997, Morgan Kaufmann Publishers Inc. p. 313-316.
38. Hull, D.A., *Stemming Algorithms - A Case Study for Detailed Evaluation.* Journal of the American Society for Information Science, 1996. **47**: p. 70-84.
39. Hasan, M.M. and Y. Matsumoto, *Document Clustering: Before and After the Singular Value Decomposition.* Joho Shori Gakkai Kenkyu Hokoku, 1999. **99**(95(NL-134)): p. 47-54.
40. Berry, M.W., S.T. Dumais, and G.W. O'Brien, *Using Linear Algebra for Intelligent Information Retrieval.* SIAM Review, 1995. **37**: p. 573-595.
41. Tu, N.C., *Hidden Toopic Discovery Toward Classification and Clustering in Vietnamese Web Documents*. 2008, Vietnam National University: Hanoi.
42. Kakkonen, T., et al. *Comparison of Dimension Reduction Methods for Automated Essay Grading*. 2008.
43. Hofmann, T., *Probabilistic Latent Semantic Analysis*, in *Proceedings of Uncertainty in Artificial Intelligence, UAI'99*. 1999: Stockholm.
44. Kontostathis, A. and W. Pottenger, *Detecting patterns in the LSI term-term matrix*. 2002.

Anexo 1