**UNIVERSIDAD CENTRAL "MARTA ABREU" DE LAS VILLAS**
**FACULTAD DE MATEMÁTICA, FÍSICA Y COMPUTACIÓN**


Trabajo para optar por el Título Académico

Máster en Ciencia de la Computación

# Segmentación y detección de tópicos enfocado a la minería de opinión

Autor: Ing. Carmen Torres López

Tutor: Dra. Leticia Arco García

2016

*Dedicado a mi madre Carmen y a mi esposo Humberto*

*A mi tutora Dra. Leticia Arco García por haberme dado la oportunidad de investigar en el área de la minería de textos, por el tiempo dedicado, por sus enseñanzas. Sus ideas y orientación fueron fundamentales en la realización de esta tesis.*

*A Mario por su colaboración y su asesoramiento en el área de minería de opinión.*

*A Adrián por sus consejos e ideas en los intercambios realizados.*

*A Ariel por sus aclaraciones sobre las extensiones de los algoritmos Estrella.*

*Al Dr. José Eladio Medina Pagola por compartir sus experiencias sobre el área de la segmentación del texto por tópicos.*

*Al profesor Dr. Dánel Sánchez Tarrago, por la supervisión de las validaciones estadísticas de la investigación.*

*A todos los profesores de la maestría de Ciencia de la Computación por su dedicación y entrega en cada uno de los cursos impartidos.*

*A todos mis compañeros de Desoft, por el apoyo que me dieron durante el transcurso de toda la investigación.*

*A mis compañeros de curso por compartir conmigo noches de estudio y a mis amigos por su constante preocupación.*

*A mi madre, por toda la ayuda que me ha dado estos dos años, por estar siempre cuando la necesito, por ser mi ejemplo y mi guía.*

*A mi esposo Humberto, por todo su apoyo y comprensión, por darme ánimo y confiar en mí en los momentos más difíciles.*

# Resumen

La detección y seguimiento de tópicos es un área que ha sido aplicada en disímiles campos, entre ellos, en el análisis de sentimientos. PosNeg Opinion es una herramienta desarrollada en el CEI-UCLV que detecta de manera no supervisada la polaridad de opiniones mostrando excelentes valores de precisión y exactitud; sin embargo, solo es capaz de clasificar las opiniones expresadas en una oración o la opinión que expresa el texto en su totalidad, y por tanto no realiza un análisis del sentimiento por tópicos. El objetivo de la presente investigación consiste en desarrollar una herramienta que permita de manera no supervisada y efectiva descubrir tópicos en las opiniones. Los resultados obtenidos son: el diseño de un esquema para la segmentación y detección de tópicos en opiniones que propone identificar unidades textuales, pre-procesar el texto, representar las unidades textuales, aplicar técnicas de segmentación, representar la colección de segmentos, agrupar los segmentos y etiquetar los grupos de segmentos; el marco de trabajo OpinionTopicDetection que aplica el esquema propuesto para el descubrimiento de tópicos en opiniones; la instanciación del marco de trabajo con la aplicación OpinionTD para el descubrimiento de tópicos; un analizador léxico para pre-procesar los textos de opiniones y un algoritmo para etiquetar los grupos de tópicos basado en la identificación de sustantivos y sus similitudes semánticas. La evaluación realizada de la propuesta arrojó una precisión de 0.74 y exhaustividad 0.86, mostrando un resultado satisfactorio de la investigación desarrollada en el descubrimiento de tópicos en opiniones.

# Abstract

Topic detection and tracking is a research field that has been applied in several fields, one of them is sentiment analysis. PosNeg Opinion is a tool developed in the CEI-UCLV that detects unsupervised polarity of opinions showing excellent values of precision and recall; however, it is only able to classify the opinions expressed in a sentence or the opinion that text expresses as a whole, and therefore does not perform sentiment analysis by topic. The objective of this research is to develop a tool that allows unsupervised and effective discovering of topics in opinions texts. The main results are: the design of a scheme for segmentation and detection of topics in opinions that aims to identify textual units, text pre-processing, textual units representation, applying segmentation techniques, represent the collection of segments, segments clustering and labeling the groups of segments; OpinionTopicDetection framework which applies the proposed scheme for discovering topics in opinions; instantiation of the framework with OpinionTD desktop application for topics discovery; a lexical analyzer to pre-process the opinions texts and an algorithm for labeling groups of topics based on identifying nouns and their semantic similarities. The assessment of the proposal showed precision of 0.74 and recall of 0.86, showing a successful outcome of the research conducted in the topics discovery in opinions.

# Introducción

En la actualidad se generan constantemente grandes volúmenes de datos, debido, entre otras causas, al desarrollo de tecnologías como la computación en la nube, la Internet de las cosas, las redes sociales y la computación móvil. De ahí que los datos en la World Wide Web y en repositorios locales de diferentes entidades a nivel global se presentan actualmente en el orden de quintillones de bytes[1]. La mayor parte de los datos son no estructurados y su formato más común de almacenamiento es el texto (Bessis & Dobre 2014), razón por la que durante varias décadas se han dedicado esfuerzos investigativos en el desarrollo de herramientas computacionales que permitan manejar tales cantidades de datos textuales de forma rápida y efectiva; sin embargo, lograrlo constituye un desafío. Por ejemplo, resulta muy difícil que alguna persona posea el tiempo para leer toda la información disponible sobre un tema dado, por tal motivo, podría ser mucho más efectivo leer y buscar por temas específicos. De ahí que uno de los campos de investigación que ha tomado la iniciativa en el procesamiento de datos textuales es la Detección y Seguimiento de Tópicos (Topic Detection and Tracking; TDT).

Este campo surgió en 1996 con la llegada constante de servicios de noticias y el surgimiento de sistemas automáticos de discursos textuales que monitoreaban programas seleccionados de televisión, radio y noticias de difusión Web (Allan 2002; Wayne 2000; Allan et al. 1998). En 1997 se realizó un estudio piloto que sentó las bases esenciales de TDT e identificó las tecnologías potenciales para automáticamente organizar textos de noticias[2]. Durante 1998 y 1999, la investigación sobre TDT floreció con nuevas y desafiantes tareas. La investigación de TDT fue realizada bajo el programa de Detección de Información, Extracción y Resumen en varios idiomas (Translingual Information Detection, Extraction, and Summarization; TIDES) de DARPA. El objetivo del programa de TDT fue originalmente desarrollar tecnologías que buscan, organizan y estructuran materiales textuales orientados a noticias de una variedad de medios de difusión. Con el surgimiento de las redes sociales en el año 2006 como Twitter y Facebook, TDT ha sido una técnica importante para realizar descubrimiento del conocimiento en grandes colecciones de páginas web o microblogs donde se publican billones de datos por los usuarios (Huang et al. 2013)

[1] Cada día se crean 2.5 quintillón de bytes de datos (1 quintillón = $10^{30}$) (Bessis & Dobre 2014)

[2] Este estudio fue desarrollado por un conjunto de investigadores de la Agencia de Proyectos Avanzados de Investigación y Defensa del gobierno de Estados Unidos (Defense Advanced Research Projects Agency; DARPA), la Universidad de Carnegie Mellon (Carnegie Mellon University; CMU), la compañía norteamericana Dragon Systems (Dragon), y la Universidad de Massachusetts en Amherst (University of Massachusetts - Amherst; UMass).

(Perez-Tellez et al. 2010) (Hattori & Nadamoto 2013) (Wang & Wang 2013) (Yin et al. 2013). Se han realizado estudios para identificar en tiempo real tópicos emergentes y subtópicos referidos a noticias y otros dominios, publicados en los mensajes enviados por los usuarios en Twitter (tweets) a través de nuevas técnicas de detección (Cataldi et al. 2010) (Becker et al. 2011) (Xie et al. 2012) (Koike et al. 2013) (Panem et al. 2014) (Petkos, Aiello, et al. 2014). También se ha destacado la importancia de la detección de tópicos en las miles de publicaciones de blogs existentes en la web; una de las áreas en la que es ventajosa esta tarea de investigación es para la Inteligencia de Negocios (Glass & Colbaugh 2010). Los dispositivos móviles son muy utilizados en la actualidad, algunos estudios enfatizan el hecho que el conocimiento de tópicos populares a partir de microblogs por parte del usuario puede brindarle información de interés como su ubicación, actividades principales y relación social (Han et al. 2010).

TDT se ha utilizado además en contextos académicos e investigativos, por ejemplo para analizar colecciones de documentos de artículos investigativos biomédicos; así como la identificación de subtópicos de conceptos médicos usando como fuente un tesauro correspondiente al dominio de la Medicina (Ye et al. 2006) (Berlanga-llavori et al. 2008). También se han estudiado técnicas para identificar tópicos en los mensajes que publican diferentes comunidades en línea sobre temas de salud (Lu et al. 2013).

La expansión de sistemas de mensajería instantánea ha sido una motivación para el análisis de mensajes de conversación como los mensajes de charla (chat messages), donde se han aplicado técnicas de detección de tópicos (Dong et al. 2006). Otro dominio de aplicación es para detectar tópicos en mensajes enviados por correo electrónico (Wattenhofer & Cselle 2007).

Muchas organizaciones utilizan las revisiones y opiniones de productos en la red para la toma de decisiones (Li & Cardie 2013). En este contexto, identificar tópicos resulta de gran importancia para determinar sobre cuál tema los usuarios están emitiendo sus criterios (Lin et al. 2012) (Jiang et al. 2011) (Cai et al. 2008). Se ha afirmado actualmente que los problemas claves en la investigación de la opinión pública emitida en Internet consisten en cómo distinguir y clasificar los tópicos referentes al público, cómo descifrar las actitudes de las personas frente a eventos sociales y cómo analizar y capturar la volatilidad de temas sociales populares (Ren & Han 2014). También se han estudiado técnicas para descubrir tópicos y analizar sentimientos en tweets en idioma español (Fernández & Núñez 2013); así como la identificación de subtópicos en una opinión (Gangemi et al. 2014). Igualmente se propusieron metodologías para detectar tendencias

de tópicos en la web, a partir de la detección de estos en documentos de opiniones para un período de tiempo (Dueñas et al. 2013). Precisamente, resulta de gran interés y constituye una motivación en esta investigación el desarrollo de aquellas técnicas de detección de tópicos aplicadas al contexto del análisis de opiniones. En este dominio un tópico es considerado *"…algo sobre lo que los participantes debaten o expresan sus opiniones*".

El fácil acceso y el desarrollo de nuevas aplicaciones en Internet han motivado a muchas personas a publicar y compartir sus experiencias, conocimientos, opiniones y emociones. De esta forma, se ha dedicado especial atención a explorar y descubrir la información subjetiva generada por el usuario en el área de investigación nombrada Minería de Opinión, conocida también por Análisis de Sentimiento [12] [13]. Las tareas fundamentales de este campo están dirigidas a la detección de opiniones, la polaridad y el reconocimiento de emociones (Moens et al. 2014).

Las opiniones de usuarios no expertos pueden servir como complemento de puntos de vistas publicados en medios de noticias; y las valoraciones sobre productos y servicios pueden tener gran impacto económico tanto para consumidores como organizaciones, por ejemplo, en estrategias de negocios, actividades de mercado, etc. Por ello, la tarea de detección de la polaridad está siendo muy estudiada en la actualidad (Lin 2011) (Thompson 2014) (Dueñas et al. 2013) (Toh et al. 2014). Específicamente se han trabajado dos puntos de vistas: modelar la detección de la polaridad como un problema de clasificación donde el objetivo es determinar si la polaridad es positiva o negativa, o modelar la detección de la polaridad como un problema de extracción de información, en el cual se recuperan expresiones de sentimiento y se evalúan (Moens et al. 2014).

De forma general, la minería de opinión se ha trabajado en tres niveles fundamentales: documento, oración y aspecto (Zhang & Liu 2014). En el nivel de documento se obtiene una valoración general de la opinión pero no se ofrecen detalles de las preferencias de los usuarios respecto a un aspecto u otro. El nivel de oración es un nivel más detallado en el que la polaridad de la oración puede ser positiva, negativa o neutral pero solo tienen en cuenta las palabras de opinión presentes en el texto. Por último, el nivel de aspecto se define a partir de los atributos de las entidades, para los que se identifican opiniones negativas o positivas; de esta forma se pueden conocer las diferentes opiniones acerca de los aspectos que identifican las entidades que se mencionan en un texto. Esta última clasificación se ha estudiado en una rama de la minería de opinión conocida por minería de opinión basada en aspectos (Aspect Based Sentiment Analysis; ABSA); la cual se define como el proceso de "*extraer y resumir opiniones de personas expresadas sobre entidades y aspectos*"

(Zhang & Liu 2014). Una de sus tareas es la extracción de términos de aspectos (Pavlopoulos 2014), la cual ha sido abordada con distintos enfoques, uno de ellos es la detección de tópicos.

Varios estudios sugieren combinar el análisis de sentimiento y la detección de tópicos, y así beneficiar el funcionamiento de los sistemas de minería de opinión (Zhang & Liu 2014; Hattori & Nadamoto 2013; Titov & Mcdonald 2008; Cambria et al. 2013; Pang & Lee 2008; Dueñas et al. 2013). Por ejemplo, un documento puede ofrecer opiniones acerca de múltiples tópicos. En tales casos, es importante identificar los tópicos y separar las opiniones asociadas a cada tópico. Las opiniones y sentimientos no ocurren solo a nivel de documento, ni están limitados a un único objetivo. Un documento puede contener opiniones negativas y positivas a través de uno o más tópicos (Cambria et al. 2013).

Teniendo en cuenta la actualidad e importancia de la minería de opinión, así como los intereses expresados por algunos usuarios en el tema, especialistas del laboratorio de Inteligencia Artificial del Centro de Estudios Informáticos (CEI) de la Universidad Central "Marta Abreu" de Las Villas (UCLV), se dieron a la tarea de elaborar la aplicación PosNeg Opinion para la detección no supervisada de la polaridad de opiniones en Inglés y Español (Liu 2010; Amores 2013). La aplicación implementa un esquema general compuesto por cinco etapas. PosNeg Opinión permite que el usuario analice un gran cúmulo de opiniones de manera sencilla. Esta aplicación puede utilizarse como un módulo para una aplicación más general de minería de opinión pues resuelve una de las fases de este proceso y es fácilmente reutilizable, además, se puede comunicar con otras aplicaciones mediante ficheros XML. Fue desarrollado completamente en JAVA, por lo que es multiplataforma. Necesita como entrada un fichero XML con todas las opiniones a analizar y como salida muestra cuántas fueron positivas y cuántas negativas. A petición del usuario también retorna el porcentaje de las opiniones negativas y positivas así como una lista con las opiniones negativas y otra con las opiniones positivas, además de destacar cuales fueron las opiniones de mayor puntuación en cada caso (positivas/negativas). Desafortunadamente, PosNeg Opinion solo permite detectar la polaridad de las opiniones a nivel de oración y de documentos, por tal motivo, no es capaz de calcular la polaridad por aspectos o tópicos, limitándose de esta forma la futura toma de decisiones a partir de los resultados que la herramienta ofrece. Esta desventaja de PosNeg Opinion constituye una problemática a la cual aún no se le ha dado respuesta, lo cual justifica el planteamiento del **problema de investigación** siguiente:

La herramienta PosNeg Opinion detecta de manera no supervisada la polaridad de opiniones mostrando excelentes valores de precisión y exactitud; sin embargo, solo es capaz de clasificar las opiniones expresadas en una oración o la opinión que expresa el texto en su totalidad, y por tanto no realiza un análisis del sentimiento por tópicos.

El **objetivo general** de la investigación consiste en desarrollar una herramienta que permita de manera no supervisada y efectiva descubrir tópicos en las opiniones. Este se desglosa en los siguientes **objetivos específicos**:

1. Identificar, a partir del estudio de los principales métodos de segmentación y detección de tópicos, aquellos elementos aplicables de manera efectiva cuando los textos a analizar son opiniones.
2. Establecer un esquema general que permita la segmentación y detección de tópicos, especificando los elementos principales a considerar en cada etapa.
3. Aplicar de manera integrada aquellas herramientas disponibles para el procesamiento textual y la detección de tópicos que permitan la implementación del esquema propuesto.

Las **preguntas de investigación** planteadas son:

- ¿Cuáles métodos de segmentación y detección de tópicos son aplicables de manera efectiva en colecciones de opiniones?
- ¿Qué elementos del procesamiento textual considerar y cómo integrarlos para obtener un esquema que permita descubrir tópicos en opiniones?
- ¿Qué herramientas libres y de código abierto facilitan la implementación de un esquema de descubrimiento de tópicos en opiniones y cómo garantizar la integración entre las mismas?

Después de haber realizado el marco teórico se formuló la siguiente **hipótesis de investigación** como presunta respuesta a las preguntas de investigación: La detección de tópicos utilizando el agrupamiento de segmentos previamente identificados, contribuye a un efectivo análisis de sentimientos por tópicos de opiniones.

Para lograr los objetivos trazados y demostrar la hipótesis establecida se acometieron las **tareas de investigación** siguientes:

1. Análisis de los principales métodos dentro del campo de investigación de la detección y seguimiento de tópicos.
2. Estudio de los principales métodos para las tareas de segmentación y detección de tópicos.
3. Estudio de las principales técnicas de minería de opinión, particularizando en las diferencias entre detección de la polaridad de cada oración, documento y basada en aspectos.
4. Diseño de un esquema que integre técnicas de segmentación y detección con el objetivo de descubrir tópicos en opiniones y contribuir así al cálculo de la polaridad por tópicos.
5. Identificación y estudio de las principales herramientas para la minería de textos y selección de aquellas que contribuyan a desarrollar el esquema.
6. Diseño e implementación de un marco de trabajo que permita aplicar el esquema para el descubrimiento de tópicos en opiniones.
7. Diseño e implementación de una aplicación que instancie el marco de trabajo.
8. Búsqueda de colecciones de opiniones etiquetadas por tópicos que permitan la validación externa de la propuesta que se presenta.
9. Evaluación de las etapas del esquema propuesto y de los tópicos obtenidos.

El **valor teórico** de la investigación radica en la concepción y definición de un esquema para la segmentación y detección de tópicos que contribuye al análisis de sentimiento por tópicos, así como el análisis de los métodos a considerar en cada una de las etapas que lo componen, y cómo interrelacionarlos. Resulta novedoso el método definido para etiquetar los grupos de tópicos.

El **valor práctico** se relaciona con el marco de trabajo OpinionTopicDetection que implementa el esquema diseñado para el descubrimiento de tópicos en opiniones, la selección de las herramientas a utilizar en cada etapa del esquema propuesto, así como la aplicación OpinionTD 1.0 que permite identificar los tópicos de lo que opinan los usuarios en diferentes dominios, como por ejemplo restaurantes, hoteles, etc. Esta aplicación puede utilizarse como un módulo de una aplicación más general de minería de opinión, es además fácilmente reutilizable, debido a que está construida sobre OpinionTopicDetection. Este marco de trabajo puede ser usado, además, como una biblioteca por otras aplicaciones para el análisis textual, debido a que las funcionalidades que reutiliza de otras bibliotecas se pueden instanciar de forma independiente.

La tesis está estructurada en tres capítulos. El Capítulo 1 aborda los principales elementos del campo TDT, así como las clasificaciones y métodos más destacados hasta el momento referentes

a las tareas de segmentación y detección de tópicos. De esta forma se generaliza un esquema por etapas para cada una de dichas tareas. En el Capítulo 2 se propone un esquema general para realizar análisis de sentimiento por tópicos, este contiene los aspectos más importantes a tener en cuenta por cada etapa así como las herramientas seleccionadas para su desarrollo. El Capítulo 3 describe los elementos principales del diseño e implementación del marco de trabajo OpinionTopicDetection para la segmentación y detección de tópicos en opiniones, así como la aplicación OpinionTD desarrollada sobre el marco de trabajo propuesto para descubrir tópicos en opiniones. Además, se presentan los resultados de los experimentos realizados. Este documento culmina con las conclusiones, recomendaciones, referencias bibliográficas y los anexos.

# Capítulo 1 Acerca de la segmentación y detección de tópicos

Dos tareas fundamentales del campo TDT son la segmentación y la detección de tópicos, las cuales proponen técnicas para identificar tópicos en segmentos de textos y obtener grupos que representan tópicos, respectivamente. A continuación se describirán los principales componentes de TDT y los distintos enfoques de tópicos que existen; luego se presentarán las diferentes formas de representación textual utilizadas para encontrar tópicos. Además, se expondrán las clasificaciones de los diferentes métodos de segmentación y detección, y se mencionarán ejemplos de algunos de ellos.

## 1.1 Detección y seguimiento de tópicos

TDT es referido de forma general como "*las técnicas automáticas para encontrar material relacionado tópicamente en datos textuales*" (Wayne et al. 2007). El programa de investigación de TDT se propuso idear algoritmos completamente automáticos para determinar la estructura tópica del lenguaje humano. De esta forma declararon que los algoritmos deberían ser independientes del dominio, fuente y lenguaje. Para estos algoritmos propusieron cinco tareas, cada una es vista como un componente cuya solución ayuda a guiar el problema de organizar los documentos por tópicos (Allan 2002) (Wayne 2000). En el Anexo 1 se muestran las nociones básicas de cada tarea.

La segmentación de historias parte de una muestra de documentos textuales, y su objetivo es detectar automáticamente los límites entre las historias. Algunas técnicas propuestas para solucionar este problema consisten en buscar cambios en el vocabulario, palabras, frases y pausas que ocurran cerca de los límites de la historia para distinguir conjuntos de rasgos ubicados en el inicio, medio o final de una historia. La segmentación constituye la tarea primaria de TDT y tiene gran efecto en la detección (Wayne 2000), debido a que el resto de las tareas se realizan a nivel de historia, es decir, que en su mayoría necesitan tener las historias como entrada.

La tarea de detección de la primera historia tiene como objetivo encontrar la primera y solo la primera historia sobre un tópico (Wayne 2000). Esta tarea es un caso especial de la tarea que se enfoca en la detección de una nueva información, es decir, conocer cuándo empieza un nuevo grupo (Allan 2002).

La tarea de detección de grupos se considera una extensión de la detección de la primera historia, requiere que los sistemas agrupen las historias de entrada en grupos de tópicos, y que creen nuevos grupos a medida que sea necesario. Un grupo debe crearse cuando llega la primera historia. La creación de grupos es una tarea no supervisada ya que el sistema no tiene conocimiento por adelantado de la cantidad de grupos esperada. De forma general, la idea de los sistemas de detección es descubrir grupos de historias que tratan el mismo tópico.

Los sistemas de seguimiento de tópicos detectan historias que abordan un tópico previamente conocido, es decir, historias sobre un evento que ya ocurrió. El sistema es provisto con una cantidad pequeña de historias conocidas que tratan del mismo tópico y luego se espera encontrar todas las historias en el flujo de textos que abordan ese tópico. En esta tarea todos los tópicos son seguidos de forma independiente, así, una historia puede pertenecer a múltiples tópicos.

Los sistemas de detección de enlace detectan cuando un par de historias tratan el mismo tópico, o sea, las historias son "enlazadas" por un tópico común. Una ventaja de esta tarea es que hace posible comparar funciones de similitud para determinar cuál distingue mejor pares de tópicos.

En esta tesis se pretende descubrir tópicos en textos de opiniones, sin tener en cuenta el flujo en el que estos textos aparecen, es decir, solo se analizarán colecciones de opiniones sin considerar el momento en que fueron generadas, de ahí el interés en la segmentación y detección de tópicos.

### 1.1.1 Enfoques de tópicos

Un tópico es considerado generalmente como "*un tema importante cuando grandes volúmenes de datos son enviados continuamente al usuario*" (Hamamoto & Pan 2005) o como "*un conjunto coherente de términos relacionados semánticamente que expresan un único argumento*" (Guille et al. 2013). De forma general, los tópicos se presentan frecuentemente en dos enfoques. Primeramente los tópicos globales, los cuales incluyen varios documentos; es decir, un conjunto de documentos que tratan de un tema. Por otra parte, están los tópicos locales, en los que está dividido un documento; es decir, pequeños tópicos que se identifican dentro de los documentos.

En (Lloret 2009) se presenta una clasificación de los tópicos que refina la clasificación anterior de tópicos locales, haciendo referencia a los tópicos de discurso y los tópicos de oración. Si un texto es considerado como un todo, usualmente habla de un solo tópico, pero en un análisis más profundo se pueden identificar varios subtópicos, dando información adicional sobre el tópico principal, y de esta forma se detectan los tópicos de discurso. Considerando la estructura de la oración, se

plantea que cada oración tiene un tópico (la parte de la estructura que está siendo presentada) y un comentario (lo que se afirma sobre el tópico); así se obtienen los tópicos de oración.

Para analizar los tópicos desde el nivel de documento, a nivel global, se deben tener en cuenta los cambios en los subtópicos, debido a que un texto posee generalmente un tópico principal, pero a su vez este contiene varios subtópicos. Es importante detectar estas partes del texto, donde el cambio de tópico ocurre. Esto conlleva a una organización jerárquica de un texto en tópicos y subtópicos, la concatenación de tópicos y el regreso semántico[3]. Los textos son usualmente subdivididos en diferentes párrafos; un párrafo puede ser definido como un segmento coherente de texto enfocado a un único tópico (Lloret 2009). Cada párrafo necesita una oración que indica el tópico relacionado y ésta es usualmente la primera oración del párrafo, además le brinda al lector una idea sobre lo que trata el párrafo. La separación del tópico y el enfoque es relevante no solo para la posible ubicación de una oración en un contexto, sino para su interpretación semántica. Desde este punto de vista el tópico es la información presentada en la oración, mientras el enfoque (comentario) refleja aspectos del tópico que adiciona información nueva o impredecible.

En los datos generados por los usuarios de las redes sociales y para las noticias se pueden identificar dos tipos de tópicos: los tópicos temporales (eventos del mundo real que adquieren popularidad en un momento dado) y los tópicos estables (intereses comunes de los usuarios) (Allan 2002). Algunos ejemplos de tópicos temporales son las emergencias (terremotos), eventos políticos (elecciones), eventos públicos (mundial de fútbol), eventos de negocios (liberación de un nuevo celular), etc. Ejemplos de tópicos estables son la adopción de animales, la discusión de un tema científico o cultural, de forma general son temas debatidos comúnmente por una comunidad estable de usuarios (Yin et al. 2013). En las opiniones es interesante descubrir tópicos temporales y estables, no obstante, en esta tesis sólo abordaremos el descubrimiento de tópicos estables.

En el caso de Twitter, los tópicos temporales son denominados bursty topics, hot topics o emerging topics (Xie et al. 2012). Para que un tema sea emergente debe sustentarse por bases confiables, es decir aparecer en múltiples fuentes y ser novedoso, en otras palabras ser diferente de otros tópicos populares (Prasad et al. 2011). Las técnicas tradicionales de detección de tópicos desarrolladas para analizar colecciones estáticas no son adecuadas para los flujos de mensajes generados por las

[3] Es un patrón recurrente por el cual un tópico es suspendido en un punto y resumido luego en el discurso (Moens & Busser 2001).

redes sociales en línea. En (Guille et al. 2013) se muestra una taxonomía sobre los desafíos actuales de la detección de tópicos en medios sociales.

En el Segmentación de historias (Story segmentation): encontrar regiones homogéneas en el texto tópicamente.

1. Seguimiento (Tracking): encontrar historias adicionales sobre un tópico dado.
2. Detección de la primera historia (First Story Detection; FSD): reconocer el comienzo de un nuevo tópico en el flujo de historias.
3. Detección de grupos (Cluster Detection): detectar y agrupar nuevos tópicos, es decir, agrupar todas las historias tal como llegan, basándose en los tópicos que ellas presentan.
4. Detección de enlaces de historias (Story Link Detection)**:** decidir si dos historias seleccionadas aleatoriamente pertenecen al mismo tópico.

**Anexo *2*** se muestra el resumen de estas clasificaciones. Para la presente tesis son de interés los tópicos estables, locales de discurso y globales, ya que nuestro objetivo consiste en detectar tópicos en colecciones de opiniones, de forma tal que se obtengan grupos de segmentos que abordan el mismo tópico en la opinión o en un conjunto de opiniones.

## 1.2 Formas de representación textual

Para el desarrollo de aplicaciones de detección de tópicos resulta importante seleccionar un modelo de representación textual que tenga en cuenta tanto la información sintáctica como semántica. Los modelos de representación se dividen mayormente en tres grupos, los basados en el modelo espacio vectorial, los basados en grafos y los probabilísticos (Torres & Arco 2015b).

### 1.2.1 Modelo espacio vectorial

El modelo espacio vectorial (Vector Space Model; VSM) es una forma de representación textual que ha sido utilizada en campos de recuperación de información y el procesamiento de textos de forma general para representar documentos textuales a través de vectores de términos (Salton et al. 1975). Una interpretación de este modelo es: "*En VSM cada documento se identifica como un vector de rasgos en un espacio en el cual cada dimensión corresponde a términos indexados distintos (palabras). Un vector documento dado, en cada componente tiene un valor numérico para indicar su importancia*" (Arco 2008). De forma general este modelo se basa en dos conceptos fundamentales: el esquema de pesos y la similitud entre dos vectores de términos (Seijo et al. 2011). El esquema de pesos determina la manera en la que se asignan los pesos a los términos en los documentos, dependiendo de su importancia en el contenido de los mismos. El esquema de

pesos más usado es la frecuencia de aparición de los términos en los documentos (Term Frequency / Inverse Document Frequency; TF-IDF) para expresar el peso relativo del rasgo *w* en el vector asociado a un documento dado y se calcula según la expresión (0.1), donde *idf* (*w*) se calcula según la expresión **¡Error! No se encuentra el origen de la referencia.**:

$$tfidf(w,d) = tf(w,d) * idf(w) \tag{0.1}$$

$$idf(w) = log\,\frac{N}{df(w)} \tag{0.2}$$

Así, *tf* (*w*,*d*) es la frecuencia del término (cantidad de ocurrencias de la palabra *w* en un documento *d*), *idf* (*w*) es la frecuencia inversa de documentos (cantidad de documentos donde aparece la palabra *w* pero de forma inversa, debido a que se le otorga mayor peso a las palabras que ocurren en una menor cantidad de documentos), *df* (*w*) es la frecuencia de documento (cantidad de documentos que contienen la palabra *w*) y *N* representa la cantidad total de documentos en el corpus (Aggarwal & Zhai 2012) (Manning et al. 2008).

El enfoque lingüístico que utiliza este modelo es léxico y se ha utilizado tanto para la segmentación como para la detección de tópicos. La representación VSM se utiliza con frecuencia en la segmentación de textos para representar los documentos con el objetivo de determinar la similitud entre unidades textuales (Abella Raúl & Medina 2013) (Abella & Medina 2014) y en el caso de la detección para representar los documentos antes de agruparlos (Pons-porrata et al. 2002). En sentido general, la principal dificultad de este modelo es que se basa en una comparación estricta de los términos, por lo que la eficacia se ve afectada por palabras distintas que describen el mismo concepto (sinonimia) y por palabras con distintos significados (polisemia). Esta forma de representación tiene la desventaja que opera en el plano estadístico de los documentos, es decir, considera los documentos como bolsas de palabras (Bag of Words; BOW), y sufre de varias limitaciones para capturar estructuras estadísticas de los documentos (Berry & Kogan 2010). En el

Anexo ***3*** se muestra un ejemplo de VSM.

Una propuesta interesante que utiliza el VSM para representar tópicos es el modelo espacio vectorial basado en tópicos (Topic Vector Space Model; TVSM) creado por (Becker & Kuropka 2003) (Wibowo et al. 2011). Otras investigaciones destacan cómo incluir elementos semánticos en una representación VSM. Por ejemplo, Turney describe tres tipos de representaciones de matrices sobre las que se calculan similitudes entre documentos, palabras y relaciones[4] (Turney 2010). Recientemente se han realizado estudios sobre los llamados espacios vectoriales semánticos, los cuales se basan en la idea que el significado de una palabra puede ser aprendido de un entorno lingüístico y poseen dos enfoques, la semántica distribucional y la semántica composicional. El primer enfoque analiza el significado de palabras individuales y el segundo enfoque el significado de frases, oraciones y párrafos. Los métodos existentes de semántica distribucional se pueden basar en vectores de conteo o en la predicción de contextos[5]. Los primeros obtienen patrones estadísticos de las palabras, a partir de los que se descubren diferencias o similitudes entre ellas (Grefenstette et al. 2014) (Clark 2014) y los segundos aprenden representaciones de palabras en vectores de contexto y usan estos vectores para predecir cuán probable es que una palabra ocurra dado su contexto (Bengio et al. 2003) (Mikolov 2013).

### 1.2.2 Análisis semántico latente

El análisis semántico latente (Latent Semantic Analysis; LSA) es un modelo que representa conceptos semánticos presentes en los documentos (Deerwester et al. 1990). Para obtener esta representación, primeramente se realiza la representación VSM del corpus textual. Por tanto, se representa el texto como una matriz término-documento, en el que cada fila representa una palabra y cada columna una oración, párrafo o documento. El valor en la intersección de fila y columna es la frecuencia con la cual la palabra aparece en el documento. Este valor se calcula a través de una función de peso que exprese la importancia del término en el documento. Para una colección real, esta matriz tiene más de 100000 filas y columnas, por ello se aplica la técnica de factorización de matrices conocida por descomposición de valores singulares (Singular Value Descomposition; SVD), donde la matriz representada anteriormente puede ser descompuesta en el producto de tres matrices como se expresa en la ecuación (0.3):

[4] Matriz término-documento, matriz palabra-contexto y matriz par-patrón.

[5] En esta tesis son de interés solo los espacios vectoriales semánticos distribucionales basados en vectores de conteo.

$$X = T_0 S_0 D_0 \tag{0.3}$$

La clave de LSA es usar SVD para construir una matriz (concepto x documento) donde el número de conceptos sea varios órdenes de magnitud menor al número de términos en la colección (típicamente para el número de conceptos se escoge un valor bastante inferior a 1000). En la ecuación (0.3) *X* representa una matriz *M x N* (*M* es el tamaño del vocabulario y *N* es el tamaño de la colección), $T_0$ es una matriz *M x K* (*K* es la cantidad de conceptos), $S_0$ es una matriz *K x K* (matriz diagonal de valores singulares), $D_0$ es una matriz *K x N* (la matriz que va a asociar cada documento con sus conceptos). *T* y *D* son matrices cuadradas con columnas ortonormales[6] y se denominan las matrices izquierda y derecha de vectores singulares; respectivamente. *S* es la matriz diagonal de valores singulares, sus elementos son todos positivos. Los valores más altos de $S_0$ se corresponden a las correlaciones dominantes en la colección. Si los valores singulares de $S_0$ están ordenados, se pueden tomar los *k* valores más altos (significa quedarse con los *k* conceptos más prominentes en la colección), fijar el resto a 0 y obtener una aproximación a *X*, que es la matriz de rango *k* más cercana a *X* según mínimos cuadrados (Seijo et al. 2011). Al descartar las dimensiones (filas y columnas) con valor 0 se obtiene la matriz aproximada $\hat{X}$, donde *S* es *k x k* y *T* (*M x k*) y *D'* (*k x N*) surgen a partir de eliminar en $T_0$ y $D_0$ las columnas asociadas a las dimensiones descartadas en $S_0$. Así se obtiene el modelo reducido expresado en (0.4):

$$\mathrm{X} \sim \hat{X} = \mathrm{TSD}' \tag{0.4}$$

La elección de *k* es importante pues determina la cantidad de conceptos resultantes en el espacio semántico. Este valor debe ser suficientemente alto para ajustarse apropiadamente a la estructura conceptual que se tiene en la colección pero sin llegar a ser tan alto como para sobreajustar la colección con detalles insignificantes (Seijo et al. 2011). En el Anexo 4 se muestra la descomposición de valores singulares reducidos gráficamente. Este modelo representa los vectores de documentos en un espacio dimensional asociado a los conceptos presentes en la colección, y por tanto, se considera una forma de representación textual, no obstante, otros autores clasifican este modelo como un método de reducción de dimensionalidad.

LSA se sobrepone a los asuntos de sinonimia y polisemia lo cual es útil para la detección de tópicos (Aggarwal & Zhai 2012) (Abella & Medina 2014). También presenta como ventaja que reduce la

[6] Es ortonormal si es a la vez un conjunto ortogonal y la norma de cada uno de sus vectores es igual a 1.

dimensionalidad de documentos al proyectar los vectores BOW en un espacio semántico construido a partir de la matriz término-documento de SVD. Este modelo ha sido usado recientemente para detectar tópicos populares en contextos sociales como en los textos de microblogs (Yan 2013). LSA tiene como limitación que dada la naturaleza de alta dimensionalidad de los datos textuales, el cálculo de SVD puede ser costoso y las dimensiones resultantes pueden ser difíciles de interpretar debido a que cada dimensión es una combinación lineal de un conjunto de palabras a partir del espacio original (Berry & Kogan 2010).

### 1.2.3 Grafos

Los documentos se pueden representar también utilizando grafos, los cuales pueden ser clasificados de acuerdo a las representaciones de sus nodos y sus aristas. Los nodos de los grafos representan unidades textuales que pueden ser términos, palabras, oraciones, párrafos, documentos o conceptos, estos últimos son considerados componentes semánticos (Chang & Kim 2014). Las aristas representan las relaciones entre unidades textuales y pueden ser clasificadas en dirigidas o no dirigidas para indicar el orden de los nodos o las interacciones entre ellos. También pueden ser pesadas y no pesadas considerando el peso en una arista como la frecuencia con que aparecen en un grafo de coocurrencia de palabras relacionadas, así como la distancia de dos palabras en un texto. Las aristas igualmente pueden ser etiquetadas y no etiquetadas; una etiqueta indica relaciones entre palabras, por ejemplo son usadas para representar roles gramaticales de las palabras: un nodo sujeto y un nodo objeto están enlazados por una arista etiquetada con "verbo". Las relaciones entre vértices puede representarse por la coocurrencia de palabras juntas en una oración, párrafo, sección o documento, así como por relaciones semánticas por ejemplo, sinónimos, antónimos, etc. (Sonawane & Kulkarni 2014). En el Anexo 5 se muestran algunos tipos de grafos.

La representación de textos usando grafos ha sido utilizada en la detección de tópicos; por ejemplo, aplicando algoritmos de agrupamiento jerárquico basados en grafos de conceptos (Huang et al. 2013). Los vértices del grafo son conceptos, los cuales corresponden a subtópicos de un tópico, y se relacionan a través de la coocurrencia de palabras claves (Huang et al. 2013). Otras técnicas basadas en grafos han sido propuestas para clasificar mensajes de Twitter (Cordobés et al. 2014) y para capturar la estructura de conversaciones en blogs (Joty et al. 2013). Los árboles son casos especiales de grafos, algunos métodos de representación textual usan árboles de análisis gramatical para representar relaciones sintácticas de las oraciones (Massung & Hockenmaier 2013). La

representación textual utilizando árboles sintácticos ha sido utilizada en la segmentación por tópicos (Nallapati & Allan n.d.). También existen trabajos que utilizan modelos híbridos combinando grafos y VSM para representar el texto (Ngoc & Do 2012) (Valle & Ozt 2011).

### 1.2.4 Análisis semántico latente probabilístico

El análisis semántico latente probabilístico (Probabilistic Latent Semantic Analysis; PLSA) se propuso como una versión probabilística de LSA (Hofmann 1999). PLSA es un modelo generativo[7] probabilístico desarrollado para el análisis estadístico del texto. En el análisis de textos este modelo se utiliza para descubrir la semántica de tópicos ocultos en los documentos usando una representación BOW (Ren & Han 2014). PLSA se basa en un modelo de aspectos; en el contexto de la detección de tópicos se puede describir de la siguiente forma:

Dada una colección de documentos $D = \{d_1, \dots, d_N\}$ con palabras de una colección de términos $w \in W = \{w_1, \dots, w_M\}$ la cual es nombrada también como vocabulario; los datos para la colección de documentos se representan en una matriz de coocurrencia $V$ x $N$ de cantidades $N_{ij}=n(w_i,d_j)$, donde la cantidad $n(w_i,d_j)$ indica cuán a menudo el término $w_i$ ocurre en el documento $d_j$. De esta forma se considera un modelo variable latente al asociar a una variable de tópico oculta $z \in Z = \{z_1, \dots, z_z\}$ con cada observación, esta última representa la ocurrencia de una palabra en un documento ($w_i$, $d_j$). En el modelo generativo primeramente se selecciona un documento $d_j$ de acuerdo a la distribución de probabilidad $P(d_i)$, se escoge una variable de tópico oculta $z_k$ de acuerdo a la distribución de probabilidad $P(z_k|d_i)$ y se genera una palabra $w_i$ de acuerdo a la distribución probabilística $P(w_j|z_k)$. $P(d_i)$ se usa para nombrar la probabilidad que una ocurrencia de palabras será observada en un documento particular $d_i$. $P(w_j|z_k)$ es la probabilidad condicional de una palabra específica $w_i$ condicionada en el tópico oculto $z_k$. $P(z_k|d_i)$ es la probabilidad condicional de un tópico oculto $z_k$ en el documento $d_j$ (Ren & Han 2014).

El modelo de aspectos para PLSA se presentó en (Hofmann 2001). Los parámetros del modelo se estiman utilizando el algoritmo iterativo de Maximización – Expectación (Expectation – Maximization; EM) (Abella & Medina 2014). PLSA provee una buena base para el análisis de textos, pero tiene limitantes (Aggarwal & Zhai 2012). Primero, contiene una gran cantidad de

[7] Un modelo generativo para documentos está basado en reglas simples de muestreo probabilístico que describen cómo las palabras en los documentos pueden estar generadas en las bases de variables latentes (aleatorias). Cuando se adapta un modelo generativo, el objetivo es encontrar el mejor conjunto de variables latentes que pueden explicar los datos observados (es decir, palabras observadas en los documentos), asumiendo que el modelo genera datos realmente (Steyvers & Griffiths 2004).

parámetros en el modelo que crece linealmente con el tamaño de la colección y no es un modelo generativo completo, porque no existe una forma exacta de modelar la distribución tópica de un documento que no esté incluido en el conjunto de datos. Sin embargo, presenta ventajas por las cuales se ha usado en la detección de tópicos, entre ellas está que permite tratar con palabras polisémicas y distinguir explícitamente entre significados y tipos diferentes de usos de palabras. Además, hace reducción de dimensionalidad, ya que la cantidad de variables latentes (tópicos) es menor que la cantidad de términos.

### 1.2.5 Asignación latente de Dirichlet

El modelo de asignación latente de Dirichlet (Latent Dirichlet Allocation; LDA) es un modelo probabilístico generativo para colecciones de datos discretos (Blei et al. 2003). LDA es un modelo bayesiano jerárquico de tres niveles (documento, palabra y tópico), el cual considera a un tópico como "*una distribución sobre un vocabulario fijo*". El modelo toma previamente una cantidad de tópicos predefinida para toda la colección y se definen las palabras que pertenecen a esos tópicos. El procesamiento del modelo consiste básicamente en identificar en qué medida esos tópicos se presentan en los documentos; primero se escoge una distribución sobre los tópicos, es decir, el conjunto de tópicos predefinidos con sus palabras más probables. Luego, para cada palabra del documento se escoge una asignación de tópicos y se selecciona la palabra para el tópico correspondiente. En el Anexo 6 se muestra un ejemplo de estas distribuciones por tópicos. El histograma representa la distribución de Dirichlet y la salida son las palabras más frecuentes por tópicos predefinidos.

Una característica importante de LDA es que todos los documentos en la colección comparten el mismo conjunto de tópicos, pero cada documento exhibe los tópicos en diferentes proporciones. Los documentos son observados, mientras que la estructura de tópicos (los tópicos, las distribuciones de tópicos por documentos y las asignaciones de tópicos por palabra por documento) está oculta. El problema computacional consiste en usar los documentos observados para inferir la estructura de tópicos oculta[8]. Para cada documento en la colección LDA genera palabras en un proceso de dos etapas: (1) escoger la distribución sobre los tópicos aleatoriamente (distribución de Dirichlet) y (2) para cada palabra en el documento escoger un tópico aleatoriamente de la

[8] Algunas técnicas de inferencia propuestas son la inferencia de variación del campo medio (mean field variational inference), inferencia de variación colapsada, maximización de la expectativa (expectation-maximization) y el muestreo de Gibbs.

distribución sobre tópicos en el paso 1 y escoger una palabra aleatoriamente de la distribución correspondiente sobre el vocabulario.

En la modelación probabilística generativa, se tratan los datos que surgen de un proceso generativo e incluyen variables ocultas. Este proceso generativo define una distribución de probabilidad unificada sobre las variables aleatorias ocultas y las observadas. LDA es considerado un modelo generativo de tópicos no supervisado, realiza reducción de dimensionalidad y el enfoque lingüístico que utiliza es semántico (Aggarwal & Zhai 2012).

Una ventaja de LDA es que los parámetros de tópicos y la distribución de datos generados pueden adaptarse a otros tipos de observaciones con solo pequeños cambios a los algoritmos de inferencia. LDA ayuda también con la polisemia; por ejemplo, si se considera un término con dos significados distintos, el modelo ubica igual probabilidad para los tópicos correspondientes a los significados. Sin embargo, si las otras palabras en el contexto ubican una probabilidad mayor en el primer significado y una probabilidad menor en el segundo significado, entonces LDA será capaz de usar el contexto para desambiguar el tópico, es decir selecciona el significado de mayor probabilidad (Aggarwal & Zhai 2012). Una limitación mencionada a menudo de LDA es la incapacidad de escoger la cantidad óptima de tópicos (Carenini et al. 2011); aunque se han propuesto métodos cuantitativos para medir el significado semántico de los tópicos inferidos (Chang et al. 2009). El modelo LDA ha sido juzgado por su incapacidad de capturar correlaciones entre las palabras de tópicos, es decir, el orden de las palabras y la información estructural inherente del texto; por ejemplo, un documento sobre el "medio ambiente" es más probable que sea de "salud" que de "religión" (Shafiei & Milios 2008). Así se han propuesto varias extensiones de LDA, como el modelo de tópicos correlacionados, el modelo de tópicos dinámicos y el modelo LDA jerárquico para superar esa dificultad (Blei & Lafferty 2008) (Aggarwal & Zhai 2012).

Los modelos PLSA y LDA pueden utilizarse como una forma de representación textual, considerando que los tópicos identificados serán los términos a utilizar para describir los documentos, e incluso pueden considerarse como formas de reducción de la dimensionalidad, ya que permiten caracterizar los documentos por el subconjunto de términos (tópicos) identificados. No obstante, es posible considerar estos modelos como métodos propiamente para detectar tópicos, considerando que cada tópico se corresponde con cada término identificado, por lo que se consideran también ellos mismos como una forma de detección de tópicos.

## 1.3 Segmentación por tópicos

La segmentación por tópicos es considerada por algunos autores la tarea inicial de TDT, debido a que los segmentos obtenidos en esta etapa constituyen la entrada para el resto de las tareas (Allan 2002). Segmentar es "*la tarea de dividir un documento en unidades sintácticas (párrafos, oraciones, etc.) o en bloques semánticos, usualmente en tópicos*" (Abella & Medina 2014), "*particionar un documento en secciones, donde cada sección se enfoca en un subtópico diferente del documento*" (Turney 2010), o encontrar segmentos con alta cohesión léxica. Aunque algunos algoritmos para la detección de tópicos no realizan la segmentación y que la segmentación en si no garantiza la detección de tópicos, esta tarea resulta de gran importancia en el desarrollo de algoritmos de TDT. A continuación se expondrán los principales métodos de segmentación y sus clasificaciones.

### 1.3.1 Clasificaciones de los métodos de segmentación

Los métodos de segmentación por tópicos pueden clasificarse en planos o jerárquicos (Carenini et al. 2011). Para un tópico plano el texto se representa como una secuencia de segmentos de tópicos sin descomposición adicional, mientras que en los tópicos jerárquicos los segmentos pueden ser divididos en subtópicos. Esta clasificación también se conoce por segmentación lineal o segmentación jerárquica (Abella & Medina 2014). Los tópicos jerárquicos facilitan la navegación, recuperación y resumen de documentos, debido a la amplia y detallada información que proveen. La dificultad de la representación de tópicos varía en la existencia de diferentes dominios de textos. En monólogos editados, tales como libros y artículos, la representación de tópicos se considera relativamente simple, debido a que los materiales en diferentes tópicos vienen organizados típicamente por capítulos, secciones y párrafos, de esta forma reflejan la estructura tópica. En contraste, en documentos menos formales, incluyendo textos de conversaciones, donde los materiales no están editados ni organizados explícitamente, la modelación de tópicos se convierte en algo más complejo. Por ejemplo, en una conversación el inicio de un tópico y el final del anterior a menudo se sobreponen y un tópico puede ser introducido múltiples veces antes de convertirse en el foco de la discusión. Para guiar estos desafíos, se han realizado trabajos sobre el descubrimiento de tópicos en conversaciones (Carenini et al. 2011).

Los métodos de segmentación se pueden clasificar además en dos grupos de acuerdo al mecanismo que siguen para realizar la segmentación: los métodos golosos, los cuales de una sola pasada identifican los límites de los segmentos, y están los métodos que utilizan programación dinámica,

los cuales hacen varias segmentaciones con el objetivo de encontrar la segmentación óptima. Aunque existen varios métodos de segmentación la eficacia de estos no es del todo aceptable, ya que usualmente generan segmentos incompletos o espurios. Además, pocos consideran la mezcla de tópicos en los segmentos que descubren, o las diferentes relaciones que estos presentan. Las debilidades que presentan los métodos de segmentación, tanto los golosos como los de optimización, es que son dependientes de umbrales. La definición de un umbral es realmente difícil; tanto para decidir si dos unidades textuales están cohesionadas, la cantidad de segmentos que tiene un documento o el tamaño de un tópico[9] (Abella & Medina 2014).

Estos métodos pueden tener enfoque supervisado o no supervisado de acuerdo a si se tienen previamente un conjunto de documentos con los límites de los segmentos identificados o no, respectivamente (Purver 2011).

Los algoritmos desarrollados para la segmentación de tópicos se apoyan en dos ideas fundamentales: en los cambios de contenido y la detección de límites. Para la primera idea se ha afirmado que el uso repetitivo de los mismos objetos o conceptos indica la presencia de un tópico, mientras que si se introduce un nuevo vocabulario hay un cambio de tópico. Así se considera que las regiones con cambios pequeños deben corresponder a segmentos de tópicos, y los grandes cambios a los límites de segmentos. Para seguir esta idea Purver menciona tres enfoques principales de métodos, los cuales están guiados por la idea que los tópicos están asociados con el contenido, y por lo tanto, caracterizados por un conjunto de palabras, conceptos y referencias particulares (Purver 2011). Los enfoques son:

Discriminativo: utiliza una medida de similitud para medir la diferencia entre las secciones vecinas del discurso directamente, y descubrir los límites donde se indican grandes diferencias; principalmente se concentran en los cambios que ocurren en distribuciones léxicas.

Agrupamiento: consiste en agrupar oraciones vecinas muy similares hasta que se construye un conjunto de grupos de tópicos que cubre todo el discurso.

Generativo o probabilístico: estima modelos de lenguajes para tópicos, y descubre límites al encontrar la secuencia más probable de estados de tópicos para generar el discurso observado.

[9] Existen diferentes tipos de umbrales, tales como: umbral de cohesión, umbral de cantidades de segmentos y umbral de tamaño de tópico (Abella & Medina 2014).

Conocer los rasgos distintivos de los límites de tópicos permite identificar cuando un tópico inicia y cuando termina y de esta forma permite realizar la segmentación del texto en tópicos. Para ello se utilizan también los marcadores de discurso (palabras de indicios y frases) que directamente proveen pistas sobre la estructura del discurso. Los rasgos más indicativos del cambio de tópico dependerán de la naturaleza del dato: dominio, medio de difusión y número de participantes. La tarea de segmentación es muy usada también para el discurso hablado (Purver 2011).

### 1.3.2 Métodos para la segmentación por tópicos

A continuación se describen algunos métodos de segmentación por tópicos, los que se agrupan fundamentalmente en dos enfoques, los basados en cohesión léxica que analizan los cambios de contenido en el texto y los que analizan técnicas para la detección de límites (Purver 2011).

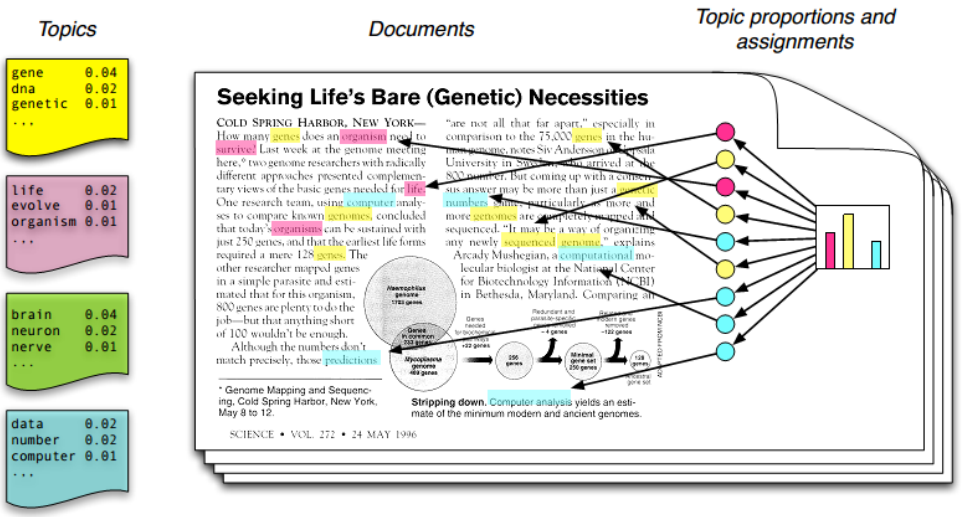
#### 1.3.2.1 Basados en cohesión léxica

La cohesión léxica considera que un discurso es coherente si exhibe ciertos tipos de relaciones con las entidades sobre las que trata, presentándolas y siguiéndolas de forma centrada; esta coherencia es llamada coherencia basada en entidades (Jurafsky & Martin 2007). Un factor clave en la segmentación de tópicos es que las oraciones dentro de un segmento están más conectadas entre ellas que con oraciones de otros segmentos. La fuerza de conexión entre dos oraciones es conocida en el campo de la lingüística como cohesión y se determina por cuán cerca están las palabras en dos oraciones, lo cual es llamado cohesión léxica, y además por el uso de otros mecanismos lingüísticos, incluyendo sobre todo a los pronombres (Carenini et al. 2011). La cohesión fue definida también por Jurafsky y Martin como "*el uso de ciertos mecanismos lingüísticos para enlazar o juntar unidades textuales"*, y a la cohesión léxica como *"la cohesión indicada por relaciones entre palabras en las dos unidades, tal como el uso de una palabra idéntica, un sinónimo o un hiperónimo*". Los términos cohesión y coherencia tienen significados distintos, debido a que la cohesión se refiere a la forma en que unidades textuales están juntas o enlazadas y la coherencia se refiere a la relación de significado entre dos unidades. Una relación de coherencia explica como el significado de diferentes unidades textuales pueden combinarse de forma conjunta para construir un significado de discurso para una unidad más grande (Jurafsky & Martin 2007). La idea de los métodos basados en cohesión para la segmentación es que las oraciones o párrafos en un subtópico son cohesivos entre ellos, pero no con párrafos en un subtópico vecino. Los métodos de segmentación basados en cohesión léxica se encuentran clasificados en tres enfoques: detectar los cambios de contenido por similitudes léxicas, por similitudes basadas en agrupamiento

y por inferencia probabilística (Purver 2011). A continuación se expondrán algunos métodos por cada enfoque.

- Métodos que se basan en cambios de contenido por similitudes léxicas

La idea básica de este enfoque es que los cambios de tópicos tienden a ser marcados por un cambio en el vocabulario utilizado, el cual puede ser detectado al buscar el mínimo en alguna medida de cohesión léxica (Purver 2011). Uno de los métodos de segmentación basado en cohesión léxica más citados por la comunidad de investigadores de este campo es TextTiling (Hearst 1997); el cual aplica técnicas para subdividir textos en unidades multi-párrafos que representan pasajes o subtópicos, es decir, los segmentos de textos. Para ello identifica los cambios mayores de subtópicos a través de patrones de coocurrencia léxica y de distribución. El objetivo de este método es encontrar tópicos principales distribuidos globalmente y subtópicos que ocurren de forma local. Luego de hacer un pre-procesamiento del texto, TextTiling realiza una segmentación lineal para lo cual determina una puntuación de cada oración. El algoritmo propone emplear uno de tres métodos para determinar la puntuación, como se muestra en el




Anexo *7*.

El primer método consiste en una comparación de bloques adyacentes de texto para ver cuán similar son de acuerdo a cuántas palabras los bloques tienen en común; si una puntuación léxica baja es precedida y seguida por puntuaciones léxicas altas, se asume que hay un cambio en el vocabulario; los bloques actúan como ventanas en movimiento (deslizantes) sobre el texto. El algoritmo propuesto calcula las puntuaciones como el producto interior de dos vectores, donde un

vector contiene la cantidad de veces que cada objeto léxico ocurre en su bloque correspondiente. El segundo método consiste en guardar las posiciones en la que los términos son introducidos por primera vez. Se utiliza una ventana deslizante, la cantidad de términos nuevos que ocurren en cualquiera de los lados del punto medio o en el espacio entre las oraciones de interés se adicionan y grafican contra la cantidad de espacios entre oraciones. De esta forma se asigna una puntuación a cada espacio de la secuencia de tokens basada en la cantidad de palabras nuevas que son vistas en el intervalo definido por la ventana. Cada bloque se representa usando VSM y la similitud entre ellos se calcula con la medida coseno, así se asigna una puntuación de similitud entre estos bloques. En el tercer método se considera activa una cadena léxica para un término a través del espacio entre oraciones, si las instancias del término ocurren dentro de un umbral de distancia de uno a otro. La puntuación para el espacio entre oraciones es la cantidad de cadenas activas que se encuentran en ese espacio.

Luego de obtener la puntuación por oraciones, TextTiling detecta los límites de los subtópicos para todos los métodos de puntuación léxico y se asigna una puntuación de profundidad del valle[10], siempre que exista, para cada espacio entre la secuencia de tokens. La puntuación de profundidad corresponde a cuán fuertemente cambian las señales de indicio para un subtópico en ambos lados del espacio y está basado en la distancia de los picos en ambos lados del valle. De esta forma, TextTiling es considerado un método para descubrir la estructura tópica de un documento usando la repetición de términos (Hearst 1997).

Una propuesta similar a TextTiling fue la realizada por Heinonen, al proponer un método de optimización para la segmentación lineal multi-párrafos basado en un algoritmo de programación dinámica para determinar los límites de tópicos. El algoritmo considera todas las segmentaciones posibles y determina la de mínimo costo (Heinonen 1998).

LCseg es un algoritmo de segmentación de tópico orientado a conversaciones, específicamente transcripciones de reuniones (meeting transcripts); usa la repetición de términos para detectar los límites de los tópicos como mecanismo de cohesión léxica. Identifica y pesa repeticiones de

[10] Depresión en un gráfico que representa las puntuaciones obtenidas aplicando la estrategia de ventana deslizante. Se asume que los límites ocurren en los valles más grandes en el gráfico que resulta de representar unidades de oraciones contra las puntuaciones. Los valles más profundos reciben mayores puntuaciones que los menos profundos.

términos usando cadenas léxicas y detecta los límites de tópicos a partir de la similitud coseno entre las cadenas léxicas que se superponen con dos ventanas (Galley & Mckeown 2003).

SeLeCT (Segmentation using Lexical Chaining on Text) es un sistema que usa cadenas concatenadas de textos para devolver segmentos de reportes de noticias. El sistema consiste fundamentalmente en un analizador léxico para el procesamiento del texto (LexNews) el cual representa los documentos en VSM y usa un detector de límites basado en cadenas (Stokes 2004) (Hernández & Pagola 2009).

TextLec es un método desarrollado para segmentar automáticamente textos por tópicos sobre colecciones de documentos científico-técnicos. Este método utiliza la cohesión léxica como señal de cambio de tópico, representa los párrafos como unidades textuales basándose en VSM y evalúa la similitud entre ellas con la medida del coseno. TextLec utiliza, además, la teoría computacional de Skorochod'ko sobre la estructura lineal del discurso y una ventana de párrafos inferiores, por debajo a cada párrafo, con vistas a localizar el párrafo cohesionado más lejano y evitar la interrupción de los tópicos. TextLec considera que dos párrafos mantienen una cohesión léxica significativa o suficiente para pertenecer al mismo segmento si, después del cálculo de la misma, ésta es mayor que un umbral determinado. Realiza una etapa de tratamiento de segmentos cortos o de pocos párrafos (espurios) a partir del tamaño de ventana especificado por el usuario (Hernández & Pagola 2009) (Abella & Medina 2014). Otras investigaciones no supervisadas que utilizan el enfoque de cambios de contenido por similitudes léxicas se publicaron en (Kern et al. 2009) (Ferret 2002).

- Cambios de contenido por similitudes basadas en agrupamiento

Un punto de vista para detectar cambios de contenido es que en vez de observar áreas de baja cohesión (los límites) se pueden observar áreas de alta cohesión (los segmentos de tópicos). Para ello se han empleado algoritmos de agrupamiento jerárquico; aquellos divisivos han sido más efectivo para la tarea de segmentación (Purver 2011). En 1994, Reynar usó la técnica de representación gráfica de puntos (dot-plotting) para segmentar un texto por tópicos. La aplicación de esta técnica puede ser realizada de forma manual al examinar un gráfico o automáticamente usando un algoritmo de optimización. Se basa en la repetición de unidades léxicas debido a que el gráfico usado para descubrir los límites de tópicos se crea al enumerar las unidades léxicas de un

documento y se representan los puntos que corresponden a repeticiones de palabras (Reynar 1994) (Reynar 1998).

El algoritmo C99 descubre las ubicaciones de los límites con un algoritmo de agrupamiento divisivo (Choi 2000). Luego de pre-procesar el texto se construye para cada oración un diccionario de frecuencia de palabras con su raíz; el cual está representado por un vector de cantidades de frecuencias. Los valores de similitud coseno entre las oraciones se representan en una matriz, a la cual se le aplica un esquema de ordenación de imágenes y se obtiene una matriz de rango, donde el rango es la cantidad de elementos vecinos con menor valor de similitud. Finalmente, se aplica un algoritmo de agrupamiento divisivo para determinar los límites de los tópicos. Una investigación posterior sobre C99 conocida por CWM utiliza como forma de representación a LSA (Choi et al. 2001). Sus autores expresan que la lematización no siempre mejora la exactitud de la segmentación y el orden (ranking) es de gran importancia para la medida del coseno pero no para LSA; declaran que LSA posee una medida de similitud más exacta que la del coseno. El método trabaja satisfactoriamente en segmentos de tópicos largos, mayor a seis oraciones (Choi 2000) (Choi et al. 2001).

ClustSeg es otro método de segmentación lineal por tópicos, que tiene como objetivo identificar límites de tópicos en documentos de textos utilizando la repetición de términos como mecanismo de cohesión léxica. Considera como unidad textual a los párrafos, y asume que los tópicos están definidos por el conjunto de todas las oraciones o términos de un mismo párrafo. La representación que utiliza es VSM y la cohesión entre estos se calcula con la medida del coseno. La identificación de los límites de tópicos se obtiene al aplicar un algoritmo de agrupamiento y utilizar la estrategia de ventanas deslizantes. ClustSeg se basa en que los cambios de vocabulario en un documento coinciden con los cambios de subtópicos, por lo que los párrafos que presentan una cohesión léxica significativa entre sí, respecto a los términos léxicos que usan, tratan sobre los mismos tópicos. Sin embargo, a diferencia de métodos anteriores no consideran la distancia que exista entre párrafos, debido a que en un documento el autor puede referirse a un tópico anterior, es decir, que un párrafo puede estar relacionado con más de un tópico. Debido a esto se utiliza un algoritmo de agrupamiento con solapamiento para identificar los límites de tópicos (Abella & Medina 2010). NClustSeg es un método de segmentación basado en ClustSeg, el cual adicionalmente propone una estrategia para el cálculo automático del umbral que utiliza el algoritmo de agrupamiento para formar los conjuntos de párrafos que están en un mismo grupo. El método NClustSeg obtiene

mejor eficacia que el método ClustSeg en textos largos; sin embargo su eficacia es mala en textos cortos (Abella & Medina 2014).

- Cambios de contenido por inferencia probabilística

Se han desarrollado métodos para segmentar textos que utilizan mecanismos de inferencia probabilística para identificar los límites de los segmentos. Estos métodos se basan en que cuando el tópico cambia, el vocabulario utilizado cambia; así si se infiere la secuencia de tópicos más probable de las palabras observadas y se pueden obtener posiciones de los límites entre ellos. Se debe destacar que este enfoque no requiere una medida de similitud entre expresiones o ventanas directamente, el hecho que las expresiones vecinas dentro del mismo segmento de tópico son similares entre ellas está implícito en el hecho que han sido generadas a partir del mismo tópico (Purver 2011). TextSeg es un método estadístico de optimización que encuentra la segmentación de probabilidad máxima de un texto (Kern et al. 2009). El método no necesita datos de entrenamiento para estimar probabilidades, selecciona la segmentación óptima en términos de probabilidad definida por un modelo estadístico. La tarea de segmentación se modela como el problema de encontrar la segmentación de costo mínimo o la segmentación de probabilidad máxima. Recientemente Simon y colaboradores extendieron la propuesta basada en grafo de TextSeg para utilizar la combinación de cohesión y la discontinuidad léxica. Para la segmentación de tópicos existen dos estrategias que siguen el principio general de cohesión léxica, por una parte una medida de cohesión léxica puede ser usada para determinar segmentos coherentes, y por otra, los cambios en el uso del vocabulario pueden ser buscados para directamente identificar los límites de segmentos al medir la discontinuidad léxica (Simon et al. 2013).

Misra y colaboradores dieron un enfoque a la segmentación de tópicos desde una perspectiva probabilística utilizando el método de LDA que devuelve los límites de los segmentos y la distribución de tópicos asociada con cada segmento. Debido al alto costo computacional de LDA usan un algoritmo de programación dinámica basado en los principios del método TextSeg pero considerando los segmentos con puntuaciones asignadas por LDA (Misra et al. 2011).

Mimi Lu y colaboradores propusieron un método de segmentación lineal de historias para noticias de difusión aplicando el modelo PLSA. En esta propuesta se utilizan los fonemas n-gram como unidad básica de términos para medir la cohesión léxica entre las palabras del vocabulario de un discurso. Los términos individuales no proveen evidencia confiable sobre sus conceptos y modelos

como PLSA permiten realizar una correspondencia conceptual, de esta forma las técnicas empleadas intentan explorar la estructura semántica latente subyacente en el dato, que es oscurecida parcialmente por la aleatoriedad de las elecciones de palabras (Lu et al. 2011).

TopicTiling es un método de segmentación de tópicos basado en el algoritmo de TextTiling. Consiste básicamente en que en vez de usar palabras directamente como rasgos para caracterizar unidades textuales, usa los identificadores (ID) de los tópicos asignados por el método de inferencia bayesiana de LDA, esto incrementa la dispersión de vectores palabras, ya que el espacio de la palabra es reducido a un espacio de tópicos de baja dimensión. Por ello los documentos que serán segmentados deben ser anotados con ID de tópicos (Riedl & Biemann 2012a) (Riedl & Biemann 2012b). TopicTiling tiene la ventaja que realiza la segmentación en tiempo lineal $O(n)$, $n$ representa la cantidad de oraciones, por lo que es computacionalmente menos costoso que otros métodos de segmentación basados en LDA. En (Du & Johnson 2013) y (Shafiei & Milios 2008) se proponen otros métodos de segmentación sobre modelos bayesiano jerárquicos. Es importante conservar el orden de las palabras para la segmentación de texto por tópicos (Jameel & Lam 2013). Un ejemplo es el modelo generativo no supervisado NTSeg, el cual mantiene la estructura de segmentos de un documento tales como párrafos y oraciones y conserva el orden de las palabras en el documento (Jameel & Lam 2013).

#### 1.3.2.2 Basados en detección de límites

Otro enfoque para segmentar consiste en buscar los rasgos característicos de los límites, por ejemplo, las frases o palabras de indicio (cue words) o marcadores de discurso, que se usan para señalar un cambio de tópico (Purver 2011). Algunos autores destacan que estas palabras son usadas a menudo para la segmentación supervisada; definen un marcador de discurso como: "*una palabra o frase que funciona para señalar una estructura de discurso"* (Jurafsky & Martin 2007). Los marcadores de discurso tienden a ser muy específicos del dominio. Ejemplo de marcadores para la segmentación de noticias pueden ser "buenas noches" al inicio de la noticia, "se nos une ahora" para el inicio de algunos segmentos y "hasta mañana" puede indicar el fin de un segmento. Otro ejemplo son los pronombres que usualmente indican que una oración está relacionada a otra anterior (Misra et al. 2011).

Es posible escribir reglas o expresiones regulares para identificar los marcadores de discurso para un dominio dado. Tales reglas a menudo se refieren a entidades nombradas. Existen métodos automáticos para encontrar marcadores de discurso para la segmentación. Primero codifican todas

las palabras posibles o frases como características para un clasificador, y luego hacen una selección de rasgos en el conjunto de entrenamiento para encontrar solo las palabras que son los mejores indicadores de un límite o frontera (Jurafsky & Martin 2007). Las palabras que ofrecen algún indicio para la segmentación han sido muy utilizadas en el análisis de texto hablado (Galley & Mckeown 2003) (Purver 2011).

Una propuesta de segmentación de textos basada en la detección de límites fue realizada por Beeferman, siguiendo un enfoque estadístico para particionar el texto en segmentos coherentes. La idea de su propuesta es construir un modelo que asigna una probabilidad al final de cada oración, la probabilidad de que exista un límite entre esa oración y la siguiente (Beeferman et al. 1999). Reynar en 1999 concluyó que un algoritmo para la segmentación de tópicos que utilice frases de indicio, frecuencia de palabras, la repetición de entidades nombradas (personas, lugares, compañías), el uso de pronombres, es decir, un enfoque de cohesión léxica y de detección de tópicos juntos obtiene mejores resultados que un algoritmo que solo utiliza la frecuencia de palabras (Reynar 1999).

BayesSeg es un enfoque bayesiano para la segmentación por tópicos de manera no supervisada (Eisenstein & Barzilay 2008). Este método presenta el enfoque de la cohesión léxica al modelar las palabras en cada segmento de tópico como una extracción de un modelo de lenguaje multinomial asociado con el segmento; y además provee una forma integral de incorporar características adicionales tales como frases de indicio. Por lo que BayesSeg realiza una combinación de dos enfoques de métodos de segmentación: cohesión léxica y detección de límites.

### 1.3.3 Consolidación de las etapas para la segmentación por tópicos

A partir del estudio de los métodos de segmentación encontrados en la literatura, se realizó, como parte de esta investigación, una generalización del proceso de segmentación en cuatro etapas principales (Torres & Arco 2015c), las cuales están representadas en un diagrama que se muestra en el Anexo 8 a). Y son:

Pre-procesar el texto: etapa en la cual se aplican técnicas como la tokenización, eliminación de palabras vacías, la lematización, eliminación de signos de puntuación y espacios, reducción de mayúsculas y otros, con el fin de obtener los términos más representativos del texto. Tiene como entrada el corpus textual y como salida los tokens[11].

[11] Representan los términos procesados.

Representar el documento: en esta etapa se utiliza un modelo computacional que representará las unidades textuales por documento. Es útil que el modelo a utilizar permita considerar información sintáctica y semántica, así como que pueda identificar clasificaciones léxicas en el texto como la ambigüedad, la sinonimia y la polisemia. Se debe tener en cuenta el tipo de análisis lingüístico que se desea realizar, por ejemplo, a nivel de grafema, léxico, sintáctico o semántico. La entrada de esta etapa son los tokens y la salida pueden ser vectores, grafos o distribuciones probabilísticas, en dependencia de la forma de representación textual escogida.

Calcular una medida de semejanza o similitud: se debe escoger una medida de similitud que permita comparar las unidades textuales sobre las que se analiza el texto (morfema, palabra, oración o párrafo). La representación realizada en el paso anterior constituye la entrada para obtener los valores de similitud y como salida se obtienen las unidades textuales más semejantes.

Identificar los límites de los segmentos: el cálculo de las similitudes permite encontrar los límites de los tópicos y de esta forma realizar la tarea de segmentación. Algunos métodos empleados para encontrar los límites son las ventanas deslizantes, las cadenas léxicas, programación dinámica, agrupamiento jerárquico aglomerativo y divisivo. Tiene como entrada las unidades textuales similares y como salida los tópicos segmentados.

## 1.4 Detección de tópicos

La detección de tópicos inicialmente fue declarada como una tarea dependiente de la segmentación, debido a que la entrada de los algoritmos de detección estaba representada por segmentos (Allan 2002). Sin embargo, varias han sido las propuestas que utilizan como entrada el corpus textual sin segmentar, es decir, aplican técnicas que extraen los términos de los documentos, los agrupan y estos grupos representan los tópicos. De esta forma se define a la detección de tópicos como: "*la tarea que automáticamente encuentra nuevos tópicos en datos textuales*" (Huang et al. 2013), "*el descubrimiento de rasgos de palabras y fragmentos correspondientes a un tópico en los datos textuales, considerando a un tópico como un tema específico*" (Hamamoto & Pan 2005) o "*el proceso de agrupar documentos con tópicos similares en el mismo grupo"* (Ye et al. 2006). A continuación se expondrán los principales métodos de detección encontrados en la literatura y sus clasificaciones.

### 1.4.1 Clasificaciones de los métodos de detección

En las últimas dos décadas ha habido un auge en el surgimiento de los algoritmos para la detección de tópicos. Las tendencias en las implementaciones se encuentran fundamentalmente divididas en la clasificación supervisada y no supervisada, de acuerdo a si se posee o no un conjunto de documentos de entrada previamente asignados a tópicos. Si se tiene el conocimiento sobre este conjunto entonces la detección es supervisada, generalmente se requieren expertos del dominio para decidir cuándo un documento pertenece a un tópico predefinido; se parte de tópicos predefinidos y se entrenan algoritmos para identificar tópicos predefinidos en documentos textuales, luego se predicen los tópicos para nuevos documentos utilizando la experiencia adquirida en el entrenamiento. Por otra parte, para el enfoque no supervisado no se cuenta con ese conjunto inicial de documentos de entrenamiento, sino que los algoritmos propuestos descubren tópicos sin involucrar expertos del dominio (Dong et al. 2006). El enfoque supervisado tiene la desventaja que se requiere gran esfuerzo para entrenar los clasificadores; aunque una vez que el clasificador ha sido entrenado la detección es eficiente y efectiva (Dong et al. 2006). El enfoque no supervisado detecta los tópicos basándose en las similitudes entre sus unidades textuales.

Existen dos tipos de detección de tópicos de acuerdo al análisis de la colección: en línea o retrospectiva (Seijo et al. 2011). En la detección en línea (de inmediato), el sistema debe tomar la decisión según van llegando los documentos (artículos, noticias, mensajes, etc.) al sistema, mientras que en la retrospectiva (retardado), al sistema se le presenta de una vez toda la colección a procesar y la tarea consiste en estructurar esa colección en temas. Otra clasificación interesante ubica los métodos de detección de tópicos textuales en tres clases (Petkos, Aiello, et al. 2014):

Métodos documento-pivote: agrupan a documentos individuales de acuerdo a su similitud.

Métodos rasgo-pivote: agrupan términos de acuerdo a sus patrones de coocurrencia.

Modelos de tópicos probabilísticos: tratan el problema de la detección de tópicos como un problema de inferencia probabilístico.

Los métodos que pertenecen a cada una de estas clases representan los tópicos de forma diferente. Un documento-pivote representa un tópico con un conjunto de documentos relevantes; un método de rasgo-pivote representa tópicos con un conjunto de términos y un modelo de tópico probabilístico representa un tópico por distribuciones de términos. Es difícil concluir cuál clase produce los mejores resultados. La similitud de pares de documentos puede ser dominada

fácilmente por características ruidosas y por ello los objetos pueden ser incorrectamente agrupados al usar métodos documento-pivote. Los enfoques probabilísticos producen buenos resultados, sin embargo son computacionalmente muy costosos [8].

Los enfoques de documento-pivote pueden calcular típicamente una medida de similitud entre un par de documentos o entre un documento y un grupo. En el primer caso, si la similitud entre el documento entrante y el mejor documento coincidente que se encuentra en la colección está por encima de un umbral, entonces el documento entrante se adiciona al mismo grupo como el mejor documento coincidente. De lo contrario, se genera un nuevo grupo. Los enfoques de documento-pivote que aparecen en la literatura utilizan uno de estos dos casos. Típicamente, ellos difieren en que calculan la similitud en formas diferentes, aplican técnicas especiales para encontrar el mejor objeto o grupo coincidente, o utilizan algún paso post procesamiento (Petkos, Aiello, et al. 2014).

Los métodos rasgos-pivote intentan agrupar términos de acuerdo a sus patrones de coocurrencia. Generalmente, primero se seleccionan los términos a agrupar y se calculan los patrones de coocurrencia. En el segundo paso algunos calculan similitudes entre términos y son generalmente usadas en conjunción con algún procedimiento de agrupamiento. En la literatura se presenta una gran variedad de formas de seleccionar un conjunto de términos para ser agrupados, para calcular la similitud entre términos y ejecutar el agrupamiento (Petkos, Aiello, et al. 2014). La mayoría de los métodos rasgos-pivote, independientemente de emplear un mecanismo de selección de términos, examinan los patrones de coocurrencia (en varias formas) entre los pares de términos. En la práctica, dependiendo del algoritmo de agrupamiento que se emplee, existe la posibilidad de agrupar términos incorrectamente, especialmente cuando son términos comunes que pueden quedar enlazados a una gran cantidad de tópicos.

Los modelos de tópicos probabilísticos representan la distribución conjunta de tópicos y términos usando un modelo generativo probabilístico, el cual consiste de un conjunto de variables latentes que representan tópicos, términos, hiperparámetros, etc.

### 1.4.2 Métodos para la detección de tópicos

Se han enunciado tres tipos de métodos para la detección de tópicos en los textos (Petkos, Aiello, et al. 2014), los métodos basados en inferencia probabilística, los métodos basados en patrones de coocurrencia de términos y los métodos basados en el cálculo de similitudes entre documentos. Los primeros detectan tópicos a partir de una lista de tópicos prefijados; los segundos detectan

tópicos basados en la coocurrencia de los términos y los terceros consideran que un tópico incluye varios documentos. A continuación se expondrán algunos ejemplos de las tres categorías de métodos.

#### 1.4.2.1 Basados en inferencia probabilística

Los modelos de tópicos probabilísticos son modelos generativos no supervisados que modelan el contenido de los documentos como un proceso de generación de dos pasos: los documentos son observados como mezclas de tópicos latentes, mientras que los tópicos son distribuciones de probabilidad sobre palabras del vocabulario, y las palabras más representativas tienen las probabilidades más altas (Vulic et al. 2012). El modelo PLSA es capaz de capturar de forma eficiente información tópica de mensajes instantáneos pequeños y multilingües (Zhang et al. 2014). Para la detección de tópicos populares en microblogs se realizó una propuesta de detección global basada en dos pasos, construcción de una matriz de correlación de términos y se aplica PLSA basado en la técnica de reducción de dimensiones de factorización no negativa simétrica (Mat et al. 2014).

El análisis de cambio de sentimientos de tópicos es un nuevo problema de investigación que consiste de dos componentes principales: extraer las opiniones de un tópico determinado y detectar los cambios significativos de sentimiento de las opiniones en el tópico, así como identificar las razones posibles que causan ese cambio. Recientemente se propuso un marco de trabajo para el análisis de sentimiento a nivel de tópico (Topic Sentiment Change Analysis; TSCA) en el cual se aplica PLSA para extraer los tópicos del corpus (Jiang et al. 2011).

La mayoría de los modelos de tópicos como LDA son no supervisados, aunque también se han hecho propuestas supervisadas basadas en algoritmos de predicción (Blei & Mcauliffe 2008). Asimismo, se han desarrollado propuestas semisupervisadas basadas en LDA las cuales presentan como desventaja que no son aplicables cuando no se tiene un conocimiento previo de la colección (Ramage & Manning 2011).

Se ha expresado que modelos como LDA y PLSA no modelan aspectos apropiados de las revisiones que hacen los usuarios a objetos en la web (Titov & Mcdonald 2008), por ejemplo opiniones de propiedades de productos; sino que tienden a construir tópicos que clasifican globalmente términos (por ejemplo reproductores Mp3 versus iPods); por ello se ha propuesto el uso de modelos de tópicos multi-gránulos basados en LDA (multi-grain LDA; MG-LDA). MG-

LDA extrae aspectos de objetos de revisiones de los usuarios en línea. El modelo extrae y agrupa los aspectos en tópicos coherentes. Modelan tópicos locales y globales, la distribución de tópicos globales es fija para un documento, y la distribución de tópicos locales varía en el documento. La hipótesis consiste en que los tópicos locales capturan los aspectos y son utilizados entre distintos objetos presentes en el texto; los tópicos globales capturan las propiedades de los objetos revisados y pertenecen solo a tipos particulares de objetos. Una palabra en el documento es muestreada desde cualquiera de los tipos de tópicos. Por ejemplo, para el texto "…*el transporte público en Londres es directo, la estación se encuentra a 8 minutos de camino…o usted puede tomar el ómnibus por $1.50*" un tópico global es Londres, y un tópico local es la ubicación, especificada por el contexto relacionado a las palabras "transporte", "camino", "ómnibus" (Titov & Mcdonald 2008). Esta propuesta tiene dos salidas, los tópicos globales y los locales, y para cada uno el grupo de palabras más destacadas clasificadas por etiquetas. Estas últimas son asignadas manualmente a los tópicos con anterioridad (Titov & Mcdonald 2008).

Otras propuestas de detección de tópicos basadas en inferencia probabilística consideran por ejemplo, la evolución de los tópicos en el tiempo (Bolelli et al. 2009) y escenarios multilingües (Vulic et al. 2012) (Vulic 2011). Existen metodologías para agrupar mensajes de acuerdo a los tópicos tratados en ellos (Perez-Tellez et al. 2010).

Hattori y Nadamoto usan LDA para detectar tópicos, con el objetivo de identificar información relacionada con los medios sociales (Hattori & Nadamoto 2013). TwitterLDA es un modelo autor-tópico para descubrir tópicos en Twitter (Zhao 2011). Para conocer los tópicos emergentes analizar la reputación en línea de las compañías se han propuesto técnicas como la de Martin-Wanton y colegas (Martín-Wanton et al. 2013); su propuesta presenta un enfoque de aprendizaje de transferencia (transfer learning) no supervisado basado en inferencia probabilística.

#### 1.4.2.2 Basados en patrones de coocurrencia de términos

Se han realizado varias investigaciones para la detección de tópicos basados en patrones de coocurrencia de términos (Petkos, Aiello, et al. 2014) (Kleedorfer et al. 2008) (Sayyadi & Raschid 2013) (Liu et al. 2014), estas se diferencian fundamentalmente en el mecanismo empleado para la selección de los términos, los cuales son posteriormente agrupados de acuerdo al tópico al cual están relacionados. Algunos de estos mecanismos son modelos de reducción de dimensiones como la factorización de matrices no negativas, técnicas para obtener coocurrencia de palabras claves, modelos de redes neuronales, modelo n-gram y otros.

La factorización de matriz no negativa (Non-Negative Matrix Factorization; NMF) es un algoritmo de factorización de matrices que se enfoca en el análisis de matrices de datos cuyos elementos son no negativos[12]. Este algoritmo, al igual que como sucede con LDA, LSA y PLSA, también puede ser utilizado como una forma de representación textual, e incluso como una forma de realizar reducción de dimensionalidad. Esta técnica representa una matriz por la factorización de dos matrices como muestra la ecuación (0.5). Lee y Saung plantean que dado un conjunto de vectores de datos *n*-dimensionales, los vectores se ubican en las columnas de una matriz $X_{nxm}$ donde *m* es la cantidad de ejemplos del conjunto de datos. Esta matriz es luego factorizada en una matriz $W_{nxr}$ y una matriz $H_{rxm}$. Usualmente se selecciona *r* para que sea menor que *n* o *m*, de tal forma que *W* y *H* son más pequeñas que la matriz original *X* (Lee et al. 2000). Se puede alcanzar una buena aproximación si los vectores bases descubren una estructura latente u oculta en el dato.

$$X \approx WH \tag{0.5}$$

Esta técnica se ha aplicado en diferentes dominios. En la minería de textos se ha utilizado para el agrupamiento de documentos por tópicos. Para este enfoque se considera que cada documento en el corpus pertenece a un tópico o está relacionado a varios tópicos, de esta forma se proyecta el corpus en un espacio semántico de *k* dimensiones, donde cada dimensión es un tópico, y cada documento puede ser representado como una combinación lineal de *k* tópicos. NMF se aplica para encontrar la estructura semántica latente del corpus e identificar los grupos de documentos (Xu et al. 2003). En el Anexo 9 se muestra un ejemplo de NMF donde se representan tópicos de los documentos. Algunos ejemplos del uso de esta técnica son la detección de tópicos en letras de canciones (Kleedorfer et al. 2008) y la detección de tópicos emergentes en mensajes de Twitter (Prasad et al. 2011) (Yan et al. 2013).

KeyGraph es un método basado en grafos para la detección de tópicos aplicado a colecciones de datos de medios sociales, el cual representa una colección de documentos como un grafo de coocurrencia de palabras claves. Aplica un algoritmo de detección de comunidades para agrupar palabras claves coocurrentes en comunidades y cada comunidad forma una constelación de palabras claves que representan un tópico (Sayyadi & Raschid 2013). También se han dedicado esfuerzos para identificar tópicos en los textos generados por los usuarios de comunidades de salud

[12] Matriz de enteros o reales donde cada elemento es un número no negativo (mayor que cero). http://mathworld.wolfram.com/NonnegativeMatrix.html

en línea. Para ello se emplea el modelo VSM como forma de representación textual y el método de agrupamiento EM (Expectation Maximization) para formar los grupos de instancias (Lu et al. 2013).

La mayoría de los algoritmos TDT no consideran la distancia temporal entre un par de tópicos y el efecto mutuo entre términos de tópicos altamente correlacionados (Holz & Teresniak 2010) (Wang & Mccallum 2006) (Huang et al. 2013). Una propuesta para responder a esta limitante consiste en aplicar el agrupamiento jerárquico basado en grafos de conceptos (Hierarchical Clustering on Concept Graph; HCCG), donde un concepto se utiliza para describir un tópico y este está representado al menos por dos palabras claves (Huang et al. 2013). El procedimiento de HCCG genera un grafo de conceptos, realiza un pre-agrupamiento usando el algoritmo de paso único (Single-Pass[13]) modificado para ejecutar el pre-agrupamiento del grafo conceptual y luego realiza el agrupamiento en grafos conceptuales usando un algoritmo de agrupamiento jerárquico aglomerativo (Agglomerative Hierarchical Clustering; AHC) para pre-agrupar los resultados. Una variante similar se presenta en (Liu et al. 2014).

La minería de patrones frecuentes (Frequent Pattern Mining; FPM) posee un conjunto de técnicas para descubrir patrones frecuentes en grandes bases de datos transaccionales. En el contexto de métodos rasgo-pivote se buscan términos que frecuentemente ocurren juntos, la representación de documentos se enriquece con los patrones que involucran los términos en el documento, ya que los patrones comunican relaciones entre términos que pueden perderse si solo se consideran los términos en el documento. FPM examina la coocurrencia simultánea entre cualquier cantidad de términos mayor a dos mientras que los métodos típicos rasgo-pivote examinan solo coocurrencia en pares. Una investigación reciente propone el uso de la Minería de Patrones Frecuentes Suaves (Soft Frequent Pattern Mining; SFPM) para detectar tópicos la cual presenta un algoritmo para suavizar patrones de coocurrencia de grado mayor a dos (Petkos, Aiello, et al. 2014).

En (Rajaraman & Tan 2001) se propuso un algoritmo incremental para resolver el problema de la detección y seguimiento de tópicos utilizando una red neuronal difusa (Adaptive Resonance Theory; ART). El aprendizaje competitivo y constructivo de redes neuronales se utilizó también en el descubrimiento de tópicos basado en algoritmos de agrupamiento (Seo & Sycara 2004).

[13] El algoritmo de agrupamiento de paso-único (Single-Pass) es un tipo de agrupamiento incremental donde se procesan los documentos uno a uno (Xiaolin et al. 2013).

Una tendencia reciente en el campo de la detección de tópicos es encontrar tópicos populares (hot topic). Para ellos se ha utilizado una metáfora biológica basada en las etapas del ciclo de vida (nacimiento, crecimiento, decadencia y muerte) y así simular el ciclo de vida de los tópicos. Esta teoría del envejecimiento (aging theory) expresa que el valor de una función de energía muestra cuán activo es un tópico; además que la energía de un tópico aumenta cuando el tópico es popular y disminuye con el tiempo (Zheng & Li 2009) (Chen et al. 2003) (Cataldi et al. 2010). Otra propuesta para la detección de tópicos populares se basada en un modelo *n*-gram. Estos modelos capturan los patrones de coocurrencia de palabras locales contiguas en oraciones (Wang & Wang 2013). BNgram es otro método de detección de tópicos que aplica técnicas *n*-gram en mensajes de Twitter (Aiello et al. 2013).

#### 1.4.2.3 Basados en similitudes entre documentos

La idea básica de este enfoque es utilizar algoritmos que agrupen documentos y traten cada grupo como un tópico.

En el contexto de la detección de tópicos en flujos de noticias una propuesta basada en similitudes entre documentos fue dada por (Pons-porrata et al. 2002), la cual se basa en la aplicación del algoritmo de agrupamiento $\beta_0$ compacto incremental. Se caracteriza por usar oraciones temporales extraídas de los documentos textuales para definir la función de similitud y trabajar de forma jerárquica. Este método calcula la similitud coseno entre términos y presenta dos funciones para medir la similitud global entre pares de documentos. El criterio de agrupamiento aplicado se apoya en relaciones topológicas entre documentos. La salida del algoritmo son grupos de documentos que representan los eventos identificados. El sistema JERARTOP permite la detección de tópicos en línea utilizando el algoritmo de agrupamiento propuesto (Pons-porrata et al. 2004). Otras propuestas de algoritmos de agrupamiento no supervisados relacionadas a este se describen en (Porrata 2004) (Pons-porrata & Berlanga-llavori 2007) (Lazo-cortes et al. 2001).

Un enfoque interesante de detección de tópicos en Twitter basados en similitudes entre documentos fue propuesto por (Petkos, Papadopoulos, et al. 2014). Este pre-procesa los tweets y utiliza un algoritmo que agrupa en dos niveles: según el URL al que pertenezca el tweet y su respuesta. Posteriormente se ordenan los tópicos por la cantidad de tweets que contienen. Por cada tópico se extraen los títulos, las palabras claves y las imágenes referentes a cada tópico.

### 1.4.3 Consolidación de las etapas para la detección de tópicos

A partir del estudio de los métodos de detección encontrados en la literatura, se realizó, como parte de esta investigación, una generalización del proceso de segmentación en cinco etapas principales (Torres & Arco 2015a), las cuales están representadas en un diagrama que se muestra en el Anexo 8 b). Las dos primeras etapas son similares a las descritas en el epígrafe 1.3.3; seguidamente se describirán el resto de las etapas.

Aplicar métodos de reducción de la dimensionalidad: aplicar métodos como PLSA, LDA, NMF, técnicas de reducción de dimensiones, de patrones de coocurrencia, de agrupamiento u otros. La entrada son formas de representación textual y la salida pueden ser grupos de palabras claves, fragmentos de textos o subtópicos.

Agrupar unidades textuales por tópicos: emplear un algoritmo de agrupamiento que permita agrupar las unidades textuales por tópicos. La entrada son las representaciones reducidas que se obtienen de la etapa anterior y la salida son conjuntos de palabras o documentos por tópicos.

Etiquetar grupos por tópicos: se refiere a otorgarle a cada grupo una breve descripción que identifique el tópico que tratan. Tiene como entrada los grupos de tópicos y como salida puede ser una o más palabras que identifican cada tópico.

## 1.5 Conclusiones parciales del capítulo

La segmentación de textos por tópicos es una tarea del campo de TDT que tiene como objetivo dividir el texto en unidades textuales semánticamente iguales, para identificar los tópicos y subtópicos que posee un documento. La tarea de detección de tópicos permite descubrir el tema del cual trata un documento y aplica fundamentalmente técnicas de agrupamiento para obtener los conjuntos de tópicos.

Para segmentar y detectar tópicos es necesario conocer los modelos de representación textual, los cuales se presentan mayormente en forma vectorial, grafos y probabilística. Otras representaciones pueden considerarse también métodos de detección como PLSA y LDA; sus implementaciones son muy usadas pero presentan desventajas; por ejemplo, no conservan el orden de las palabras, necesitan un corpus de entrenamiento y requieren una cantidad específica inicial de tópicos. VSM es el modelo que sentó las bases para la representación de textos y ha sido ampliamente utilizado en algoritmos para la segmentación y detección de tópicos. Sin embargo, esta representación tiene la desventaja que se pierde el orden de los términos en la unidad textual que se utilice, por lo que

se han realizado propuestas donde se incluyen elementos semánticos que permiten obtener mayores relaciones entre las unidades textuales.

Los métodos de segmentación y detección de tópicos se basan fundamentalmente en la cohesión léxica, cálculos probabilísticos y técnicas de agrupamiento, siendo más comunes los enfoques no supervisados. Predominan las investigaciones para cada tarea por separado; no obstante, se han hecho algunas propuestas que combinan ambas tareas. Por ejemplo, agrupar los segmentos relacionados a un mismo tópico, y etiquetarlos con una breve descripción. Además, mantener la estructura del documento a través de segmentos permite capturar la naturaleza semántica del texto.

# Capítulo 2 La segmentación y detección de tópicos para contribuir al análisis de sentimientos

En este capítulo se describirán los principales elementos del análisis de sentimiento y se abordarán aquellos trabajos que han empleado técnicas de detección de tópicos en colecciones de opiniones. Se describirá la aplicación PosNeg Opinion que permite detectar la polaridad de las opiniones, señalando las limitaciones que presenta. De ahí que se propone un esquema que utiliza técnicas de TDT para contribuir al análisis de sentimiento por tópicos.

## 2.1 Análisis de sentimiento

La minería de opinión o el análisis de sentimiento es un área de investigación que se encarga de explorar y descubrir la información subjetiva generada por el usuario. Se ha definido como "*la tarea de detectar, extraer y resumir opiniones, polaridades y/o emociones, basadas en la presencia o ausencia de rasgos de sentimiento*" (Moens et al. 2014). Está motivada por el hecho que las opiniones de usuarios no expertos pueden servir como complemento de puntos de vistas publicados en diferentes medios; y las valoraciones sobre productos y servicios pueden tener gran impacto económico tanto para consumidores como organizaciones (Moens et al. 2014).

De forma general una opinión se representa por la combinación de cinco componentes: el nombre de una entidad, un aspecto de la entidad, la orientación de la opinión sobre el aspecto de la entidad, el propietario de la opinión, y el tiempo donde la opinión es expresada por el propietario de la opinión (Zhang & Liu 2014). La orientación de la opinión se conoce por orientación del sentimiento, polaridad de la opinión u orientación semántica y puede clasificarse en positiva, negativa o neutral, o ser expresada con diferentes niveles de intensidad (Liu 2010). Un término tiene polaridad cuando porta información subjetiva positiva o negativa. Los cálculos de polaridad en la minería de opinión se pueden estructurar en varias fases: detectar la subjetividad, clasificar la opinión, determinar la fuerza de la opinión, determinar la fuente de la opinión, determinar el objetivo de la opinión y resumir las opiniones y/o visualizar gráficamente los resultados.

La tarea de detección de la polaridad de las opiniones ha sido muy estudiada actualmente (Lin 2011) (Thompson 2014) (Dueñas et al. 2013) (Toh et al. 2014). Existen dos enfoques principales para tratar de resolver automáticamente la polaridad de un texto: el aprendizaje supervisado y la orientación semántica (Moghaddam & Ester 2013). En el primero se construye un clasificador a partir de un conjunto de entrenamiento formado por una colección de textos etiquetados, donde se

expresa una opinión favorable o desfavorable. Y en el segundo se emplean diccionarios (lexicon) donde cada palabra se encuentra etiquetada con su orientación semántica, permitiendo medir en qué grado esa palabra es positiva o negativa. Existen tres niveles fundamentales para desarrollar el análisis de sentimiento (Zhang & Liu 2014), ellos son: nivel de documento, considerando que cada documento expresa una opinión negativa o positiva de forma general; nivel de oración, donde se asume que las oraciones expresan opiniones positivas y negativas; y el nivel de aspecto, considerando que las opiniones son detalladas y expresan sentimientos sobre diferentes aspectos de entidades y de las entidades en sí mismas. Este nivel está muy relacionado a la detección de tópicos, debido a que los aspectos pueden representar tópicos en el texto. El nivel de documento presenta como limitante que no da detalles de las preferencias de los usuarios, solo se tiene una valoración global de la polaridad de la opinión pero no se sabe sobre cuáles aspectos específicos se tiene una opinión positiva o negativa. El nivel de oración es un nivel más detallado pero solo tiene en cuenta las palabras de opinión. La cantidad de palabras positivas y negativas son contadas a partir de las oraciones identificadas. Si predominan las palabras positivas la opinión es positiva, de lo contrario son las negativas y en caso que sean iguales es neutral. El nivel de aspecto se define a partir de los atributos de las entidades, para los que se identifican opiniones negativas o positivas. Si el texto está mal escrito gramaticalmente puede que no de buenos resultados. Sin embargo, es el modelo que brinda mayor información de los tres mencionados (Sharma & Chitre 2014). Para obtener un análisis de opinión más preciso se recomienda el nivel de aspecto y para ello se han propuesto combinar tres subtareas (Zhang & Liu 2014): identificar y extraer entidades en textos, identificar y extraer aspectos de entidades y posteriormente determinar polaridades de sentimiento en entidades y aspectos de entidades.

### 2.1.1 Extracción de términos de aspectos

El análisis de sentimiento basado en aspectos contribuye a obtener polaridades de las opiniones asociadas a aspectos de las entidades. Un aspecto ha sido definido como los componentes o atributos de una entidad (producto, servicio, persona, evento, organización o tópico); y una expresión de aspecto es una palabra o frase que aparece en el texto indicando un aspecto (Zhang & Liu 2014). Por ejemplo, en el dominio del teléfono celular un aspecto puede ser la calidad de la voz. Existen varias expresiones que pueden indicar el aspecto, por ejemplo “sonido”, “voz”, y “calidad de voz”. Las expresiones de aspectos son usualmente sustantivos y frases sustantivas, pero pueden ser también verbos, frases verbales, adjetivos y adverbios (Zhang & Liu 2014). Las

tareas extracción de aspectos y extracción de entidades[14] son fundamentales para la minería de opinión a nivel de aspecto.

Existen tres enfoques para la extracción de aspectos: reglas del lenguaje, modelos de secuencia y modelos de tópicos probabilísticos (Zhang & Liu 2014). Las reglas del lenguaje se basan fundamentalmente en encontrar sustantivos frecuentes y frases sustantivas como aspectos frecuentes (Hu et al. 2004). Para encontrar sustantivos y frases sustantivas frecuentes se utilizan etiquetadores de partes del discurso, y posteriormente se cuentan los sustantivos encontrados y se almacenan los más frecuentes. En (Hu et al. 2004) se precisa que la razón para usar este enfoque es que las personas comentan diferentes aspectos de un producto y el vocabulario que usan generalmente converge. Los modelos de secuencia estiman la extracción de aspectos como una tarea de etiquetamiento de secuencia, dos ejemplos son los campos aleatorios condicionales (Conditional Random Fields; CRF) y los modelos ocultos de Markov (Hidden Markov Model; HMM). Los modelos de tópicos probabilísticos se basan en la idea que los documentos son mezclas de tópicos, y cada tópico es una distribución de probabilidad de palabras. La ventaja que presentan es que palabras diferentes que expresan los mismos aspectos o expresiones de aspectos pueden ser agrupadas automáticamente baja el mismo aspecto (Zhang & Liu 2014).

El análisis de la polaridad basada en tópicos o aspectos[15] es útil para realizar resúmenes de opiniones y clasificar entidades basadas en opiniones. Así como realizar predicciones de la polaridad de las opiniones sobre aspectos específicos (Wang et al. 2011); por ejemplo, para valorar opiniones sobre un hotel algunos aspectos que los revisores consideran más importantes en las evaluaciones son "limpieza", "ubicación" y "servicio", y las puntuaciones que le otorgan los usuarios indican si las opiniones son negativas o positivas. Otra propuesta con este perfil es una herramienta para el análisis de evaluaciones de aspectos ocultos (Latent Aspect Rating Analysis; LARA), la cual utiliza una etapa de segmentación de aspectos para obtener los aspectos que trata el texto y una segunda etapa donde aplican un modelo probabilístico generativo para inferir las evaluaciones de los revisores de opiniones por cada aspecto a partir de la evaluación general de la opinión (Wang et al. 2010) (Wang et al. 2011). Recientemente, se ha hecho énfasis también en cómo combinar técnicas como la extracción de aspectos y la detección de las categorías de aspectos

[14] Para la extracción de entidades predominan los métodos supervisados y semi-supervisados (Zhang & Liu 2014).
[15] En esta tesis se considerarán estos términos indistintamente

para posteriormente determinar la polaridad de cada término y a su vez de la categoría a la que pertenece (Zhang et al. 2015).

La tarea de ABSA ha sido generalizada en tres etapas (Pavlopoulos 2014). La primera es la extracción de términos de aspectos para detectar términos únicos y múltiples referentes a las entidades mencionadas en un texto, para lo cual se han propuesto métodos basados en identificar sustantivos de la opinión y el uso de herramientas para aprender representaciones vectoriales de palabras basadas en redes neuronales[16]. Una segunda etapa agrega términos de aspectos similares utilizando algoritmos de agrupamiento jerárquico y WordNet. Por último, se estima la polaridad de los términos de aspectos, para lo cual es muy usado un clasificador SVM.

La mayoría de las técnicas para la extracción de términos de aspectos son supervisadas, por lo que se requiere de un corpus de entrenamiento inicial, el cual no siempre está disponible. Otras desventajas de estas técnicas es que se necesitan procesar opiniones con sentimiento explícito y la portabilidad es limitada debido a que su funcionamiento puede cambiar significativamente cuando el mismo método es aplicado a dominios diferentes (Gangemi et al. 2014).

### 2.1.2 PosNeg Opinion

PosNeg Opinion 1.0 es una herramienta que sigue cinco etapas para la detección no supervisada de la polaridad de opiniones en Inglés y Español (Amores 2013). La Etapa 1 es la encargada de leer las opiniones que fueron especificadas en el XML de entrada y seleccionar los términos que aporten información útil. En la Etapa 2, se parte de cada término que aporta información útil, éste se lematiza y se desambigua lexicalmente. Posteriormente, en la Etapa 3 se traducen los términos seleccionados en la Etapa 2, obteniendo todas las acepciones del término en inglés. En la Etapa 4 se calcula la polaridad de los términos, considerando la polaridad de cada una de las acepciones del término. Así, en la Etapa 5, al terminar de analizar todos los términos y sus acepciones, la opinión cuenta con un valor positivo y otro negativo, los cuales son comparados, y se toma como polaridad de la opinión el mayor valor (Amores, Arco, et al. 2015). PosNeg Opinion 2.0 tiene el mismo propósito de PosNeg Opinion 1.0, la diferencia radica en que en esta segunda versión se fusionan las etapas 3 y 4 a partir de la aplicación del SpanishSentiWordNet (Amores, Borroto, et

[16] http://code.google.com/p/word2vec/

al. 2015) para procesar opiniones en Español sin la necesidad de combinar el SentiWordNet[17] con otras herramientas.

Ambas versiones de PosNeg Opinion permiten realizar un análisis de la polaridad de forma local (nivel de oración) y de forma global (nivel de documento). En el primer caso se realiza por oraciones y se identifican solo las palabras que indican la opinión. Por ejemplo, en su versión 1.0 analiza la siguiente oración "*La hp 2000 tiene muy buena batería*" y obtiene solo las palabras "*muy*", "*buena*" y "*batería*". Inicialmente se eliminan los términos "*la*", "*hp*", "*2000*" y "*tiene*" por ser palabras que no ofrecen información para la detección de la polaridad de la opinión. Se buscan las acepciones en Inglés de los términos obtenidos y se suman sus puntuaciones en SentiWordNet para obtener la mayor polaridad, de esta forma se determina que la opinión tiene polaridad positiva. PosNeg Opinion es capaz de calcular la polaridad de una oración, pero tiene la limitante que no es capaz de determinar cuál es el tópico al que se refiere el criterio emitido, tampoco la entidad y los aspectos que se abordan de la misma.

Supongamos que se desea procesar la siguiente opinión con PosNeg Opinion: "*El hotel es fantástico y muy elegante. El servicio del conserje es ineficiente. La habitación no estaba limpia cuando llegamos. El restaurante está clasificado como uno de los 3 mejores de la ciudad*." En este caso PosNeg Opinion calcula la polaridad de toda la opinión mediante el voto global positivo y negativo para la opinión. Las oraciones 1 y 4 tienen polaridad positiva, mientras que las oraciones 2 y 3 tienen polaridad negativa, entonces, ¿cuál será la polaridad de la opinión? ¿Ayudará un valor global de polaridad al gerente del hotel a tomar alguna decisión? Este ejemplo evidencia que PosNeg Opinion no permite calcular la polaridad de las opiniones por tópicos, siendo esta una de sus principales desventajas.

Aunque PosNeg Opinion permite obtener excelentes valores de precisión y exactitud en la detección de la polaridad de las opiniones solo es capaz de clasificar las opiniones expresadas en una oración o la opinión que expresa el texto en su totalidad, y por tanto no realiza un análisis del sentimiento por tópicos. De ahí que se hayan realizado estudios sobre cómo los métodos de TDT pueden ser útiles en la minería de opinión y contribuir al análisis de sentimientos por tópicos.

[17] Recurso léxico para la minería de opinión, asigna puntuaciones de sentimiento a cada synset de WordNet. http://sentiwordnet.isti.cnr.it

## 2.2 Métodos de segmentación y detección de tópicos aplicados a la minería de opinión

Varios investigadores de minería de opinión han estudiado las potencialidades que ofrece la inclusión de técnicas de detección de tópicos en la minería de opinión (Zhang & Liu 2014; Hattori & Nadamoto 2013; Titov & Mcdonald 2008; Cambria et al. 2013; Pang & Lee 2008; Dueñas et al. 2013). Los enfoques se apoyan principalmente en la frecuencia y relación entre los términos, así como el empleo de modelos probabilísticos (Moghaddam & Ester 2013). Un método de detección de tópico basado en el esquema de pesos (Pointwise Mutual Information; PMI) y la distribución de frecuencias de términos fue presentado en (Cai et al. 2008). El método sigue la idea de combinar un enfoque de clasificación de sentimiento con un enfoque de detección de tópicos que descubra términos que son altamente relacionados con diferentes categorías de sentimiento. Específicamente, para identificar las palabras que representan los tópicos siguen los siguientes pasos:

1. Clasificar documentos en categorías de positivo, negativo y neutral usando técnicas de clasificación de sentimiento basadas en puntuaciones obtenidas de WordNet[18] e Inquirer[19].
2. Identificar todas las palabras en los documentos y filtrar las palabras vacías y las palabras de sentimiento para de esta forma obtener como candidatos de tópicos a las palabras que no corresponden a un sentimiento.
3. Calcular la frecuencia de las palabras obtenidas en el paso 2 en cada categoría de sentimiento así como a través de todas las categorías.
4. Calcular el valor de PMI de las palabras en cada categoría de sentimiento.
5. Combinar la frecuencia de palabras en cada categoría con su valor de PMI y seleccionar las primeras palabras más frecuentes con mayor PMI como palabras de sentimiento final.

Algunos estudios sugieren que manejar las tareas del análisis de sentimiento y la detección de tópicos puede beneficiar el funcionamiento de sistemas que ejecutan el análisis de sentimiento sobre los tópicos que trata un documento (Cambria et al. 2013) (Ren & Han 2014) (Jiang et al. 2011) (Gangemi et al. 2014). Por ejemplo, un pasaje de un documento donde no se consideren los tópicos puede contener información afectiva irrelevante y crear una polaridad de sentimiento global sobre el tópico principal. Además, un documento puede contener información en múltiples tópicos que sea de interés del usuario. En tales casos, es importante identificar tópicos y separar

[18] http://wordnet.princeton.edu/
[19] http://www.theinquirer.net

las opiniones asociadas con cada tópico. Las opiniones y sentimientos no ocurren solo a nivel de documento, ni están limitados a un único objetivo. Un documento puede contener opiniones negativas y positivas a través de uno o más tópicos (Cambria et al. 2013).

El análisis de cambio de sentimientos de tópicos consiste en dos componentes principales: extraer las opiniones de un tópico determinado y detectar los cambios significativos de sentimiento de las opiniones en el tópico, así como identificar las razones posibles que causan ese cambio. En (Jiang et al. 2011) se propuso un enfoque para el análisis de sentimiento a nivel de tópico en el cual se aplica el modelo de PLSA para extraer los tópicos de un corpus. Otra propuesta para detectar tópicos de las comunidades en línea utiliza el modelo LDA (Hattori & Nadamoto 2013). Sin embargo, los modelos LDA y PLSA no modelan aspectos apropiados de las revisiones que hacen los usuarios a objetos en la web; sino que tienden a construir tópicos que clasifican globalmente los términos. En (Titov & Mcdonald 2008) se presenta el modelo de tópico multi-gránulos basado en LDA (MG-LDA) que supera las limitaciones anteriores ya que extrae aspectos de objetos de revisiones de usuarios en línea.

Un elemento importante para detectar la polaridad es hacerlo para segmentos de los textos. Existen análisis de opinión a nivel de segmento para identificar las oraciones subjetivas y después calcularles su polaridad (Pang & Lee 2008). Algunas propuestas utilizan bases probabilísticas para detectar tópicos y sentimientos; y asumen que los tópicos se generan dependiendo de distribuciones de sentimientos y las palabras de acuerdo a pares de tópicos-sentimiento (Lin et al. 2012). Otros trabajos utilizan la tarea de detección de tópicos para identificar tendencias en la Web. Para ello se basan principalmente en inferir estructuras tópicas de los documentos y recuperar documentos sobre los que se han dado opiniones, y extraen la información de sentimientos para cada tópico (Dueñas et al. 2013).

Dado un documento o un conjunto de documentos, estos pueden contener opiniones sobre múltiples tópicos que son interesantes para los usuarios, en este escenario es muy importante detectar tópicos correctamente, y separar las opiniones asociadas con cada tópico. En (Gangemi et al. 2014) se presenta el método Sentilo, el cual es independiente del dominio, automático, no supervisado y enfocado en representar la semántica de oraciones al modelar los roles que juegan sus elementos respecto a un modelo de oraciones de opinión. La calidad de algoritmos de análisis de sentimiento mejora cuando se consideran rasgos semánticos; por ello, la propuesta primero realiza una representación semántica RDF-OWL de una oración de opinión y luego la anotan con

un modelo que identifica el emisor de la opinión, tópico y sub-tópicos. En este caso se considera a los eventos o situaciones como tópicos principales, y cuando no encuentran eventos o situaciones, entonces buscan objetos u otras entidades. Cuando el tópico principal es una situación, las entidades involucradas son subtópicos, mientras que cuando el tópico principal es un evento, las entidades que tienen una relación de dependencia con ellos son subtópicos. En esta propuesta se emplean recursos léxicos como SenticNet[20], SentiWordNet y VerbNet[21].

Considerando las limitaciones de PosNeg Opinion y los beneficios de los métodos de TDT en el análisis de sentimiento, así como los elementos del ASBA, se propone un esquema general que explota las ventajas de estas técnicas de una manera integrada, con el fin de descubrir tópicos en una opinión o conjunto de opiniones para facilitar el análisis de la polaridad asociada a ellos.

### 2.3 Esquema general para realizar análisis de sentimiento por tópicos

Generalmente para el descubrimiento de tópicos en documentos textuales se representa a un tópico de tres formas diferentes: palabras claves, conjuntos de términos y segmentos de textos. El primer y el segundo caso son los más empleados por los métodos de detección que se han desarrollado en los últimos años. Por ejemplo, los conjuntos de términos es una representación característica de modelos probabilísticos como PLSA y LDA; sin embargo estos modelos presentan la desventaja que se basan en la representación BOW y por tanto no mantienen el orden de las palabras, además de que requieren que se le especifique una cantidad de tópicos inicial, así como un corpus de entrenamiento. La representación por palabras claves es un enfoque que es útil cuando se quieren identificar frases sustantivas, bigramas o entidades nombradas que pueden representar el tópico del cual trata un texto. En el contexto de las opiniones se pueden observar las palabras claves como expresiones de aspectos en las que se resaltan términos representativos de una entidad sobre la que se opina. Sin embargo, algunos autores hacen énfasis en la necesidad de mantener la estructura del documento a través de segmentos para capturar la naturaleza semántica del texto (Jameel 2014); para alcanzar esta meta se han propuesto técnicas de detección basadas en el uso de segmentos, como por ejemplo el agrupamiento de segmentos (Tagarelli & Karypis 2012) (Joty et al. 2013) (Jameel 2014).

[20] Es una base de conocimiento que provee un conjunto de significados, categorías de emociones y un valor de polaridad para alrededor de 30000 conceptos, http://sentic.net/

[21] Es un diccionario de verbos en inglés, es independiente del dominio y presenta enlaces con otros recursos léxicos como WordNet. http://verbs.colorado.edu/~mpalmer/projects/verbnet.html

Estas ideas constituyen una motivación para el presente trabajo en el que se propone fusionar las dos tareas de segmentación y detección con el objetivo de obtener tópicos de opiniones, donde cada tópico estará representado por un grupo de segmentos. Así, con el objetivo futuro de lograr un análisis de sentimiento a partir de tópicos, los grupos de segmentos detectados brindarán una representación de las opiniones en la cual se obtendrán los criterios otorgados para un mismo tema, en los cuales se preserva el orden de las palabras escritas por el usuario y consecuentemente el significado semántico de lo que se expresa. El análisis de la polaridad de las opiniones se podrá realizar a partir de los segmentos que conforman el tópico, al contrario de como ocurriría en un enfoque de conjuntos de términos o palabras claves en el cual se obtienen los tópicos pero sin conocer a que parte del texto pertenecen.

Los enfoques de detección que no segmentan obtienen conjuntos de términos sin orden específico que representan un tópico de forma general; y en los enfoques que segmentan y no detectan solo se obtiene el texto con fragmentos de tópicos identificados. La mayor parte de las propuestas para aplicar técnicas de detección de tópicos en opiniones no segmentan, se basan en modelos de inferencia probabilística, el empleo de representaciones semánticas, consideran la frecuencia y relación entre los términos o incluso prefijan categorías de tópicos como se mencionó en el epígrafe anterior (Cai et al. 2008) (Hattori & Nadamoto 2013) (Gangemi et al. 2014).

En la Fig. 1 se muestra un esquema general que se propone como parte de la presente investigación para obtener los tópicos de las opiniones a través del uso de técnicas de segmentación y de detección de forma conjunta. Como primera etapa se establece identificar las unidades textuales, luego el pre-procesamiento del texto y la representación de documentos debido a que estas tres etapas son comunes tanto para la segmentación como para realizar la detección de tópicos. Con el objetivo de encontrar los límites de los tópicos y de esta forma identificar los segmentos se propone la etapa de segmentar. Luego, se presenta la etapa de agrupamiento para encontrar aquellos segmentos que se relacionan entre sí y de esta forma corresponden a un tópico determinado. Pero antes de agrupar es necesaria una etapa previa de representación en la que se presenten los segmentos obtenidos en un modelo computacional para su posterior agrupamiento. Se sugiere realizar finalmente un etiquetamiento por cada grupo obtenido para describir con pocos términos cada tópico encontrado.

La entrada del esquema es un corpus de opiniones y la salida son fragmentos del corpus que representan tópicos y las etiquetas asociadas (Torres & Arco 2016) (Torres et al. 2015). A continuación se describirán en detalle cada una de las fases que conforman el proceso.

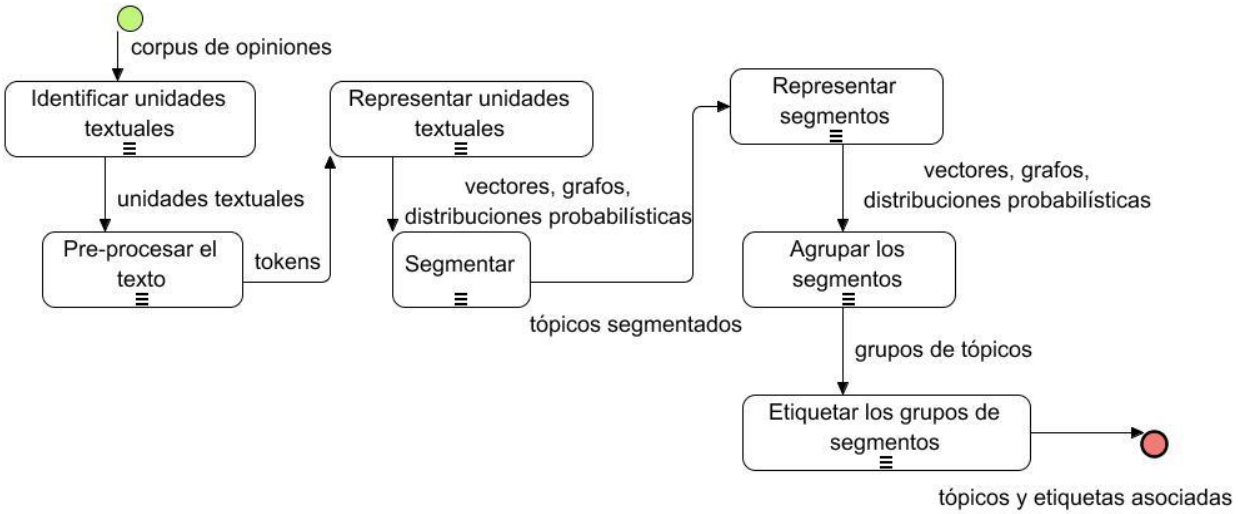



**Fig. 1 Esquema general de segmentación y detección de tópicos para el análisis de sentimiento.**

### 2.3.1 Etapa 1 Identificar unidades textuales

Las unidades textuales que comúnmente se utilizan para el análisis de textos son las palabras, oraciones, párrafos o los bloques. Estos últimos se refieren a fragmentos en los que se divide el texto; se pueden definir, por ejemplo, especificando una cierta cantidad de tokens o de oraciones. El tamaño de los bloques de textos de opiniones se pudiera definir considerando un por ciento de palabras de la longitud del texto (de la longitud promedio de todas las opiniones) o definir el tamaño con un por ciento de palabras de la longitud por opinión. En esta etapa se seleccionó como unidad textual las oraciones, debido a que los textos de opiniones se caracterizan por presentar un estilo de escritura en forma de composición, con oraciones cortas, escritas de forma informal sin una estructura específica, contenidas en un solo párrafo. Además, los algoritmos de segmentación analizados[22] utilizan esta unidad textual como entrada para segmentar.

La desambiguación de los límites de las oraciones es una tarea que ha sido estudiada con el objetivo de aplicar métodos para identificar los límites de oraciones. Los signos de puntuación son ambiguos, por ejemplo, un punto puede denotar un punto decimal, una abreviación, direcciones de correo electrónico, el fin de una oración, etc. La mayoría de los sistemas usan gramáticas de

[22] TextTiling y C99 son descritos en la etapa 4

expresiones regulares y reglas de excepción para desambiguar los signos de puntuación. Dos trabajos iniciales fueron los presentados en (Palmer & Hearst 1994) y (Reynar et al. 1997). En el primero se propuso un algoritmo basado en una red neuronal y un vocabulario que contiene información de partes del discurso. En el segundo se presenta una solución basada en un modelo de máxima entropía que no requiere reglas manuales, sino que estima la distribución de probabilidad de un token y el contexto que lo rodea. Para ello utilizan plantillas contextuales como prefijos, sufijos y grados honoríficos, entre otros; luego estiman la probabilidad de identificar los límites. De forma general existen tres enfoques para detectar los límites de las oraciones: basados en reglas, basados en técnicas de aprendizaje automático supervisado y no supervisado.

Existen varias herramientas de código abierto que detectan oraciones de forma automática, tales como Stanford CoreNLP[23], Apache OpenNLP[24], UIMA[25], GATE[26], LingPipe[27], y otras (Read et al. 2012). Los marcos de trabajo UIMA y GATE tienen una curva de aprendizaje mucho mayor que herramientas como Stanford CoreNLP (Manning et al. 2014). Para esta investigación se seleccionó el detector de oraciones de Apache OpenNLP, el cual se caracteriza por su fácil integración en una aplicación a través de su API[28] y por poseer buena documentación (Ingersoll et al. 2013). Además, en un estudio comparativo realizado en (Read et al. 2012) esta herramienta obtuvo buenos resultados. Este estudio se realizó sobre colecciones textuales de diferentes dominios, y algunas de ellas correspondientes a textos informales generados por usuarios en la Web, lo cual es válido para la presente investigación. OpenNLP demoró 2.2 s en procesar el corpus Brown (mejor 0.3 s y peor caso 60.6 s). En el estudio realizado OpenNLP obtuvo el 97.4% de clasificación correcta como promedio en colecciones textuales de distintos dominios (mejor caso 97.6% y peor caso 95.3%). En colecciones de textos informales obtuvo 93.6% (mejor caso 95.1% y peor caso 93.26%).

El detector de oraciones de OpenNLP es de tipo supervisado, debido a que utiliza un modelo de entropía máxima para evaluar los caracteres ".", "!", y "?" en una cadena y así determinar si significan el final de la oración o no. Asume que el primer carácter que no está en blanco es el

[23] http://nlp.stanford.edu/software/corenlp.shtml
[24] http://opennlp.apache.org/
[25] http://uima.apache.org/index.html
[26] https://gate.ac.uk
[27] http://alias-i.com/lingpipe/demos/tutorial/sentences/read-me.html
[28] http://opennlp.apache.org/documentation/1.5.3/manual/opennlp.html

inicio de la primera oración y el último que no está en blanco es el final de la oración. El detector de OpenNLP[29] no requiere tokenizar el texto para detectar las oraciones.

### 2.3.2 Etapa 2 Pre-procesar el texto

Para el pre-procesamiento del texto se buscaron herramientas que permitieran realizar las técnicas de tokenización y lematización de las palabras, eliminar las palabras que no aportan información significativa del texto, llevar a minúscula las palabras que están en mayúscula y corregir palabras mal escritas. Algunas herramientas de código abierto en el lenguaje de programación Java que implementan estas funciones son Apache Lucene, TreeTagger, Stanford CoreNLP, LingPipe y Apache OpenNLP.

Para la tokenización del texto se seleccionó a Apache Lucene porque brinda más flexibilidad que las otras bibliotecas para realizar el análisis léxico de los documentos. Posee diferentes analizadores que permiten eliminar palabras vacías, convertir las palabras a minúsculas y presenta además un componente para corregir la ortografía[30]. El resto de las herramientas poseen algunas implementaciones para estas funcionalidades pero no son igual de configurables como Lucene; por ejemplo Stanford CoreNLP lematiza, pero requiere etiquetar las partes del discurso al mismo tiempo, así como tokenizar y dividir el texto en oraciones, es decir, que las funcionalidades dependen unas de otras. La herramienta TreeTagger[31] permite realizar el etiquetamiento de partes del discurso y la lematización de forma independiente. En esta etapa solo se utilizó la lematización. LingPipe posee clases para detectar errores ortográficos[32], crea un índice de Lucene para guardar las palabras escritas correctamente dado un corpus de entrenamiento, se basa en almacenar frases y mantener estadísticas sobre ellas, es sensible al dominio y hace correcciones sensibles al contexto. Lucene también brinda la posibilidad de ser sensible al dominio a partir de la especificación de un corpus para crear un índice sobre el cual se corrigen las palabras; por lo cual posee dos opciones de entrada, un diccionario o un corpus, de esta forma es más flexible. Por tanto, para el pre-procesamiento de las opiniones se propone utilizar Apache Lucene y TreeTagger.

[29] opennlp.tools.sentdetect.SentenceDetectorME

[30] Lucene suggest API se basa en indexar palabras n-gram y el cálculo de la distancia Levenshtein entre palabras. Se utilizó un diccionario de palabras, el cual es utilizado para verificar si las palabras de las opiniones se encuentran, si aparecen en el diccionario entonces son correctas, de lo contrario se sugiere el término utilizando un índice de Lucene que fue construido sobre el diccionario. De los candidatos sugeridos se escoge el primero. Se escogió el diccionario en inglés de los sistemas Unix ubicado en /usr/share/dict/words.

[31] https://github.com/reckart/tt4j

[32] lingpipe-4.1.0/demos/tutorial/querySpellChecker/read-me.html

Como en ninguna de las herramientas existe una implementación que integre las técnicas de pre-procesamiento deseadas se desarrolló un analizador léxico que hereda del analizador general de Lucene el cual pre-procesa el texto en el siguiente orden: corregir palabras mal escritas, eliminar las palabras vacías, llevar a minúscula todo el texto, tokenizar y lematizar. La entrada de esta etapa son las unidades textuales identificadas del corpus de opiniones y la salida son tokens o términos. Solo se analizarán las opiniones que poseen más de una oración, para tener información suficiente para el análisis.

### 2.3.3 Etapa 3 Representar las unidades textuales o bloques

En esta etapa se propone utilizar un modelo computacional que representará las unidades textuales por documento. Es necesario que el modelo a utilizar realice un análisis léxico del texto y/o permita representar información semántica. De las formas de representación textual existentes se seleccionó el VSM porque es un modelo que no es costoso computacionalmente como ocurre con los modelos probabilísticos, y se puede adaptar al contexto de las opiniones debido a que estas son textos cortos. Además, es un modelo que ha sido muy empleado en las tareas de segmentación y detección de tópicos. Debido a que VSM no permite representar relaciones semánticas entre las palabras, se propone experimentar también con LSA, modelo que también aplica técnicas de reducción de la dimensionalidad y utiliza matrices término-documento para la búsqueda de similitudes semánticas entre los vectores. Los modelos de semántica distribucional [33] fundamentalmente utilizan representaciones de matrices término-término y matrices término-documento (Turney 2010) para representar relaciones entre palabras como por ejemplo la coocurrencia de palabras (Sahlgren 2006). VSM y LSA no representan la coocurrencia de palabras sino la frecuencia de palabras individuales en los documentos, de ahí que pudiera ser útil el uso de otros modelos semánticos que utilicen matrices término-término. Además, se sugiere aplicar en esta etapa técnicas que contribuyen a obtener significados de las palabras como la desambiguación de su sentido[34] y la identificación de colocaciones[35] (Ingersoll et al. 2013).

Actualmente existen herramientas que incluyen implementaciones de representaciones espacio vectorial que incorporan elementos semánticos siguiendo los enfoques distribucional y

[33] Se conocen también por modelos espacio-palabra (Word Space Model).

[34] Permite seleccionar el sentido correcto de una palabra dados múltiples significados.

[35] Las colocaciones son grupos de palabras que infieren más significado juntas que separadas; es decir, el significado del todo es mayor o diferente del significado de las partes.

composicional. Para el enfoque distribucional existen dos tipos de herramientas, aquellas orientadas al conteo de vectores y las que se basan en la predicción de contextos. Las primeras se apoyan en cuatro etapas: realizar una representación del texto para extraer la cantidad de coocurrencias; utilizar un esquema de pesos para estimar dichas cantidades; reducir la dimensionalidad de las representaciones y comparar las unidades textuales a través de medidas de similitud. Las segundas representan vectores de aprendizaje de palabras usando redes neuronales; por lo que desafortunadamente requieren de un corpus de entrenamiento, de ahí que no se proponen para esta etapa[36]. Por tanto, se propone aplicar los modelos distribucionales semánticos basados en vectores de conteo y utilizar las implementaciones disponibles en las bibliotecas SemanticVectors[37] y S-Space[38].

SemanticVectors construye espacios semánticos a partir de un índice de Lucene que incluye tantos documentos como tenga el corpus. Esta representación interna constituye un inconveniente para la etapa 5 debido a que cada segmento de una opinión deberá ser observado como un documento en el espacio vectorial, y así, obtener grupos de tópicos donde cada tópico este representado por uno o más segmentos [39] . SemanticVectors utiliza un analizador específico de Lucene (StandardAnalyzer), por lo que para adicionar nuevos filtros en la etapa de pre-procesamiento es necesario modificar el código de dicha biblioteca. S-Space supera a SemanticVectors ya que presenta un marco de trabajo para el fácil desarrollo e instanciación de algoritmos para la representación textual y la creación de espacios semánticos (Jurgens & Stevens 2010). Las implementaciones en S-Space son autocontenidas; mientras que en SemanticVectors dependen de un índice de Lucene y sólo se ejecutan desde líneas de comando (Command Line Interface; CLI). S-Space permite la ejecución realizando tanto llamadas CLI como llamadas desde el API.

Así, se seleccionó a S-Space para emplearse en esta etapa, específicamente se proponen sus implementaciones de VSM y LSA. Para el descubrimiento de tópicos locales se propone la representación de las opiniones con VSM, mientras que para descubrir los globales se sugieren las representaciones VSM y LSA. Esta selección está motivada por el hecho que los tópicos locales se descubren a partir de cada opinión, y una opinión generalmente está descrita por pocos términos

[36] Aunque sería de interés emplearlas posteriormente para establecer una posible comparación de resultados en el contexto de opiniones.
[37] https://github.com/semanticvectors/semanticvectors/wiki
[38] https://github.com/fozziethebeat/S-Space/wiki/GettingStarted
[39] Un documento contiene una sola opinión.

y por tanto tiene poca dimensionalidad, mientras que al descubrir tópicos globales es necesario utilizar una representación que incluya todas las opiniones de la colección, por lo que sería útil emplear un modelo de reducción de dimensiones como LSA.

### 2.3.4 Etapa 4 Segmentar

Se refiere a emplear técnicas para encontrar límites de tópicos, algunas de ellas se basan en las ventanas deslizantes, las cadenas léxicas, la programación dinámica y los algoritmos de agrupamiento. Para los métodos de segmentación se ha destacado que la técnica de cohesión léxica es muy útil para encontrar cambios en los tópicos, utilizando fundamentalmente el cálculo de similitudes entre unidades textuales (Hernández & Pagola 2009). De esta forma, para esta etapa se definió como objetivo encontrar segmentos con alta cohesión léxica. Implementaciones recientes de algoritmos de segmentación son BayesSeg y TopicTiling, los que se basan en modelos bayesianos jerárquicos, pero realizan un entrenamiento a partir de colecciones etiquetadas previamente. Por ello, los algoritmos propuestos para experimentar son TextTiling y C99[40], los cuales hallan de manera no supervisada la cohesión léxica entre segmentos de textos a partir de una representación VSM de las oraciones identificadas y tokenizadas. Estos tienen menor costo computacional que los basados en modelos probabilísticos. La evaluación cualitativa realizada permitió concluir que TextTiling obtiene segmentos más detallados que el algoritmo C99, como se ilustra en el Anexo 10. TextTiling devuelve los segmentos del texto original, los cuales son pre-procesados con Lucene y TreeTagger para obtener los tokens por cada segmento.

### 2.3.5 Etapa 5 Representar la colección de segmentos

En esta etapa se representarán los segmentos tokenizados obtenidos de todos los documentos del corpus y se escogerá uno de los modelos mencionados en la etapa 3. En esta investigación se utilizaron las implementaciones que brinda S-Space; específicamente VSM y LSA porque el objetivo es usar matrices documento-término, que en este caso serán matrices segmento-término.

### 2.3.6 Etapa 6 Agrupar los segmentos

El agrupamiento es la forma más común de aprendizaje no supervisado. El objetivo de los algoritmos de agrupamiento es crear grupos que son coherentes internamente, es decir, que sus documentos sean lo más similares posibles y a la vez que sean disimilares o difieren de los documentos de otros grupos (Manning et al. 2008). Para detectar los tópicos es necesario emplear

[40] http://web.archive.org/web/20040810103924/http://www.cs.man.ac.uk/~mary/choif/software/C99-1.2-release.tgz

un algoritmo que permita agrupar los segmentos por tópicos, es decir, cada grupo representará un tópico. De esta forma, segmentos de la misma opinión representarán los tópicos locales que ella contiene; y segmentos de varios documentos que traten un mismo tema serán agrupados para analizar los tópicos globales obtenidos de la segmentación por cada opinión de todo el corpus.

En el contexto de la detección de tópicos se han empleado fundamentalmente tres tipos de algoritmos de agrupamiento, los algoritmos jerárquicos aglomerativos (Huang et al. 2013) (Wattenhofer & Cselle 2007), los partitivos (Seo & Sycara 2004) y los probabilísticos como Expectation-Maximization (Lu et al. 2013).

Los algoritmos de agrupamiento jerárquicos pueden ser aglomerativos (Hierarchical Agglomerative Clustering; HAC) o divisivos (Manning et al. 2008). El agrupamiento divisivo ha sido más efectivo para la segmentación de tópicos que para la detección (Purver 2011).

Dado un conjunto de *N* objetos para agrupar y una matriz de distancia o similitud el algoritmo básico de agrupamiento jerárquico consiste en (Anon n.d.):

1. Asignar a cada objeto su propio grupo (conformar *N* grupos, cada uno con un solo objeto).
2. Encontrar el par de grupos (más similar) y mezclarlos en un único grupo.
3. Repetir el paso 2 hasta que todos los objetos son agrupados en un único grupo.

Los algoritmos HAC han sido utilizados para detectar tópicos (Huang et al. 2013) (Wattenhofer & Cselle 2007) y son típicamente visualizados en un dendrograma, Entre ellos se destaca el agrupamiento de enlace único (Single-linkage), agrupamiento de enlace promedio del grupo (Average-linkage) y el agrupamiento de enlace completo (Complete-linkage). El agrupamiento de enlace único calcula la similitud entre dos grupos considerando la similitud entre sus dos objetos más similares. Considera un criterio local, es decir, solo tiene en cuenta las áreas donde dos grupos están más cerca uno del otro; las partes más distantes del grupo no se tienen en cuenta (Manning et al. 2008). El agrupamiento de enlace promedio calcula la similitud entre dos grupos como el promedio de la similitud entre los pares de objetos de un grupo y otro. Este proceso de enlace promedio es más lento que el agrupamiento de enlace único porque se necesita determinar la similitud promedio entre una gran cantidad de pares de objetos para determinar la similitud del grupo. Por otra parte, es más robusto que enlace único en cuanto a la calidad del agrupamiento (Aggarwal & Zhai 2012). Se ha recomendado como mejor algoritmo para el agrupamiento de documentos en representaciones vectoriales (Manning et al. 2008). El algoritmo de agrupamiento de enlace completo calcula la similitud de dos grupos como la similitud de sus miembros más

disimilares. No es un criterio local, porque toda la estructura del agrupamiento puede influenciar decisiones de combinación de grupos. Este criterio favorece la obtención de grupos compactos con diámetros pequeños, pero es sensible a objetos que están lejanos. Un solo objeto lejos del centro puede incrementar el diámetro del grupo y cambiar el agrupamiento final. Tiene como desventaja que es sensible a los puntos que no se ajustan en la estructura global del grupo (Manning et al. 2008).

Los algoritmos HAC construyen la jerarquía hasta obtener un solo grupo donde se incluyen todos los objetos; sin embargo, en la presente investigación se necesita obtener cierta cantidad de grupos de segmentos que representen los tópicos que se tratan en las opiniones. Por tanto, si se aplican algoritmos HAC es necesario cortar la jerarquía en algún nivel para obtener una partición. Algunas variantes para obtener una partición a partir del dendrograma son (Manning et al. 2008):

1. Cortar según un nivel pre-especificado de similitud entre objetos en un mismo grupo.
2. Cortar el dendrograma donde el espacio entre dos combinaciones de similitudes sucesivas entre objetos en un mismo grupo es mayor.
3. Estimar la suma de cuadrados como una función para una cantidad de grupos *K*.
4. Pre-especificar la cantidad de grupos *K* y seleccionar el punto de corte que produce *K* grupos.
5. Obtener todas las posibles particiones y seleccionar aquella que ofrezca la mejor calidad del agrupamiento, para ello se pueden aplicar medidas de validación internas.

Teniendo en cuenta que no resulta trivial hacer un corte en el dendrograma para obtener los grupos de segmentos, se pudiera pensar en la aplicación de un algoritmo de agrupamiento plano que obtenga directamente una partición. El algoritmo K-means (Manning et al. 2008), por ejemplo, es un algoritmo plano partitivo clásico y es más eficiente que los algoritmos jerárquicos; sin embargo, tiene varias desventajas, entre ellas: crea grupos sin una estructura explícita que los relacione, es necesario especificar la cantidad de grupos a obtener y es sensible al conjunto inicial de semillas escogidas durante el agrupamiento [49]. En esta investigación no se cuenta con conocimiento a priori que permita especificar los parámetros que requieren la mayoría de los algoritmos planos partitivos en su inicialización, de ahí que se sugiere utilizar los algoritmos HAC. Para ello sugerimos utilizar las clases implementadas para este tipo de algoritmos en la biblioteca S-Space.

La aplicación de los algoritmos HAC impone definir cuál variante utilizar para obtener una partición a partir de la jerarquía. Inicialmente se calculó el punto de corte en dependencia de la

longitud del texto de cada opinión. Por ejemplo, para opiniones cuya matriz de representación presenta una dimensión entre 50 y 100 filas se utilizó un corte igual al 90% de las combinaciones de todo el árbol. Sin embargo, para lograr un punto de corte estándar para una variabilidad en la longitud de las opiniones, este enfoque es más difícil que aplicar un umbral que permita agrupar comparando las medidas de similitud de los grupos con dicho umbral; es decir, los grupos serán agrupados hasta que la mayor similitud de un grupo sea menor que el umbral especificado, si es igual o mayor se detiene el agrupamiento. Para realizar este enfoque se propone el uso de cuatro expresiones para calcular los umbrales (Shulcloper 2010): la media de las similitudes entre todos los pares de objetos posibles, la media de los valores máximos de las similitudes entre cualquier par de objetos, la media de los valores mínimos de las similitudes entre cualquier par de objetos y la media ponderada de la media de las similitudes y la media de los máximos (Arco 2008). Se seleccionó la medida coseno para hallar la matriz de similitud en el algoritmo de agrupamiento.

Teniendo en cuenta que algunos segmentos pudieran pertenecer a más de un grupo porque expresen opiniones de más de un tópico, se propone la aplicación de métodos de agrupamiento que generen cubrimientos. El algoritmo Estrella (Aslam, J.; Pelekhov, K. and Rus 1998) y sus variantes (Gil et al. 2003) (Gago et al. 2007) (Pérez & Medina. 2007) son ejemplos de algoritmos de agrupamiento con solapamiento que pudieran utilizarse para obtener cubrimientos del universo de segmentos. Estos algoritmos son basados en grafos, cada objeto a agrupar es un nodo del grafo y se establecen aristas entre los objetos ponderadas con los valores de similitud entre ellos. Estos algoritmos requieren la especificación de un umbral de similitud y realizan un cubrimiento goloso del grafo de similitud por medio de subgrafos que presentan forma de estrella. Cada estrella está determinada completamente por el centro de la estrella y los satélites que se encuentran en la lista de adyacencia del vértice central.

El algoritmo Estrella tiene algunas desventajas que limitan su aplicación, ya que es sensible al orden en que se presentan los objetos a agrupar y puede producir grupos ilógicos (Gil et al. 2003) (Gago et al. 2007). Por las razones anteriores varias han sido las extensiones propuestas del algoritmo Estrella, por ejemplo el algoritmo Estrella Extendido (Extended Star; ES) (Gil et al. 2003), Estrella Condensando (Generalized Star; GStar) (Condensed Star; ACONS) (Gago et al. 2007) y Estrella Generalizado (Pérez & Medina. 2007). ES es independiente del orden de los datos y obtiene un menor número de grupos respecto al algoritmo Estrella. Los algoritmos GStar y ACONS introducen nuevos conceptos de estrella y obtienen un menor número de grupos (Arco

2008). De forma general estos algoritmos están caracterizados por generar cubrimientos sobre los datos y no requieren especificar el número de grupos a obtener, de ahí que se ha decidido aplicar las variantes ES, GStar y ACONS en esta etapa para el agrupamiento de segmentos y consecuentemente descubrir los tópicos. No obstante, es importante tener en cuenta que estos algoritmos son sensibles al umbral de similitud fijado inicialmente y tienen alto costo computacional.

Para la aplicación de los algoritmos ES, GStar y ACONS en esta etapa se utilizaron las implementaciones desarrolladas por investigadores del Centro de Estudios Informáticos de la Universidad Central "Marta Abreu" de Las Villas (CEI-UCLV) en el contexto de la desambiguación del sentido de las palabras (Pérez & González 2014) e incluidas en la biblioteca RST-Disambiguation. El umbral de similitud se estimó considerando las cuatro variantes que se utilizaron para calcular el punto de corte en HAC y que se encuentran publicadas en (Shulcloper 2010).

### 2.3.7 Etapa 7 Etiquetar los grupos de segmentos

El etiquetamiento de tópicos (topic labeling) se refiere a otorgarle a cada grupo de segmentos una breve descripción que identifique el tópico que tratan. Una etiqueta puede estar constituida por una o varias palabras que mejor representen un tópico. Generalmente el etiquetamiento de tópicos es visto como una tarea de clasificación supervisada (Carenini & Ng 2013). Así, si se tiene un corpus previamente etiquetado por tópicos, se puede entrenar un clasificador capaz de predecir el tópico de un nuevo segmento. No obstante, existen métodos no supervisados que obtienen los grupos de tópicos al agrupar los segmentos, aplican una distancia para determinar los términos con mayores puntuaciones y estos conforman la etiqueta del grupo (Xu & Oard 2011). Algunos enfoques de etiquetamiento de tópicos ordenan las palabras más relevantes de un segmento de tópico y con ellas construyen las etiquetas; por ejemplo, se enfocan en obtener descripciones de las relaciones jerárquicas entre los tópicos, apoyándose en la extracción de frases significativas utilizando técnicas de análisis de secuencias de textos y generación de n-grams (Mao 2012) (Mei et al. 2007). El método de LDA puede ser usado también para la tarea de etiquetamiento de tópicos (Carenini et al. 2011) (Hingmire 2013). Una propuesta interesante es un método no supervisado basado en un grafo y que emplea el algoritmo PageRank para pesar las palabras (Aletras & Stevenson 2014). La puntuación final de las etiquetas candidatas es la suma de los pesos de sus palabras. La etiqueta con la mayor puntuación se selecciona para representar el tópico.

Seleccionar etiquetas o tópicos es más difícil que escoger objetos representativos de un grupo. La mayor parte de los enfoques se dirigen a encontrar términos importantes y frases en el grupo. Uno de los métodos propuestos por investigadores consiste en otorgar pesos a los términos; por ejemplo, utilizando TF-IDF y devolver una lista de términos ordenados por pesos (Tagarelli & Karypis 2012). También se han usado n-grams para devolver una lista de frases basadas en sus pesos en la colección, aunque generalmente no producen etiquetas de alta calidad (Mei et al. 2007). Otro enfoque es usar log-likehood ratio de los términos que están en el grupo versus aquellos que están fuera del grupo también ha sido un enfoque utilizado con éxito (Ingersoll et al. 2013).

El etiquetamiento de grupos de documentos[41] se enfoca en dos variantes (Manning et al. 2008): el etiquetamiento diferencial y el etiquetamiento interno. En el primero se utilizan métodos de selección de rasgos para seleccionar las etiquetas al comparar las distribuciones de términos de un grupo respecto a otros grupos. Algunos de estos métodos se basan en la información mutua, la ganancia de la información y el cálculo del test chi cuadrado. El etiquetamiento interno de los grupos calcula una etiqueta que solo depende de un grupo, no de otros grupos. Algunas propuestas etiquetan un grupo con el título de un documento cercano al centro. Otras usan una lista de términos con pesos altos en el vector centro del grupo. La desventaja de los métodos internos es que no tienen en cuenta cuánto discriminan entre grupos las etiquetas identificadas.

No todos los métodos de etiquetamiento de grupos obtienen etiquetas que constituyen los tópicos que aborda el grupo; por ejemplo, aquellos métodos que identifican el documento más cercano al centro del grupo efectivamente lo caracterizan pero no representa un tópico. Esto sucede con varios algoritmos de etiquetamiento; de ahí que detectar tópicos en grupos de segmentos impone retos mayores al etiquetamiento de grupos.

Si en esta etapa se desean identificar etiquetas que coincidan con los tópicos que se abordan en los grupos y éstos a su vez se correspondan en alguna medida con las entidades y los aspectos que en el futuro análisis de sentimientos se consideren, entonces es necesario explorar herramientas y técnicas utilizadas en la detección de polaridad basada en aspectos para obtener así etiquetas que mejor caractericen a los grupos y contribuyan a obtener resultados satisfactorios en las etapas de la minería de opinión que se desarrollarán tomando como punto de partida los resultados de esta investigación. Por tanto, resulta útil explorar el enfoque que permite obtener palabras o frases representativas de un texto utilizando bases de conocimiento.

[41] En esta etapa es necesario etiquetar grupos de segmentos

WordNet[42] es una base de datos léxica disponible para el idioma inglés. Los sustantivos, verbos, adjetivos y adverbios se agrupan en conjuntos de sinónimos, los cuales expresan conceptos distintos. Utiliza semántica conceptual y relaciones léxicas para calcular las relaciones entre palabras. WordNet facilita el cálculo de varias medidas de similitud semántica entre palabras, y estas han sido utilizadas para hallar la similitud entre aspectos en el contexto de la minería de opinión (Pavlopoulos 2014). La biblioteca WS4J[43] permite el trabajo con WordNet e incluye varios algoritmos para hallar la similitud semántica entre palabras.

Teniendo en cuenta las características de los métodos de etiquetamiento publicados, las técnicas de análisis de sentimiento basado en la extracción de aspectos y las potencialidades de las bases de conocimiento, como WordNet, se propone, como resultado de esta investigación, un método para etiquetar los grupos de segmentos de forma no supervisada y por términos, basado en la extracción de sustantivos que permite obtener mejores descripciones de tópicos. Este método se aplica a cada grupo y asume que se le da como entrada los términos que conforman el grupo.

**Algoritmo para la detección de tópicos basado en el etiquetamiento de grupos mediante la extracción de sustantivos**

Entrada: conjunto de términos $T=\{t_1, \ldots, t_n\}$, tamaño $k$ de la etiqueta y la similitud a utilizar entre sustantivos.

Salida: tópicos que se abordan en el grupo, es decir, la etiqueta con los sustantivos más representativos del grupo.

1. Identificar sustantivos en el grupo.
2. Para cada sustantivo:
   2.1 Calcular la similitud que tiene con el resto de los sustantivos en el grupo.
   2.2 Calcular la puntuación del sustantivo a partir de la suma de todos los valores obtenidos de la similitud con cada sustantivo del grupo.
3. Seleccionar el sustantivo de mayor puntuación.
4. Conformar la etiqueta del tópico con los $k$ sustantivos más similares al sustantivo seleccionado.

Las herramientas TreeTagger y OpenNLP permiten obtener los términos clasificados como sustantivos. Para implementar el algoritmo propuesto se seleccionó la herramienta OpenNLP

[42] http://wordnet.princeton.edu/
[43] https://code.google.com/p/ws4j/

debido a que con TreeTagger se obtuvieron en algunas ocasiones clasificaciones de sustantivos menos adecuadas; por ejemplo términos como "play" y "make" son generalmente más utilizados como verbos que como sustantivos y OpenNLP fue capaz de reconocerlos como verbos, mientras que TreeTagger los observó como sustantivos. Se propone además el uso de WS4J para obtener las palabras más representativas de los grupos de segmentos de acuerdo a la similitud semántica que poseen entre ellas. WS4J incluye varias medidas para hallar la similitud semántica entre palabras (Resnik 1995). En el Anexo 11 se describen dichas medidas. En el epígrafe 3.4 se mostrará la evaluación empírica realizada para validar esta etapa y de la cual se concluye que la medida de similitud semántica Resnik se acerca más a los sustantivos señalados por tópicos para representar los grupos de segmentos y por tanto permite obtener los mejores resultados en el etiquetamiento.

## 2.4 Conclusiones parciales del capítulo

El análisis de sentimiento es un área de investigación que se le ha prestado gran interés actualmente, específicamente al análisis de sentimiento por aspectos. El nivel de aspecto se define a partir de los atributos de las entidades por lo que es el nivel que brinda mayor información, respecto al análisis a nivel de oración o de documento.

Los métodos de segmentación y detección de tópicos han sido aplicados al análisis de sentimientos, facilitando la detección de la polaridad de las opiniones por tópicos. Sin embargo, aún existen herramientas que no garantizan un análisis de sentimiento por tópicos, como es el caso de la aplicación PosNeg Opinion, que a pesar de exhibir excelentes valores de precisión y exactitud en la detección de la polaridad de las opiniones solo es capaz de clasificar las opiniones expresadas en una oración o la opinión que expresa el texto en su totalidad, y por tanto no realiza un análisis del sentimiento por tópicos.

El esquema general que se propuso como resultado de esta investigación parte de una opinión o conjunto de opiniones y permite obtener tópicos locales y globales correspondientes a los grupos de segmentos que abordan un mismo tema. Para ello, es necesario transitar por siete etapas, donde las formas de representación siguen los modelos VSM y LSA que permiten incorporar elementos léxicos y semánticos, es posible obtener particiones y cubrimientos del universo de segmentos con el objetivo de formar grupos por tópicos y finalmente se etiquetan los grupos obtenidos a partir de la selección de los sustantivos más representativos por tópicos.

Se diseñó e implementó un método que permite etiquetar de manera efectiva los grupos de segmentos de forma no supervisada y por términos. El cálculo de las similitudes semánticas entre los sustantivos utilizando la medida Resnik permite obtener los mejores resultados en el etiquetamiento, ya que se acerca más a los sustantivos señalados por tópicos para representar los grupos de segmentos. Esta forma de etiquetamiento siguiendo las ideas del análisis de sentimiento basado en aspectos permite obtener mejores descripciones de los tópicos.

La implementación de este esquema se puede llevar a cabo satisfactoriamente utilizando las siguientes tecnologías: en la Etapa 1 Apache OpenNLP; en la segunda etapa Apache Lucene y TreeTagger; S-Space en las etapas 3 y 5, en la cuarta etapa TextTiling y en la Etapa 7 Apache OpenNLP y WS4J.

# Capítulo 3 Descubrimiento de tópicos en opiniones y su evaluación

En el presente capítulo se expondrán los resultados obtenidos en la evaluación de la propuesta presentada en esta investigación para descubrir tópicos en textos de opiniones. Para ello, primeramente se realizará una descripción del diseño e implementación del marco de trabajo OpinionTopicDetection así como la aplicación OpinionTD propuestos para el descubrimiento de tópicos en opiniones. Se presenta una descripción de las colecciones de opiniones disponibles. Luego, se describirá el procedimiento seguido para determinar el método de agrupamiento a aplicar para realizar la evaluación. Además, se mostrará la evaluación empírica realizada para seleccionar la medida de similitud semántica entre sustantivos a utilizar la etapa 7 de etiquetamiento. Y por último, se expondrán los resultados de la evaluación realizada de la detección de tópicos sobre un corpus de opinión.

## 3.1 Marco de trabajo OpinionTopicDetection

En el contexto del procesamiento del lenguaje natural existe una gran diversidad de herramientas con funcionalidades que contribuyen al descubrimiento de tópicos, por ejemplo: Apache Lucene, StanfordCoreNLP, NLTK, LingPipe, Apache OpenNLP, TreeTagger, S-Space, Apache Mahout, etc. Debido a que no se encuentra disponible un marco de trabajo (framework) en el contexto de opiniones que integre las funcionalidades que brindan algunas de estas herramientas de acuerdo al esquema propuesto en el Capítulo 2, se diseñó e implementó OpinionTopicDetection como un marco de trabajo para el descubrimiento de tópicos en opiniones.

### 3.1.1 Método de desarrollo

El desarrollo de un software puede ser guiado por el empleo de diversas metodologías y métodos, los cuales se pueden observar desde dos enfoques fundamentales: prescriptivos y adaptativos. Un enfoque prescriptivo es el Proceso Unificado de Rational (Rational Unified Process; RUP) el cual está centrado fundamentalmente en la documentación, la planificación y los procesos; se caracteriza además por guiarse por casos de uso, ser iterativo e incremental (Jacobson 2000). Es considerado muy restrictivo, porque presenta más de 30 perfiles, más de 20 actividades, y más de 70 artefactos a lo largo de cinco flujos de trabajo (requisitos, análisis, diseño, implementación y prueba) (Kniberg 2009). En el enfoque adaptativo se destacan las metodologías y métodos ágiles[44]

[44] Están guiadas por los principios declarados en el Manifiesto Ágil, http://www.agilemanifesto.org

las cuales han surgido con el objetivo de disminuir tales cantidades de restricciones al construir un software, y se proponen alcanzar un enfoque donde se realice un desarrollo incremental con iteraciones muy cortas. De esta forma se les da mayor valor al individuo, a la colaboración con el cliente y al producto de software (José H. Canós n.d.). Ejemplos de metodologías ágiles son la Programación Extrema (Extreme Programming; XP) y Iconix (Rosenberg & Stephens 2007); y ejemplos de métodos de desarrollo son Scrum y Kanban (Kniberg 2009). Algunos autores han destacado la combinación de varios métodos y metodologías como una variante efectiva para adaptarse a distintos entornos de trabajo; por ejemplo Scrum y XP (Kniberg et al. 2007) o Scrum y Kanban[45] (Kniberg 2009).

De acuerdo al objetivo del presente trabajo se seleccionaron elementos de Scrum y Kanban de forma conjunta para guiar el desarrollo de la propuesta de forma incremental y adaptativa. La selección estuvo motivada porque estos modelos adaptativos siguen principios de desarrollo ágiles y lean[46], por lo que permiten disponer en el menor tiempo posible de un producto o servicio de valor para el cliente, y mantener éste en evolución continua para aumentar sus funcionalidades. Scrum es un marco de trabajo de administración para el desarrollo de productos de forma incremental que usa equipos de trabajo multifuncionales y emplea iteraciones de tamaño fijo llamadas sprints, típicamente persisten de dos a cuatro semanas. Como artefacto principal utiliza una lista de productos o funcionalidades sobre las que realiza los sprints (James 2015) (Kniberg 2009). Kanban es una aproximación a la gestión del cambio. No es un proceso de desarrollo de software o de gestión de proyectos, es una aproximación a la introducción de cambios en un ciclo de vida de desarrollo de software o metodología de gestión de proyectos ya existente (Kniberg 2009). Está definido por un conjunto de principios claves, ellos son: visualizar el flujo de trabajo, limitar el trabajo en progreso, medir y manejar el flujo, declarar medidas de procesos de forma explícita y usar modelos para evaluar la mejora de oportunidades (Anderson & Linden-reed 2015). En el Anexo 12 se exponen las características tomadas de ambos métodos para el desarrollo de esta investigación.

De forma general se siguieron cinco fases: definir la pila de funcionalidades; planificar los sprints; desarrollar los sprints; evaluar cada sprint e integrar la funcionalidad al esquema. En el Anexo 13

[45] http://leansoftwareengineering.com/ksse/scrum-ban/

[46] Término empleado para referirse al Sistema de Producción Toyota (TPS), método de manufactura empleado por compañías Japonesas. Fue adaptado en el desarrollo de software para establecer principios para el desarrollo ágil.

se muestra la pila de funcionalidades elaborada. Para la planificación de cada iteración se consideró cada etapa del esquema propuesto. En el Anexo 14 se muestra un tablero ejemplo con las tareas correspondientes al desarrollo de un sprint. Luego de culminadas las tareas de cada sprint se realizaron pruebas unitarias al nuevo código para comprobar la correcta funcionalidad de las tareas implementadas y se realizó su integración con el marco de trabajo propuesto.

### 3.1.2 Implementación

Los marcos de trabajo contienen conjuntos de clases[47] guiadas por un diseño abstracto que definen un comportamiento. Son mayormente aplicables para solucionar problemas de un dominio donde existen requisitos o funcionalidades comunes. Con el objetivo de usarlos se necesita insertar el comportamiento deseado para una aplicación en varios lugares, ya sea heredando de las clases del marco de trabajo o incorporando las clases en una aplicación; el código del marco de trabajo llama al código de la aplicación en dichos lugares (Fowler 2005). Dos de las características principales[48] de los marcos de trabajo son la inversión de control y la extensibilidad. En la primera, el flujo de control general del programa es llamado por el marco de trabajo, no por la aplicación del usuario. La segunda, se refiere a que puede ser extendido por el usuario; por ejemplo, al sobreescribir el código para proveer una funcionalidad específica.

Siguiendo estas ideas se implementó el marco de trabajo OpinionTopicDetection, el cual realiza una adaptación en cinco etapas del esquema general propuesto para descubrir tópicos en opiniones. El flujo de estas etapas implementadas se muestra en la Fig. 2.

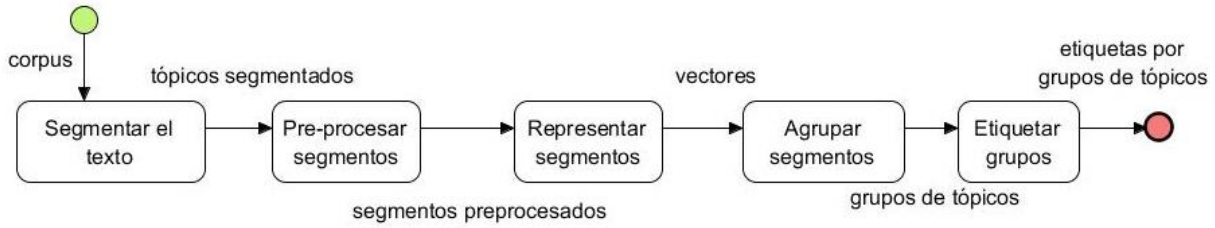



**Fig. 2 Flujo de las etapas implementación en OpinionTopicDetection.**

Estas etapas están implementadas en la clase abstracta `uclv.ai.otd.SchemeTemplate`, la cual posee cinco métodos: `segmentText`, `analyzeSegments`, `createSemanticSpace`, `createClusters`, `createTopicLabels`; sus funcionalidades están dirigidas a obtener segmentos

[47] En la programación orientada a objeto.

[48] Estas características los distinguen de otros paquetes de software como las bibliotecas (libraries); las cuales son esencialmente un conjunto de funciones que se pueden llamar, cada llamada ejecuta una función y devuelve el control al cliente.

del texto, realizar un pre-procesamiento textual, utilizar modelos de representación textual, aplicar algoritmos de agrupamiento para obtener grupos de tópicos y etiquetar los grupos, respectivamente. Los métodos de representación textual, agrupamiento y etiquetamiento son abstractos por lo que pueden ser implementados en una aplicación que utilice este marco de trabajo. También los métodos de segmentación y pre-procesamiento pueden ser sobreescritos en el caso que se desee incorporar nuevos algoritmos de segmentación o aplicar técnicas de pre-procesamiento que no se hayan tenido en cuenta en el marco de trabajo. En el desarrollo de esta propuesta se siguieron los patrones de diseño de Plantilla (Template), Composición (Composite) y Fachada (Facade). En la Fig. 3 se muestra la arquitectura por capas de OpinionTopicDetection.

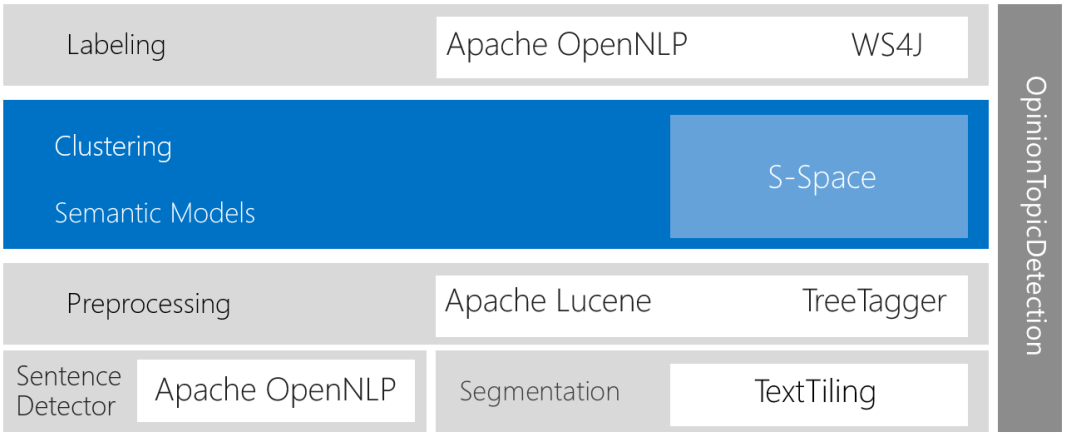



**Fig. 3 Arquitectura de la propuesta.**

En Anexo 15 se muestra el diagrama de paquetes presentes en el marco de trabajo propuesto. Los paquetes tienen dependencias entre ellos. A continuación se describen cada uno de ellos:

- `otd`: paquete que contiene las clases principales para detectar tópicos locales y globales. En el

- Anexo ***16*** se muestran las clases principales que contiene este paquete. Las clases `uclv.ai.otd.LocalTopicImpl` y `uclv.ai.otd.GlobalTopicImpl` corresponden a las implementaciones de `uclv.ai.otd.SchemeTemplate` para descubrir tópicos locales y globales respectivamente. La clase `uclv.ai.otd.OpinionTopicDetection` es la clase principal para usar el esquema propuesto, la cual contiene el método `detectTopics` para obtener los tópicos. Este paquete depende del resto de los paquetes del marco de trabajo.
- `preprocess`: posee las clases que permiten realizar análisis léxico con Apache Lucene en la etapa de pre-procesamiento, posee también una instanciación de TreeTagger para obtener los lemas y partes del discurso de las palabras.
- `segmentation`: posee una clase fachada para instanciar el algoritmo de segmentación C99 y otra para TextTiling. Requiere el uso del paquete `sentencedetectors`.
- `sentencedetectors`: contiene las clases que implementan métodos para detectar oraciones con Apache OpenNLP y Stanford CoreNLP.
- `semanticmodels`: posee una clase fachada que encapsula los principales métodos para trabajar con las implementaciones de los modelos VSM y LSA, presentes en el API de S-Space. Requiere el uso del paquete `segmentation`.
- `clustering`: posee una clase fachada que encapsula métodos para instanciar el algoritmo HAC de S-Space, así como implementaciones de algoritmos Estrella. Requiere el uso del paquete `semanticmodels` y `util`.
- `labeling`: posee las clases que realizan el etiquetamiento de los grupos de tópicos obtenidos de un algoritmo de agrupamiento. Requiere el uso del paquete `clustering` y `util`.
- `evaluation`: posee las clases utilizadas para la evaluación de la propuesta. Requiere el uso del paquete `labeling`.
- `util`: posee clases utilitarias que se utilizan en las etapas del esquema de propuesto.

Para obtener tópicos con OpinionTopicDetection siguiendo el esquema propuesto solo es necesario instanciar la clase `uclv.ai.otd.OpinionTopicDetection` y seleccionar los parámetros correspondientes a los algoritmos que se desean utilizar. En la Fig. 4 se muestra un fragmento de código de cómo realizar dicha instanciación. El código mostrado en Fig. 4 a) corresponde a la detección de tópicos locales y el código mostrado en Fig. 4 b) a la detección de tópicos globales.

a)

```
OpinionTopicDetection op = new OpinionTopicDetection(new LocalTopicImpl());
List<TopicData> topics = op.detectTopics(corpus, Algorithm.SEGMENTATION_TextTiling,
        Algorithm.REPRESENTATION_VSM, Algorithm.CLUSTERING_HAC_AVERAGE_LINKAGE, Algorithm.THRESHOLD_MEAN);
```

b)

```
OpinionTopicDetection op = new OpinionTopicDetection(new GlobalTopicImpl());
List<TopicData> topics = op.detectTopics(corpus, Algorithm.SEGMENTATION_TextTiling,
        Algorithm.REPRESENTATION_VSM, Algorithm.CLUSTERING_HAC_AVERAGE_LINKAGE, Algorithm.THRESHOLD_MEAN);
```

**Fig. 4 Fragmento de código para instanciar el marco de trabajo.**

Una vez culminada la implementación del marco de trabajo se desarrolló una aplicación de escritorio, llamada OpinionTD que utiliza la instanciación presentada en la Fig. 4. Su entrada es la ubicación del corpus a procesar y los parámetros a especificar son: el tipo de tópico que se quiere obtener (local o global); un algoritmo de segmentación; un algoritmo de representación textual; un algoritmo de agrupamiento y el tipo de umbral para obtener las particiones de grupos. La salida está constituida por los tópicos presentes por cada opinión del corpus para el caso de los tópicos locales; y para los tópicos globales la salida mostrará los tópicos encontrados en todo el corpus. En el Anexo 17 y el Anexo 18 se muestra la interfaz de OpinionTD.

OpinionTopicDetection puede ser usado además como una biblioteca por otras aplicaciones para el análisis textual, debido a que las funcionalidades que reutiliza de otras bibliotecas se pueden instanciar de forma independiente. Por ejemplo: se pueden emplear los métodos implementados para detectar oraciones con las herramientas Apache OpenNLP y Stanford CoreNLP a través de las clases `uclv.ai.otd.sentencedetectors.DetectSentencesOpenNLP` y `uclv.ai.otd.sentencedetectors.DetectSentencesStanford`. Se pueden realizar segmentaciones del texto con los algoritmos TextTiling y C99 empleando las clases `uclv.ai.otd.segmentation.C99Facade` y `uclv.ai.otd.segmentation.TextTilingFacade`. Provee clases para el pre-procesamiento textual empleando Lucene mediante las clases disponibles en el paquete `preprocess`. Proporciona también la clase `uclv.ai.otd.semanticmodels.SSpaceFacade`, para acceder a los métodos que permiten realizar representaciones textuales con la biblioteca S-Space. Así como utilizar algoritmos de agrupamiento jerárquico y extensiones del algoritmo Estrella presentes en la clase `uclv.ai.otd.clustering.ClusterFacade`.

## 3.2 Descripción de colecciones de opiniones textuales

La mayoría de las colecciones textuales utilizadas en los artículos descritos en el epígrafe 2.2 no están disponibles, lo que limita significativamente la comparación de resultados obtenidos por una u otra propuesta. Existen colecciones disponibles que permiten validar los resultados (Pang & Lee 2008); sin embargo, la mayor parte de ellas no se encuentran etiquetadas por tópicos y por tanto es difícil disponer de una clasificación de referencia. A continuación se describen algunas colecciones de opinión que se encuentran disponibles para la investigación en el campo de la minería de opinión. En ellas pueden aparecer opiniones largas, medianas y cortas las cuales se encuentran en diversos dominios como restaurantes, hoteles, productos, y otros en idioma inglés. Las definiciones de opiniones pequeñas, medianas y largas se consideraron en esta tesis de manera empírica a partir de la observación de las opiniones presentes en las colecciones textuales. Se considera opinión pequeña aquella que tenga hasta 5 oraciones. Se considera opinión mediana aquella que tiene desde 6 a 15 oraciones. Se consideran opiniones largas aquellas que tienen más de 16 oraciones. Otro posible criterio para clasificar las opiniones según su tamaño pudiera ser a partir del conteo de las palabras que las conforman, sobre todo teniendo en cuenta que las oraciones pueden tener más o menos palabras incluidas. Se decidió realizar la clasificación considerando la cantidad de oraciones ya que las oraciones son las unidades textuales que consideran los algoritmos de segmentación empleados.

- Deceptive Opinion Spam Corpus v1.4: es un corpus creado en el año 2011 para investigar la creación de opiniones falsas (opinion spam). Contiene opiniones positivas y negativas dadas en el sitio de TripAdvisor[49] sobre 20 hoteles de la ciudad de Chicago, las cuales fueron clasificadas a su vez en verdaderas y falsas. Posee un total de 1600 opiniones, y están en formato .txt (Ott et al. 2011). En este corpus predominan las opiniones medianas. http://www.cs.cornell. edu/~myleott/op_spam
- LARA: corpus creado en el año 2010 para analizar evaluaciones de aspectos (aspect rating). Posee 1850 opiniones de hoteles en Tripadvisor en formato .txt y .json. Las opiniones tienen identificados los términos de aspectos en ocho categorías: generalidades, valor, habitaciones, ubicación, limpieza, registro, servicio y servicio de negocios. En este corpus predominan las opiniones largas (Wang et al. 2010). Para dicha investigación también se creó otro corpus con

[49] http://www.tripadvisor.es

opiniones de productos del sitio de Amazon, donde se presentan 7663 opiniones en 6 categorías: cámara, teléfono móvil, TV, laptop, Tablet y sistemas de vigilancia por video. http://times.cs.uiuc.edu/~wang296/Data/

- TripAdvisor Annotated Dataset: corpus creado en el año 2014 en el contexto de la minería de opinión basada en aspectos para responder al problema de predecir para todas las oraciones de una opinión su polaridad sobre un aspecto del producto. Presenta 369 opiniones de hoteles de Tripadvisor en formato .json (Marcheggiani & Oscar 2014). Este corpus se caracteriza por presentar 11 aspectos (habitaciones, limpieza, valor, servicio, ubicación, registro, negocios, comida, edificación, otros, no relacionados) anotados por oración y predominan las opiniones medianas y pequeñas. Fue anotado manualmente y distingue opiniones positivas, negativas, neutrales o si no se expresa ninguna opinión. http://nemis.isti.cnr.it/~marcheggiani/datasets/aspects-annotated-dataset.tar.gz
- The SFU Review Corpus: corpus creado en el año 2004 con 400 opiniones del sitio Epinions[50] en formato .txt. Posee 50 opiniones por cada categoría, 25 positivas y 25 negativas. Las opiniones pueden estar relacionadas con libros, carros, computadoras, utensilios de cocina, hoteles, películas, música y teléfonos. Predominan las opiniones largas. http://www.sfu.ca/~mtaboada/download/downloadCorpus.html
- JDPASentimentCorpus: corpus creado en el año 2010, está enfocado en el dominio automotor y cámaras digitales. Posee 515 opiniones en formato .txt y predominan las opiniones largas y medianas. http://verbs.colorado.edu/jdpacorpus/
- Yelp academic dataset: corpus creado en el 2014, el cual contiene opiniones de restaurantes y servicios publicados en el sitio Yelp[51]. Se encuentra en formato .json y predominan las opiniones medianas y pequeñas. https://www.yelp.com/academic_dataset
- Large Movie Review Dataset v1.0: corpus creado en el 2011 con opiniones de películas, tiene 50000 opiniones, clasificadas en positivas y negativas en formato .txt y la mayor parte están caracterizadas por ser largas y medianas. http://ai.stanford.edu/%7Eamaas/data/sentiment/aclImdb_v1.tar.gz

[50] http://www.epinions.com
[51] http://www.yelp.com

## 3.3 Determinación del método de agrupamiento a aplicar y sus parámetros

El esquema concebido para la segmentación y detección de tópicos en opiniones se puede ejecutar con una opinión o con un corpus de opiniones. Por tanto, la etapa 6 de agrupamiento se enfrenta a dos situaciones diferentes en función de la entrada. Las opiniones generalmente están formadas por una pequeña cantidad de segmentos, y agrupar tales segmentos permite obtener tópicos locales. Esto impone un reto al método de agrupamiento ya que tiene pocos objetos a agrupar. Cuando se ejecuta el esquema con un corpus de opiniones, existen múltiples segmentos a agrupar, y por tanto, las condiciones a las que se enfrenta el método de agrupamiento a aplicar son otras. De ahí que se dividió el análisis de los métodos de agrupamiento a partir del diseño de experimentos considerando la obtención de tópicos locales y globales por separado.

### 3.3.1 Análisis del agrupamiento de tópicos locales

Los tópicos locales se forman a partir del agrupamiento de los segmentos identificados en una opinión. Las opiniones pueden tener diversos tamaños y en función de sus tamaños así será mayor o menor la cantidad de segmentos que las conforman y por tanto esto influye en el agrupamiento a realizar.

**Tabla 1 Ejemplos de opiniones para las tres clasificaciones establecidas según su tamaño (A-Pequeñas, B-Medianas, C-Grandes).**

| | |
|---|---|
| *A* | *This was a great place to be! Great views of river and lake, walk to everything, a clean and comfortable room, and very accommodating staff. We arrived early and were checked in by 10:30 am and checked out late , effectively adding two days to our vacation. It is a large hotel but the staff works hard and takes good care of the customers.* |
| *B* | *We stayed here from Nov. 30 to Dec 2 and had a wonderful time. The hotel is just beautiful and the service was excellent from check in to the maid staff to the bartenders in Kitty O'Shea's. We had a room with a king bed that was very comfortable and had very nice feather pillows. You can request other types if you have a problem with feathers. The large flat screen TV was very nice. Bath products were by Crabtree & Evelyn and included shampoo, conditioner, mouthwash and bady lotion. Plenty of coffee was provided. We drank and had snacks at Kitty O'Shea's. The crowd was fun and there was live Irish folk music each night. Good selection of beer and good Shepards Pie, Cheese Dip and Crisps! The location worked very well for us. We walked to the Art Museum and Buddy Guys Blues Club is right across the street. And shopping was a breeze using the shuttle! The cab ride to the Museum of Science and Industry was quite far though and cost about $15 each way. An excellent weekend getaway!* |
| *C* | *We have just returned from a week at the James Chicago. The hotel is fabulous, very chic and chilled. The service is beyond compare. Check in and out were swift and painless. The Concierge service is also super efficient having located an excellent hair salon and a couple of fab restaurants for us during our stay. The hotel restaurant itself is rated 5 in the top 10 in Chicago - voted by Chicago residents. We stayed on the 14th floor with great views of the city. The room (a deluxe king) was equipped with a great selection of cocktail making lovelies, hairdryer, safe, tv, robes & slippers, a very interesting selection of minibar items (one of which was a real surprise!) including crisps/snacks a first aid kit and an umbrella! We also had a spotless and elegant bathroom with Kiehls products and a cosy little dining area. The ipod dock/CD/radio and complimentary chillout CD were a nice touch and husband liked the enormous plasma tv. The super comfy bed ensured a good snooze and complimentary WiFi allowed us to plan our next day's activities from our* |

| | |
|---|---|
| | *bedroom. We were so busy touring around the city we didn't get an opportunity to use the gym or the spa - both of which looked fab. The bar/lobby area is a haven of civilised tranquility with smooth music playing in the background to help you chillout that bit more. Staff are attentive without being fussy and a good selection of drinks and bar snacks are available. The barman mixes a mean mohito ! This hotel is full of nice little thoughtful touches to make people feel welcomed and cared for e.g. The Virtual Tuck In service - basically a couple of PCs with webcams to allow people to say goodnight to their children when they're away from home on business. Also the dog biscuits and water outside the front door for our furry friends. Next visit to Chicago I'll definitely be staying here and wouldn't even consider an alternative if The James can maintain these standards.* |

Primeramente se realizó un análisis cualitativo inicial donde se exploró el comportamiento de los algoritmos de agrupamiento para opiniones de diferentes tamaños; para ello se seleccionaron 120 opiniones del corpus LARA referente a hoteles. Por ejemplo, para la opinión (A) mostrada en Tabla 1 que es considerada una opinión pequeña, al aplicarle el algoritmo HAC-Complete con un umbral medio, no se lograron identificar todos los tópicos presentes. Sin embargo, para una opinión mediana (B) se obtuvieron mejores resultados; para el caso de la opinión larga (C) se obtuvieron algunos tópicos mezclados en un mismo grupo. Un factor que influye en este último resultado es que predominan oraciones en las que se opina sobre diferentes tópicos; por ejemplo, en la siguiente oración de la opinión (C) se habla tanto de la cama como del servicio Wifi: "*The super comfy bed ensured a good snooze and complimentary WiFi allowed us to plan our next day's activities from our bedroom*" y en la opinión (A) se observa un texto más diverso: "*Great views of river and lake, walk to everything, a clean and comfortable room, and very accommodating staff."*; esto es un problema que se ha abordado a través de métodos de aprendizaje supervisado (Marcheggiani & Oscar 2014). En el Anexo 19 se muestran los tópicos encontrados para estas opiniones.

De ahí que el reto consiste en identificar el algoritmo de agrupamiento que mejor se comporte para los distintos tamaños de opinión, ya que en una aplicación práctica real las colecciones de opiniones presentan gran variabilidad de los tamaños de opinión.

**Tabla 2 Características de las colecciones creadas para tópicos locales.**

| Corpus | Fuentes | Dominios | Tamaño promedio (cantidad oraciones) |
|---|---|---|---|
| Pequeña | Corpus Deceptive Opinion Spam Corpus v1.4<br>Corpora para LARA<br>Yelp academic dataset | Hoteles, Cámaras, Restaurantes y Teléfonos celulares | 4 |
| Mediana | Corpus Deceptive Opinion Spam Corpus v1.4<br>Corpora para LARA<br>Yelp academic dataset | Hoteles, Cámaras, Restaurantes y Teléfonos celulares | 8.7 |

| | | | |
|---|---|---|---|
| Grande | Corpus Deceptive Opinion Spam Corpus v1.4<br>Corpora para LARA<br>Yelp academic dataset | Hoteles, Cámaras, Restaurantes y Teléfonos celulares | 25.45 |

De forma empírica se concluyó del análisis cualitativo (tanto para tópicos locales como globales) realizado, que los tópicos obtenidos con los algoritmos Estrella mezclaron segmentos de tópicos diferentes, aunque en varios casos los cubrimientos fueron acertados, específicamente con la variante Estrella Generalizado. Los tópicos observados con los algoritmos HAC-Complete y HAC-Average fueron más aceptables que los obtenidos con algoritmos Estrella; sin embargo, se determinó que es necesario un estudio más preciso a través de un análisis cuantitativo en el que se empleen medidas de validación interna y validaciones estadísticas para decidir cuáles de los umbrales permiten obtener mejores grupos. De esta forma, para estudiar el comportamiento de los algoritmos de agrupamiento de tópicos locales se crearon tres casos de estudio. El primero de ellos incluye 20 opiniones pequeñas, el segundo 20 opiniones medianas y el tercero 20 opiniones largas. En la Tabla 2 se ofrece una descripción de las tres colecciones creadas.

La experimentación tiene dos objetivos:

1. Identificar los valores de los umbrales a utilizar para obtener agrupamientos de mayor calidad.
2. Identificar los métodos de agrupamiento que mejor logren obtener los grupos locales.

Debido a que estas colecciones no están etiquetadas manualmente, es necesario aplicar medidas de validación interna. La mayoría de estas medidas o índices se basan en propiedades de los grupos para medir su calidad; por ejemplo, densidad, cohesión, compactación y separación. La compactación mide cuán relacionados están los elementos dentro de un grupo; la separación, mide cuán diferentes es un grupo de otros (Liu et al. 2010) (Arbelaitz et al. 2013). Las medidas a aplicar son:

- Silhouette: Es un índice de tipo suma normalizado basado en el cálculo de distancias entre y dentro de los grupos. Mide la cohesión basada en la distancia entre todos los elementos en el mismo grupo y la separación está basada en la distancia del vecino más cercano. El máximo valor indica mejores grupos (Rousseeuw 1987) (Arbelaitz et al. 2013).
- Dunn: Es un índice de tipo radio y basado en distancias. La cohesión es estimada por la distancia del vecino más cercano y la separación por el diámetro máximo del grupo. El máximo valor indica mejores grupos (Dunn 1973).

- R-Squared (RS): es considerado una medida del grado de diferencia entre grupos. Además, mide el grado de homogeneidad entre grupos. Los valores de RS se encuentran entre 0 y 1. Si RS es 0 indica que no existen diferencias entre grupos. Cuando RS es igual a 1 indica que existe una diferencia significativa entre grupos (Halkidi 2001). Así, valores mayores de RS indica mayor diferencia entre grupos y por lo tanto más homogéneos son los grupos (Kovács et al. 2005).
- Homogeneity: es una medida basada en la similitud de objetos, donde se considera que elementos en el mismo grupo son altamente similares. El máximo valor indica mejores grupos (Sharan et al. 2003) (Jiang et al. 2004).
- Separation: es una medida basada en la similitud de objetos, donde se considera que los elementos entre grupos diferentes tienen baja similitud entre ellos (Sharan et al. 2003) (Jiang et al. 2004).

Los métodos de agrupamiento considerados en el análisis son las tres variantes de HAC y las tres variantes de los algoritmos Estrella. En (Shulcloper 2010) se proponen algunas estrategias para estimar los valores de umbrales a partir de las similitudes entre objetos. Estos criterios fueron escogidos para la estimación del umbral. De ahí que la experimentación consistió en agrupar los segmentos de cada opinión considerando los algoritmos HAC y Estrella considerando las siguientes formas para estimar el umbral (Shulcloper 2010):

- Media (media de las similitudes entre todos los pares de objetos posibles).
- Máximo (media de los valores máximos de las similitudes entre cada objeto y el resto).
- Mínimo (media de los valores mínimos de las similitudes entre cada objeto y el resto).
- Combinación (media ponderada de la media de las similitudes y la media de los máximos).

En el caso del algoritmo HAC el umbral significa el valor por el cual se decide en la construcción del dendrograma si dos grupos deben mezclarse, si la similitud de los grupos es menor que el umbral, los grupos no se unirán y será detenido el proceso de agrupamiento. En el caso de los algoritmos Estrella se utiliza el umbral para hacer el corte de la similitud en la representación de los segmentos en un grafo.

Para facilitar la validación se utilizó la herramienta CVAP como componente (toolbox) del software Matlab, que permite validar diferentes algoritmos de agrupamiento con las medidas internas antes mencionadas (Wang et al. 2009). En este caso solo se utilizó la funcionalidad de

validación, es decir, se conformaron los ficheros con los resultados del agrupamiento y los ficheros de las matrices que representaban cada opinión y de esta forma se ejecutaron los índices.

El experimento realizado consistió en ejecutar el esquema propuesto para las tres colecciones descritas (Pequeña, Mediana y Larga), aplicando los diferentes algoritmos de agrupamientos para todos los umbrales. Posteriormente, los valores resultantes obtenidos para cada una de las 20 opiniones de cada corpus, fueron dados a la herramienta CVAP, la cual otorgó valores de calidad a los grupos; es decir, para las tres colecciones conformadas se evaluaron los seis algoritmos de agrupamiento, las tres variantes de HAC y las tres variantes de algoritmos Estrella. En el Anexo 20 y Anexo 21 se muestran los valores promedios obtenidos para todos los algoritmos de agrupamiento.

Luego, en respuesta al primer objetivo trazado en la experimentación, se aplicó una medida estadística de comparación entre los umbrales para cada algoritmo de agrupamiento. La técnica estadística empleada consistió en aplicar la prueba no paramétrica de Friedman para múltiples pruebas (Friedman 1937). De esta forma se buscaron diferencias significativas inferiores a 0.05 entre los umbrales. Para ello se utilizó la herramienta Keel[52], software de código abierto para la extracción del conocimiento basada en el aprendizaje evolutivo. Entre sus algoritmos posee implementado la prueba estadística de Friedman, y los métodos post-hoc de Holm, Shaffer y Iman-Davenport. La prueba de Friedman declara en la hipótesis nula que todos los elementos comparados son equivalentes y por tanto sus rangos deben ser iguales; la prueba ordena los elementos de forma separada, el primer elemento representa el mejor funcionamiento, y así sucesivamente[53]. Si se rechaza la hipótesis nula implica diferencias entre los elementos; en este caso un método post-hoc puede ser usado para encontrar las comparaciones de los pares que producen las diferencias. Para ello se observa el p-value, mientras más pequeño más fuerte es la evidencia contra la hipótesis nula (García & Herrera 2008).

En el

Anexo ***22*** se muestran las tablas correspondientes a los resultados obtenidos para cumplir el objetivo 1 de la experimentación, en este caso se muestran ejemplos para el corpus de opiniones largas. En el Anexo 23 se muestra un ejemplo de comparación de umbrales con el uso de los

[52] www.keel.es

[53] En caso que hayan valores iguales se asigna el promedio de los rangos.

métodos post-hoc para seleccionar el umbral de mayor diferencia significativa, en este caso se utilizaron los valores de la medida Separation para evaluar los grupos obtenidos con el algoritmo HAC-Complete. Se debe destacar que estos experimentos se realizaron para los tres corpus: opiniones largas, medianas y pequeñas.

De forma general las medidas de Silhouette, RSquared, Homogeneity y Separation dieron resultados aceptables para opiniones largas. En el caso de opiniones medianas y cortas RSquared, Homogeneity y Separation también dieron resultados aceptables; sin embargo, la medida Dunn no no reportó buenos resultados para las colecciones analizados. Esto se debe a que esta medida compara contra centros de grupos y los grupos que se obtienen en el contexto de las opiniones locales son muy pequeños. Una posible causa de estos resultados es que los grupos están muy cercanos y el diámetro de los mismos es grande, lo que afecto el valor del índice Dunn.

Los resultados obtenidos son satisfactorios, sobre todo si se tiene en cuenta que agrupar segmentos impone grandes retos, ya que:

- El detector de oraciones no identifica cuando no hay espacios entre signos de puntuación que sí los llevan. Si el detector de oraciones recibe como entrada las oraciones mal escritas entonces no puede identificar bien algunos segmentos, por ejemplo: "*.This camera*". Una redacción que no usa correctamente los signos de puntuación influye en gran medida en el reconocimiento de segmentos y por tanto en la obtención de similitudes entre ellos.
- Una posible causa para el hecho de obtener similitudes bajas entre segmentos es porque generalmente los usuarios hablan poco de cada tema sobre el cual opinan. En los tópicos locales es difícil que existan altos valores de similitudes entre segmentos; para los tópicos globales las similitudes entre segmentos son mayores porque varias personas pueden opinar de un mismo tema, pero una misma persona es más difícil que hable de un tópico más de una vez en una opinión.
- La resolución de correferencia o resolución de anáforas[54] es un problema que debe ser tratado en este contexto de procesamiento textual. Por ejemplo, en la siguiente oración se hace referencia a un sujeto (su primera cámara digital) que no aparece explícitamente en esa oración, sino en oraciones anteriores: *"My first was a sturdy sandproof/waterproof model, the XP, which*

[54] Son expresiones que refieren a otras expresiones, ej. pronombres (Carenini et al. 2011). Por ejemplo: "*Sally llegó pero nadie la vio a ella*", el pronombre ella es anafórico, se refiere a Sally.

*has served me very well at the beach and on boats."* Para comparar los segmentos si se tuviera en cuenta el contexto de cada uno de ellos pudieran obtenerse mayores valores de similitud. De esta forma incluir la resolución de correferencia en la etapa de segmentación constituye un factor útil con vistas a obtener segmentos más similares[55].

Luego de realizar la primera prueba estadística y obtener los mejores umbrales por agrupamiento fue necesario realizar una segunda prueba estadística, donde se compararán los algoritmos de agrupamiento a partir de los umbrales obtenidos para dar respuesta al segundo objetivo de la experimentación. En el Anexo 24 se muestra como ejemplo la entrada de datos para realizar la prueba de Friedman a partir de los resultados obtenidos en la primera prueba estadística, los cuales consisten en los valores seleccionados por algoritmo de agrupamiento y umbral evaluados por las 20 opiniones del corpus de opiniones largas. Y en el Anexo 25 se muestra para el ejemplo presentado los resultados de la prueba de Friedman al comparar los algoritmos de agrupamiento para la medida de validación interna Homogeneidad. Es necesario destacar que este experimento se repitió por cada uno de los tres corpus conformados y para las cuatro medidas de validación interna. Los resultados finales de esta segunda prueba estadística para responder al objetivo 2 de la experimentación se muestran la Tabla 3.

**Tabla 3 Algoritmos que obtuvieron mejores medidas de calidad para tópicos locales, según la prueba estadística de Friedman.**

| Medidas | Opiniones largas | Opiniones medianas | Opiniones pequeñas |
|---|---|---|---|
| Silhouette | HAC Complete (max) | - | - |
| Rsquared | HAC Complete (max) | HAC Complete (comb) | HAC Complete (max) |
| Homogeneity | HAC Complete (max) | HAC Complete (comb) | HAC Average (mean) |
| Separation | HAC Single (max) | HAC Single (comb) | HAC Average (mean) |

De forma general los criterios seguidos para el análisis de las pruebas estadísticas fueron:

1. Para saber cuál es el mejor umbral por cada medida interna:
   - Cuando hay diferencias significativas se escoge el umbral de menor ranking de los pares más diferentes señalados por el test de Holms.
   - Cuando no hay diferencias significativas se escoge el umbral de menor ranking.
2. Para escoger el mejor algoritmo de agrupamiento:

[55] Los algoritmos TextTiling y C99 no incluyen elementos de resolución de anáforas. En el estudio realizado de los algoritmos de segmentación no se encontraron algoritmos disponibles que incluyeran esta funcionalidad del procesamiento textual.

- Seleccionar el umbral que predomina en la mayoría de las medidas internas.
- Si todas las medidas para un algoritmo agrupamiento dan umbrales distintos se escoge el umbral que haya tenido mayor diferencia significativa en su análisis para ese algoritmo. Para ello se comparan los valores que dio la prueba de Friedman, si hubo diferencias significativas se selecciona el de menor ranking, pero si no hubo diferencias significativas en ninguna de las medidas entonces se selecciona el umbral donde el p-value fue menor.

De esta forma se puede concluir que aunque las pruebas estadísticas no arrojaron diferencias significativas entre los algoritmos de agrupamiento para el corpus de opiniones largas, se sugiere el uso de HAC-Complete con umbral máximo debido a que las medidas internas evaluadas coinciden en obtener en su mayoría la mejor calidad del agrupamiento para este algoritmo. Así, para el corpus de opiniones medianas, se recomienda HAC-Complete con el umbral combinado y para el corpus de opiniones pequeñas se sugiere HAC-Average con umbral medio.

### 3.3.2 Análisis del agrupamiento de tópicos globales

Para validar el agrupamiento de tópicos globales se creó un caso de estudio conformado por 12 colecciones con 50 opiniones de tamaño variable cada una. En la Tabla 4 se muestra la descripción de este caso de estudio. Para este estudio se tienen en cuenta los resultados del análisis de la obtención de tópicos locales. Además, debido a que se cuenta con una colección de opiniones mayor a la de los tópicos locales, la cual generará una representación de grandes dimensiones, resulta de interés conocer si el uso de LSA como modelo de representación textual permite encontrar mejores grupos de tópicos. El reto aquí consiste en identificar el algoritmo de agrupamiento, el umbral y el modelo de representación textual que mejor se comporte en la detección de tópicos globales a partir de colecciones de opiniones de tamaño variable. Por tanto, la experimentación tiene dos objetivos:

1. Evaluar los grupos obtenidos al identificar tópicos globales con el algoritmo HAC-Complete y umbral máximo, debido a que la detección de tópicos globales se puede modelar como la detección de tópicos locales en una opinión.
2. Identificar el modelo de representación textual (VSM o LSA) que mejor logre obtener los grupos globales.

**Tabla 4 Características de las colecciones creadas para tópicos globales.**

| Corpus | Fuentes | Dominio |
|---|---|---|

| | | |
|---|---|---|
| 1. OpinionSpam-Hotel | Deceptive Opinion Spam Corpus v1.4 | hoteles |
| 2. LARA-Hotel | LARA | hoteles |
| 3. LARA-Laptop | LARA | laptops |
| 4. LARA-Camera | LARA | cámaras digitales |
| 1. LARA-MobilePhone | LARA | teléfonos celulares |
| 6. LARA-TV | LARA | televisores |
| 7. Yelp-Restaurant | Yelp academic dataset | restaurantes |
| 8. SFUReview-Books | The SFU Review Corpus | libros |
| 9. SFUReview-Cookware | The SFU Review Corpus | utensilios de cocina |
| 10. TAD-Hotel | TripAdvisor Annotated Dataset | hoteles |
| 11. JPDA-Car | JDPASentimentCorpus | automóviles |
| 12. Imdb-Movies | Large Movie Review Dataset v1.0 | películas |

Debido a que estas colecciones, tal como se explicó en el epígrafe 3.2, no tienen los tópicos etiquetados manualmente, es necesario aplicar medidas de validación interna. De ahí que utilizaremos las mismas medidas de validación del epígrafe 3.3.1. Para facilitar la validación se utilizó la herramienta CVAP de la misma forma que se empleó para los tópicos locales.

Como parte de este análisis se repitió el experimento utilizando la forma de representación textual LSA. Los resultados fueron similares a los obtenidos con VSM, la diferencia entre los tópicos obtenidos con ambos modelos se basa en que LSA es capaz de agrupar segmentos semánticamente, a diferencia de VSM que agrupa los segmentos de acuerdo a la frecuencia de sus términos. En el Anexo 26 se expone un ejemplo donde tres segmentos que fueron agrupados por LSA en un tópico fueron ubicados por VSM en tres tópicos distintos. En la columna de LSA se destaca en negrita las palabras relacionadas semánticamente y en VSM las palabras que se repiten. En la columna de VSM están en letra cursiva aquellos segmentos que se encuentran juntos en el tópico obtenido con LSA.

Como resultado se obtuvieron los valores promedios para las medidas de Silhouette, RSquared, Homogeneity y Separation por cada uno de los modelos de representación textual utilizados, como se muestra en la Tabla 5.

**Tabla 5 Valores promedios de las medidas internas aplicadas a la validación de grupos de tópicos globales.**

| | Silhouette | RSquared | Homogeneity | Separation |
|---|---|---|---|---|
| VSM | 0.4991 | 1 | 0.5947 | 0.3451 |
| LSA | 0.4873 | 1 | 0.5245 | 0.2825 |

De los experimentos realizados se puede concluir que el empleo del algoritmo HAC-Complete con umbral máximo para los tópicos globales fue satisfactorio, mostrándose una compactación y

separación de los grupos con valores aceptables. Respecto a la experimentación con los diferentes modelos de representación textual no se puede afirmar que existan diferencias significativas entre sus resultados, esto puede ser causado porque, según algunos autores (Landauer & Foltz 1998) (Dennis et al. 2003), LSA ha obtenido mejores resultados para representar palabras individuales y párrafos (captura la similitud de lo que tratan dos palabras o dos pasajes), que oraciones; y los segmentos que analizamos en los experimentos están compuestos por oraciones.

## 3.4 Evaluación de la propuesta para etiquetar grupos de segmentos

Para evaluar la selección de la medida de similitud semántica a emplear se propuso hallar los sustantivos más relacionados entre sí por cada tópico para tres opiniones escogidas aleatoriamente de la colección Deceptive Opinion Spam Corpus en el caso de hoteles, Yelp academic dataset para los restaurantes y LARA para las opiniones sobre cámaras fotográficas. Cada opinión seleccionada corresponde a un dominio diferente de las colecciones.

Las medidas de similitud basadas en el conocimiento cuantifican en cuanto se parecen dos conceptos, basados en la información que contienen en una jerarquía “es-un” (Pedersen & Michelizzi 2004). Estas medidas pueden ser de relación semántica y de similitud semántica. Las medidas de relación semántica indican una noción más general de relación, no están específicamente atadas a la forma del concepto; la similitud es considerada un tipo de relación entre dos palabras, y cubre un rango más amplio de relaciones entre conceptos que incluye relaciones de similitud extra, tales como "es un tipo de", "es un ejemplo específico de", "es una parte de", "es el opuesto de". Por otra parte, las medidas de similitud semántica consideran cuáles conceptos semánticamente similares están relacionados en base de su parecido o semejanza (Gomaa 2013). De esta forma para esta evaluación solo se emplearon medidas de similitud semántica, es decir, todas las mencionadas en el Anexo 11, excepto HirstStOnge y Lesk.

Las etiquetas obtenidas fueron evaluadas por los tres aspectos propuestos por (Tagarelli & Karypis 2012), los que declaran que el agrupamiento basado en segmentos es capaz de producir grupos cuyas etiquetas son más útiles. Los aspectos para determinar la utilidad de las etiquetas son:

- Coherencia entre términos: se espera que la descripción de un grupo sea coherente con el resto del tópico, en esencia, la coherencia tópica debe reflejar la homogeneidad de los objetos del texto dentro de un grupo dado.

- Presencia de términos discriminativos: se refiere al entendimiento de cuántos términos descriptivos son capaces de discriminar cada grupo del resto en el grupo.
- Amplitud de la descripción: se refiere al cubrimiento del tópico de los términos descriptivos en cada grupo.

En el dominio de Hoteles se analizaron los siguientes tópicos (en negrita están marcadas los sustantivos más representativos[56] del tópico) y en la Tabla 6, Tabla 7 y Tabla 8 se muestran las etiquetas obtenidas por cada una de las medidas por cada dominio (en cursiva se señalan aquellas que coinciden con los sustantivos marcados en el tópico):

Tópico 1: [**bath**, product, crabtree, evelyn, include, **shampoo**, **conditioner**, mouthwash, baby, **lotion**]

Tópico 2: [**room**, king, **bed**, comfortable, nice, feather, **pillow**, request, type, problem]

Tópico 3: [walk, art, **museum**, buddy, guy, blue, club, street, **cab**, **ride**, science, industry, cost]

**Tabla 6 Ejemplos de etiquetas para opiniones del dominio de Hoteles.**

| Medidas | Tópico 1 | Tópico 2 | Tópico 3 |
|---|---|---|---|
| Resnik | *bath, shampoo, lotion* | *room, bed, pillow* | art, *cab, museum* |
| Lin | bath, lotion, product | *room, bed, pillow* | art, street, *cab* |
| JiangConrath | bath, product, lotion | *room, bed, pillow* | art, street, *cab* |
| Path | *bath, shampoo, lotion* | room, feather, pillow | art, *museum*, street |
| WuPalmer | *bath, shampoo, lotion* | room, feather, king | art, *museum*, street |
| LeacockChodorow | *bath, shampoo, lotion* | room, pillow, bed | art, *museum*, street |

En el dominio de Restaurantes se analizaron los siguientes tópicos:

Tópico 1: [**bread**, french, fry, **onion**, **ring**, good]

Tópico 2: [new, **order**, minute, sit, people, place, count, patio, serve, **sandwich**, **salad**, fish, fry, extra, crispy, end, thing, jerky, qualify, church, lend, all, **dinner**, companion, leave, partial, uneaten]

Tópico 3: [iced, **tea**, good, place, dispense, **sweet**, low, give, rock, crack, ask, **sugar**, carry, artificial, **sweetener**, pink, stuff, future, reuben, **hamburger**, assume, today, aberration, possibly, due, **cook**, family, emergency]

**Tabla 7 Ejemplos de etiquetas para opiniones del dominio de Restaurantes.**

| Medidas | Tópico 1 | Tópico 2 | Tópico 3 |
|---|---|---|---|
| Resnik | french, *ring* | *sandwich, salad, dinner* | *tea, hamburger, sugar* |
| Lin | french, *ring* | *sandwich, salad, dinner* | *tea*, *sweetener*, stuff |

[56] Los que indican mayor relación con el resto de los sustantivos

| | | | |
|---|---|---|---|
| JiangConrath | *bread*, french, *ring* | *order*, place, people | *sugar*, *sweetener*, stuff |
| Path | *bread*, fry, *ring* | place, end, count | place, carry, stuff |
| WuPalmer | *bread*, *onion*, fry | minute, count, people | place, crack, stuff |
| LeacockChodorow | *bread*, fry, *ring* | minute, count, people | place, carry, stuff |

En el dominio de Cámaras se analizaron los siguientes tópicos:

Tópico 1: [**zoom**, feature, **camera**, clear, make, good, **beginner**, easy, exciting, **novice**, kind, **photographer**]

Tópico 2: [originally, buy, **camera**, last, year, crush, **pocket**, working, refurbish]

Tópico 3: [win, buy, **camera**, satisfy, **picture**, turn, point, **shoot**, wall, **focus**, want, save]

De esta forma se concluye que la medida Resnik se acerca más a los sustantivos señalados por tópicos para representar los grupos de segmentos. Esta medida obtiene de forma ordenada[57] los términos más relacionados entre sí y con el resto de los términos del tópico. En su mayoría coinciden los sustantivos marcados con anterioridad de forma manual con los identificados por el algoritmo que emplea esta medida de similitud.

**Tabla 8 Ejemplos de etiquetas para opiniones del dominio de Cámaras.**

| Medidas | Tópico 1 | Tópico 2 | Tópico 3 |
|---|---|---|---|
| Resnik | *camera*, *beginner*, *novice* | *camera*, *pocket*, *crush* | *camera*, *picture*, wall |
| Lin | *camera*, *beginner* | *camera*, *pocket*, *crush* | *camera*, wall, *picture* |
| JiangConrath | *camera*, kind, *beginner* | *camera*, *pocket*, year | *picture*, wall, turn |
| Path | *beginner*, *novice*, *photographer* | *crush*, *pocket*, refurbish | *picture*, *focus*, turn |
| WuPalmer | *beginner*, *novice*, *photographer* | *camera*, *pocket*, *crush* | *camera*, *picture*, wall |
| LeacockChodorow | *beginner*, *novice*, *photographer* | *camera*, *pocket*, year | *camera*, *picture*, wall |

## 3.5 Formas de evaluación de la detección de tópicos en opiniones

Para realizar la evaluación de la propuesta primero se hizo un estudio de las validaciones realizadas en dos áreas de investigación: la aplicación de técnicas de detección de tópicos enfocada a la minería de opinión y la minería de opiniones basada en aspectos.

En el epígrafe 2.2 se describieron algunos enfoques propuestos donde se aplicaron técnicas de detección de tópicos en el contexto de la minería de opinión. De forma general las etapas comunes que desarrollan son el pre-procesamiento del texto, la identificación de unidades textuales

[57] En el mismo orden que se presenta en la opinión.

(mayormente oraciones) y la representación textual. Los corpus que han utilizado las investigaciones estudiadas abordan opiniones emitidas en blogs, artículos de noticias, tweets y otros. Para evaluar sus algoritmos gran parte de los autores etiquetan manualmente las colecciones para identificar los tópicos y en su mayoría éstas no están disponibles. Las métricas más comunes que emplean son precisión (precision), exhaustividad (recall), medida-F (F-measure) y exactitud (accuracy). En la Tabla 9 se muestran valores obtenidos con estas métricas en las investigaciones estudiadas.

En el epígrafe 2.1.1 se mencionaron algunos enfoques de la extracción de términos de aspectos para posteriormente realizar un análisis de sentimientos. Los trabajos citados realizan evaluaciones enfocadas a medir la efectividad de los algoritmos para detectar la polaridad de las opiniones y otros a validar predicciones de evaluaciones de aspectos, es decir, que no están orientados a evaluar la extracción de aspectos[58]. De esta forma, para evaluar la presente propuesta se considerarán solo los trabajos que pertenecen al área de investigación de la detección de tópicos enfocada a la minería de opinión.

**Tabla 9 Valores de medidas externas utilizadas en la literatura para evaluar tópicos detectados.**

| Fuente | Resultados obtenidos | |
|---|---|---|
| (Cai et al. 2008) | Experimento 1 | Experimento 2 |
| | Precision=41.3%<br>Recall = 61.3% | Precision=58.1%<br>Recall = 55.6% |
| (Gangemi et al. 2014) | Precision = 0.72<br>Recall = 0.64<br>F1 = 0.68<br>Accuracy = 0.66 | |
| (Fernández & Núñez 2013) | Accuracy = 58% | Precision = 0.579<br>Recall = 0.584<br>F1= 0.578 |
| (Zhang et al. 2013) | Accuracy = 76% | |
| (Hattori & Nadamoto 2013) | Experimento 1 | Experimento 2 |
| | Precision = 77% | Precision = 69% |

Con el objetivo de poder comparar nuestra propuesta con los resultados publicados, se seleccionó del corpus LARA las opiniones de hoteles (Wang et al. 2010). Este corpus tiene los aspectos segmentados por categorías; específicamente los términos de aspectos para cada categoría en las

[58] Excepto un solo trabajo encontrado (Hu et al. 2004), que evalúa la etapa de extracción de aspectos y obtiene buenos resultados en sus cálculos: Fmeasure= 0.72, Precision=0.79, Recall=0.67

que se evalúan las opiniones. Para conformar el corpus los autores extrajeron 235 793 opiniones del sitio de TripAdvisor en el período de un mes. Luego, realizaron un pre-procesamiento en el que convirtieron las palabras a minúsculas, eliminaron signos de puntuación y palabras vacías, así como los términos que ocurrían menos de 5 veces en el corpus, y hallaron las raíces de las palabras. Manualmente seleccionaron palabras "semilla" para cada aspecto predefinido y lo usaron como entrada para un algoritmo de segmentación de aspectos. Luego, descartaron aquellas oraciones que fallaron al ser asociadas con alguno de los aspectos. De esta forma obtuvieron los términos asociados a los aspectos en todas las opiniones sobre 1850 hoteles.

```
<Author>selizabethm
<Content>Wonderful time- even with the snow! What a great experience! From the goldfish in the room (which my daughter loved) to the fact that the valet parking staff who put on my chains on for me it was fabulous. The staff was attentive and went above and beyond to make our stay enjoyable. Oh, and about the parking: the charge is about what you would pay at any garage or lot- and I bet they wouldn't help you out in the snow!
<Date>Dec 23, 2008
<Rating>5 4 5 5 5 5 5 -1
<Aspects>
3 5(time):1 48(wonderful):1 2884(snow):1
1 44(experience):1
8 0(staff):1 40(love):1 66(park):1 270(fabulous):1 457(valet):1
  467(daughter):1 627(chain):1 6035(goldfish):1
0
0
4 0(staff):1 537(attentive):1 567(enjoyable):1 738(beyond):1
8 38(lot):1 42(pay):1 66(park):1 92(help):1 136(charge):1 936(garage):1
  1821(bet):1 2884(snow):1
0
```

**Fig. 5 Fragmento de una opinión del corpus de hoteles para LARA.**

En la Fig. ***5*** se muestra un fragmento de una opinión, donde es posible apreciar el formato en que estas se expresan. Los aspectos están divididos en 8 categorías, los cuales coinciden con las categorías que son evaluadas en TripAdvisor por cada opinión: Generalidades, Valor, Habitaciones,

Ubicación, Limpieza, Registro, Servicio y Servicio de Negocio. El algoritmo de segmentación que utilizan tiene como entrada una colección de opiniones, un conjunto de palabras claves, un vocabulario, un umbral y un límite de iteración, y como salida las opiniones divididas en oraciones con asignaciones por aspecto. Sus pasos son:

1. Dividir las opiniones en oraciones.
2. Hacer coincidir las palabras claves por aspecto en cada oración por opinión y guardar las coincidencias para cada aspecto.
3. Asignar a la oración una etiqueta de aspecto por el máximo de coincidencias encontradas, si hay un empate asignar la oración con múltiples aspectos.
4. Calcular la medida $X^2$ para cada palabra en el vocabulario.
5. Ordenar las palabras de cada aspecto respecto a su valor $X^2$ y unir las primeras palabras para cada aspecto en su lista correspondiente de palabras claves por aspecto.
6. Si la lista de palabras claves por aspecto no varía o la iteración excede el límite, ir al paso 7 de lo contrario ir al 2.
7. Anotar las oraciones con asignaciones de aspecto.

El corpus procesado tiene 108891 opiniones sobre 1850 hoteles. Para la experimentación se conformó un caso de estudio con las 243 opiniones emitidas sobre el hotel Melia Caribe Tropical de Punta Cana. Para descubrir los tópicos locales se empleó el algoritmo de agrupamiento HAC-Complete con umbral máximo debido a que en este corpus predominan las opiniones largas. El resto de los algoritmos utilizados para esta evaluación consistieron en el algoritmo TextTiling para la segmentación y VSM como modelo de representación textual.

Las medidas de validación externa a emplear son Precision, Recall y F-measure (Moghaddam & Ester 2013), sus ecuaciones se muestran a continuación:

$$Precision = \frac{CategoriesExtracted \cap GoldStandardCategories}{CategoriesExtracted} \quad (0.1)$$

$$Recall = \frac{CategoriesExtracted \cap GoldStandardCategories}{GoldStandardCategories} \quad (0.2)$$

$$Fmeasure = \frac{2\ x\ Recall\ x\ Precision}{(Recall + Precision)} \quad (0.3)$$

*GoldStandardCategories* está representada por las categorías señaladas en el corpus LARA para cada opinión. Específicamente se identificaron a partir del primer número que se declara dentro de

la etiqueta <Aspects> el cual representa la cantidad de los términos de aspectos presentes en esa opinión por cada categoría. Si el valor es mayor que cero indica que existen términos relacionados a la categoría, sino el usuario no emitió ningún criterio sobre dicha categoría.

La variable *CategoriesExtracted* representa las categorías extraídas por un método que propusimos para asociar las categorías a las etiquetas de los tópicos encontrados. Este método consiste en utilizar el método Resnik de cálculo de similitud de WordNet implementado en la biblioteca WS4J de forma similar a como se empleó en la etapa 7 del esquema; sin embargo, en este caso se adaptó de forma diferente, para obtener, de las ocho categorías declaradas por el corpus LARA, cuáles son similares a los términos extraídos que forman parte de la etiqueta del tópico. Así, el cálculo de la similitud con Resnik se empleó entre las categorías y los términos que componen la etiqueta del tópico. Para cada término su calcula la similitud semántica con cada una de las 8 categorías, luego se suman todas las similitudes obtenidas por categoría y se ordenan. Aquella categoría que posea mayor puntuación será la categoría asignada para ese tópico; en el caso que haya más de una categoría con el mismo valor de máxima puntuación, también es seleccionada. Por ejemplo, para la etiqueta de tópico "*bag, room, dinner*" el método obtiene la categoría "*rooms*" y para la etiqueta "*breakfast, lunch*" el método obtiene más de una categoría: "*location, overall, rooms*". En la Tabla 10 se muestran las puntuaciones obtenidas para ambos ejemplos, en el último caso se obtienen iguales puntuaciones para tres categorías por lo que se seleccionan las tres primeras.

**Tabla 10 Ejemplo de puntuaciones obtenidas para asignar categorías.**

| **Puntuaciones de las categorías para la etiqueta "*bag, room, dinner*"** | **Puntuaciones de las categorías para la etiqueta "breakfast, lunch"** |
|---|---|
| **rooms** 7.050262193094899 | **location** 1.228727898773817 |
| overall 5.601132120884786 | **overall** 1.228727898773817 |
| location 2.952764186109192 | **rooms** 1.228727898773817 |
| business 0.0 | business 0.0 |
| check 0.0 | check 0.0 |
| cleanliness 0.0 | cleanliness 0.0 |
| service 0.0 | service 0.0 |
| value 0.0 | value 0.0 |

**Tabla 11 Comparación de los valores obtenidos en la evaluación externa con los valores máximos encontrados en la literatura.**

| | **Precision** | **Recall** | **F-Measure** |
|---|---|---|---|
| Valores obtenidos en la experimentación | 0.74 | 0.86 | 0.78 |

| | | | |
|---|---|---|---|
| Valores máximos de la literatura | 0.77 | 0.64 | 0.68 |

Como resultado de este experimento se promediaron los valores de precision, recall y F-measure obtenidos para cada una de las 243 opiniones de la colección. De esta forma los valores promedios resultantes son: precision 0.7352, recall 0.8562 y F-measure 0.7770. Respecto a los resultados observados en la literatura los valores obtenidos en la experimentación de recall y F-measure dieron valores más altos. El caso de la precisión puede verse afectada por el método que se empleó en la evaluación para asociar las categorías a las etiquetas de los tópicos encontrados, ya que, con el objetivo de ser lo más restrictivos posibles al evaluar, solamente seleccionamos aquellas categorías con máxima puntuación, pudiéndose quedar fuera algunos términos con puntuaciones altas. Además, el corpus de referencia para algunas opiniones no tiene en cuenta las palabras que coinciden con el nombre de la categoría, por ejemplo "room" o "location" y no las cuentan como términos de aspecto, lo cual puede influir de forma negativa en la comparación realizada porque en la referencia anotada puede especificar 0 en la cantidad de términos de esta categoría y realmente si se habla de ella en la opinión.

## 3.6 Conclusiones parciales del capítulo

Se diseñó e implementó el marco de trabajo OpinionTopicDetection que integra las herramientas Lucene, OpenNLP, TreeTagger, S-Space y WS4J y se basa en una adaptación en cinco etapas del esquema general propuesto en el Capítulo 2. OpinionTopicDetection tiene un diseño extensible, siguiendo una arquitectura de cuatro capas y nueve paquetes que tienen interdependencias. Además, puede ser usado como una biblioteca por otras aplicaciones para el análisis textual, debido a que permite instanciar las funcionalidades de otras bibliotecas de forma independiente.

La aplicación OpinionTD, desarrollada a partir de la instanciación de OpinionTopicDetection, permite descubrir tópicos locales y globales en opiniones de manera no supervisada y con valores de precisión y exhaustividad de 0.74 y 0.86, respectivamente. Esta aplicación requiere la especificación del corpus a procesar y los valores para los parámetros: tipo de tópico a obtener; algoritmo de segmentación; forma de representación textual; algoritmo de agrupamiento y tipo de umbral para obtener las particiones de grupos.

El método de agrupamiento que obtuvo los mejores resultados de las medidas de validación interna para la mayoría de las colecciones textuales al detectar tópicos locales fue HAC Complete a partir una representación VSM, siendo el umbral calculado según la media de los máximos el que ofreció

mejores resultados para opiniones grandes, la media ponderada de la media y el máximo fue el que permitió obtener los mejores agrupamiento en opiniones medianas y el cálculo del umbral según la media de las similitudes para opiniones pequeñas. El mejor agrupamiento de segmentos en la detección de tópicos globales se obtuvo con el algoritmo HAC-Complete con umbral máximo.

El método propuesto para la detección de tópicos basado en el etiquetamiento de grupos mediante la extracción de sustantivos permite obtener resultados satisfactorios cuando se calcula la similitud semántica con la medida Resnik, ya que esta medida se acerca más a los sustantivos señalados por tópicos para representar los grupos de segmentos.

# Conclusiones y recomendaciones

Como resultado de esta investigación se desarrolló la herramienta OpinionTD que instancia el marco de trabajo OpinionTopicDetection y permite de manera no supervisada y efectiva detectar los tópicos de las opiniones, obteniéndose valores de precisión y exhaustividad de 0.74 y 0.86, respectivamente; cumpliéndose de esta forma el objetivo general propuesto, ya que:

1. La detección de tópicos utilizando el agrupamiento de segmentos previamente identificados, facilita el análisis de sentimientos por tópicos de opiniones; ya que los grupos de segmentos detectados brindan una representación de las opiniones en la cual se obtendrán los criterios otorgados para un mismo tema. Acerca de la segmentación y la detección se concluye:

    - El método de segmentación no supervisado TextTiling permitió hallar segmentos de opiniones más detallados y de alta cohesión léxica que los encontrados por el algoritmo C99 con tales propósitos. Se descartó la aplicación de los modelos probabilísticos para la segmentación porque tienen alto costo computacional y son supervisados.
    - El algoritmo de agrupamiento HAC, en sus variantes completo y promedio, permitió obtener los agrupamientos de segmentos de mayor calidad según las medidas internas de validación, superando a las otras variantes de HAC y a los algoritmos Estrella. El cálculo del umbral de similitud utilizando la media de los máximos de las similitudes garantizó que HAC completo realizara la mejor detección de tópicos en opiniones largas y medianas; mientras que HAC promedio permitió detectar tópicos en opiniones pequeñas, estimando el umbral a partir del promedio de las similitudes entre segmentos.

2. El esquema general desarrollado para el descubrimiento de tópicos en opiniones consta de siete etapas que permiten partir de una opinión o conjunto de opiniones y obtener tópicos locales y globales correspondientes a los grupos de segmentos que abordan un mismo tema. Este esquema se distingue por realizar primeramente la etapa de segmentación y posteriormente al agrupamiento de los segmentos para detectar los tópicos. Permite representar los textos con VSM y LSA, y aunque no existen diferencias significativas en los resultados alcanzados por una y otra representación, los tópicos descubiertos a partir de textos representados con VSM reportan los mejores resultados. Como resultado del desarrollo del esquema se creó:

- Un analizador léxico que integra las técnicas de pre-procesamiento deseadas y hereda del analizador general de Lucene, permitiendo pre-procesar el texto en el siguiente orden: corregir palabras mal escritas, llevar a minúscula todo el texto, eliminar las palabras vacías, tokenizar y lematizar. La entrada de esta etapa son las unidades textuales identificadas del corpus de opiniones y la salida son tokens o términos.
- Un algoritmo que permite etiquetar de manera efectiva los grupos de segmentos de forma no supervisada y por términos. El cálculo de las similitudes semánticas entre los sustantivos utilizando la medida Resnik permite obtener los mejores resultados en el etiquetamiento, ya que se acerca más a los sustantivos señalados por tópicos para representar los grupos de segmentos. Esta forma de etiquetamiento siguiendo las ideas del análisis de sentimiento basado en aspectos permite obtener mejores descripciones de los tópicos.

3. Se diseñó e implementó el marco de trabajo OpinionTopicDetection que integra las herramientas Lucene, OpenNLP, TreeTagger, S-Space y WS4J y se basa en una adaptación en cinco etapas del esquema general propuesto en el Capítulo 2. OpinionTopicDetection tiene un diseño extensible, siguiendo una arquitectura de cuatro capas y nueve paquetes que tienen interdependencias. Además, puede ser usado como una biblioteca por otras aplicaciones para el análisis textual, debido a que permite instanciar las funcionalidades de otras bibliotecas de forma independiente.

Derivadas del estudio realizado, así como de las conclusiones generales emanadas del mismo, se recomienda:

- Incluir en el marco de trabajo OpinionTopicDetection otros algoritmos de segmentación que traten la resolución de co-referencias o anáforas.
- Utilizar técnicas de programación paralela para optimizar el pre-procesamiento textual.
- Integrar el esquema desarrollado a PosNegOpinion, de forma tal que se logre calcular la polaridad de las opiniones por tópicos.

# Referencias bibliográficas


Abella, R. & Medina, J., 2014. Segmentación lineal de texto por tópico. *Serie Gris CENATAV*.

Abella, R. & Medina, J., 2010. Text Segmentation by Clustering Cohesion. *CIARP*, pp.261–268.

Abella Raúl & Medina, J., 2013. Modelos y métodos para la segmentación de texto basados en tópicos. *Proceedings of CICCI at the 16th International Convention and Fair Informatica*.

Aggarwal, C.C. & Zhai, C., 2012. *Mining Text Data*, Springer.

Aiello, L.M. et al., 2013. Sensing trending topics in Twitter. *Multimedia, IEEE Transactions*, 15(6), pp.1268–1282.

Aletras, N. & Stevenson, M., 2014. Labelling Topics using Unsupervised Graph-based methods. In *ACL Short Papers*. p. 661.

Allan, J., 2002. *Topic Detection and Tracking Event-based Information Organization*,

Allan, J. et al., 1998. Topic Detection and Tracking Pilot Study Final Report. *Proceedings of the Broadcast News Transcription and Understanding Workshop (Sponsored by DARPA)*.

Amores, M., 2013. *Detección no supervisada de la polaridad de las opiniones*. Universidad Central "Marta Abreu" de Las Villas.

Amores, M., Arco, L. & Artiles, M., 2015. PosNeg opinion : Una herramienta para gestionar comentarios de la Web. *Revista Cubana de Ciencias Informáticas*, 9(1), pp.20–12.

Amores, M., Borroto, C. & Arco, L., 2015. SentiWordNet 4.0 and SpanishSenti-WordNet assisting Polarity Detection. *Eureka Workshop*.

Anderson, D.J. & Linden-reed, J., 2015. Getting Started with Kanban for Software Development. *DZone Refcardz*. Available at: https://dzone.com/refcardz/getting-started-kanban [Accessed December 29, 2015].

Anon, A Tutorial on Clustering Algorithms. *Hierarchical Clustering Schemes*. Available at: http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/hierarchical.html [Accessed October 12, 2015].

Arbelaitz, O. et al., 2013. An extensive comparative study of cluster validity indices. *PATTERN RECOGNITION*, (December 2015).

Arco, L., 2008. *Agrupamiento basado en la intermediación diferencial y su valoración utilizando la teoría de los conjuntos aproximados*. Universidad Central "Marta Abreu" de Las Villas.

Aslam, J.; Pelekhov, K. and Rus, D., 1998. Static and Dynamic Information Organization with Star Clusters. *In Proceedings of the 1998 Conference on Information Knowledge Management, Baltimore, MD*.

Becker, H., Naaman, M. & Gravano, L., 2011. Beyond Trending Topics : Real-World Event Identification on Twitter. *ICWSM*, 11, pp.438–441.

Becker, J. & Kuropka, D., 2003. Topic-based Vector Space Model. *Business Information Systems*.

Beeferman, D., Berger, A. & Lafferty, J., 1999. Statistical Models for Text Segmentation. *Machine learning*, 34(1-3), pp.177–210.

Bengio, Y. et al., 2003. A Neural Probabilistic Language Model. *Journal ofMachine Learning Research*, 3, pp.1137–1155.

Berlanga-llavori, R. et al., 2008. Conceptual Subtopic Identification in the Medical Domain. *Lecture Notes in Artificial Intelligence*, 5290, pp.312–321.

Berry, M.W. & Kogan, J., 2010. *Text Mining Applications and Theory*, Wiley.

Bessis, N. & Dobre, C., 2014. *Big Data and Internet of Things : A Roadmap for Smart Environments*, Springer.

Blei, D.M., 2011. Introduction to Probabilistic Topic Models. *Communications of the ACM*, pp.1–16.

Blei, D.M. & Lafferty, J.D., 2008. *Topic models*,

Blei, D.M. & Mcauliffe, J.D., 2008. Supervised topic models. *Advances in neural information processing systems*, pp.121–128.

Blei, D.M., Ng, A.Y. & Jordan, M.I., 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3, pp.993–1022.

Bolelli, L., Ertekin, S. & Giles, C.L., 2009. Topic and Trend Detection in Text Collections Using Latent Dirichlet Allocation. *Advances in Information Retrieval.*, pp.776–780.

Cai, K. et al., 2008. Leveraging Sentiment Analysis for Topic Detection. *International Conference on Web Intelligence and Intelligent Agent Technology*.

Cambria, E. et al., 2013. New Avenues in Opinion Mining and Sentiment Analysis. *Knowledge based approaches to concept level sentiment analysis*, (April), pp.15–21.

Carenini, G., Murray, G. & Ng, R., 2011. *Methods for Mining and Summarizing Text Conversations*,

Carenini, G. & Ng, R.T., 2013. Towards Topic Labeling with Phrase Entailment and Aggregation. *HLT-NAACL*, (June), pp.179–189.

Carranza, J.C.G., 2014. Course of Text based Information Retrieval. In *Department of Computer Science, KU Leuven, Belgium*.

Cataldi, M. et al., 2010. Emerging Topic Detection on Twitter based on Temporal and Social Terms Evaluation. *MDMKDD*.

Chang, J. et al., 2009. Reading Tea Leaves : How Humans Interpret Topic Models. *Advances in neural information processing systems*, pp.288–296.

Chang, J. & Kim, I., 2014. Research Trends on Graph-Based Text Mining. *International Journal of Software Engineering and Its Applications*, 8(4), pp.147–156.

Chen, C.C. et al., 2003. Life Cycle Modeling of News Events Using Aging Theory. *Machine Learning: ECML*, pp.47–59.

Choi, F., 2000. Advances in domain independent linear text segmentation. *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference.*, pp.26–33.

Choi, F., Wiemer, P. & Moore, J., 2001. Latent Semantic Analysis for Text Segmentation. *Proceedings of EMNLP*, 102, pp.109–117.

Clark, S., 2014. Vector Space Models of Lexical Meaning. In S. Lappin & C. Fox, eds. *Handbook of Contemporary Semantics*. pp. 1–43.

Cordobés, H. et al., 2014. Graph-based Techniques for Topic Classification of Tweets in Spanish. *International Journal of Artificial Intelligence and Interactive Multimedia*, 2(5), pp.31–37.

Deerwester, S. et al., 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6), pp.391–407.

Dennis, S. et al., 2003. *Introduction to Latent Semantic Analysis*,

Dong, H., Hui, S.C. & He, Y., 2006. Structural Analysis of Chat Messages for Topic Detection. *Online Information Review*.

Du, L. & Johnson, M., 2013. Topic Segmentation with a Structured Topic Model. *Proceedings of NAACL-HLT 2013*, (June), pp.190–200.

Dueñas, R., L'Huillier, G. & Velásquez, J., 2013. Sentiment Polarity of Trends on the Web Using Opinion Mining and Topic Modeling.

Dunn, J.C., 1973. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, (3), pp.32–57.

Eisenstein, J. & Barzilay, R., 2008. Bayesian Unsupervised Topic Segmentation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing.. Association for Computational Linguistics.*, pp.334–343.

Fernández, A. & Núñez, L., 2013. Sentiment Analysis and Topic Detection of Spanish Tweets : A Comparative Study of NLP Techniques. *Procesamiento del Lenguaje Natural*, 50, pp.45–52.

Ferret, O., 2002. Using collocations for topic segmentation and link detection. *Proceedings of the 19th international conference on Computational linguistics. Association for Computational Linguistics.*, 1, pp.1–7.

Fowler, M., 2005. Inversion of Control. Available at: http://martinfowler.com/bliki/InversionOfControl.html.

Friedman, M., 1937. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *Journal of the American Statistical Association*, 32(200), pp.675–701.

Gago, A., Pérez, A. & Medina, J., 2007. ACONS: a New Algorithm for Clustering Documents. *Progress in Pattern Recognition, Image Analysis and Applications*.

Galley, M. & Mckeown, K., 2003. Discourse Segmentation of Multi-Party Conversation. *Proceedings of ACL*, pp.562–569.

Gangemi, A., Presutti, V. & Recupero, D.R., 2014. Frame-Based Detection of Opinion Holders and Topics : A Model and a Tool. *Computational Intelligence Magazine, IEEE*, 9(1), pp.20–30.

García, S. & Herrera, F., 2008. An Extension on " Statistical Comparisons of Classifiers over Multiple Data Sets " for all Pairwise Comparisons. *Journal of Machine Learning Research*, 9, pp.2677–2694.

Gil, R., Badía, J. & Pons, A., 2003. Extended Star Clustering Algorithm. *Proceedings of the Iberoamerican Congress on Pattern Recognition, Speech and Image Analysis (CIARP 2003)*, pp.480–487.

Glass, K. & Colbaugh, R., 2010. Toward Emerging Topic Detection for Business Intelligence : Predictive Analysis of " Meme " Dynamics. *Association for the Advancement of Artificial Intelligence*.

Gomaa, W.H., 2013. A Survey of Text Similarity Approaches. *International Journal of Computer Applications*, 68(13), pp.13–18.

Grefenstette, E. et al., 2014. New Directions in Vector Space Models of Meaning. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: Tutorials*.

Guille, A. et al., 2013. Information Diffusion in Online Social Networks : A Survey. *ACM SIGMOD Record*, 42(2), pp.17–28.

Halkidi, M., 2001. On Clustering Validation Techniques. *Intelligent Information Systems Journal*, pp.107–145.

Hamamoto, M. & Pan, J., 2005. A Comparative Study of Feature Vector-Based Topic Detection Schemes. *Proceedings of the 2005 International Workshop on Challenges in Web Information Retrieval and Integration*.

Han, J., Xie, X. & Woo, W., 2010. Context-based Local Hot Topic Detection for Mobile User. *Adjunct Proceedings of the 8th International Conference on Pervasive Computing*.

Hattori, Y. & Nadamoto, A., 2013. Tip information from social media based on topic detection. *International Journal of Web Information Systems*, 9(1), p.14.

Hearst, M.A., 1997. TextTiling : Segmenting Text into Multi-paragraph Subtopic Passages. *Computational Linguistics*, 23(1), p.32.

Heinonen, O., 1998. Optimal Multi-Paragraph Text Segmentation by Dynamic Programming. *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, 2, pp.1484–1486.

Hernández, L. & Pagola, M., 2009. TextLec: Método para la segmentación por tópicos en textos científico-técnicos. *Serie Gris CENATAV*.

Hingmire, S., 2013. Document Classification by Topic Labeling. *SIGIR*, pp.877–880.

Hofmann, T., 1999. Probabilistic Latent Semantic Indexing. *Proceedings of the Twenty Second Annual International SIGIR Conference on Resea rch and Development in Information Retrieval*, pp.289–296.

Hofmann, T., 2001. Unsupervised Learning by Probabilistic Latent Semantic Analysis. *Machine Learning*, 42(1-2), pp.177–196.

Holz, F. & Teresniak, S., 2010. Towards automatic detection and tracking of topic change. *Computational linguistics and intelligent text processing*, pp.327–339.

Hu, M., Liu, B. & Street, S.M., 2004. Mining and Summarizing Customer Reviews. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Huang, X. et al., 2013. A Topic Detection Approach Through Hierarchical Clustering on Concept Graph. *Applied Mathematics & Information Sciences*, 2295(6), pp.2285–2295.

Ingersoll, G.S., Morton, T.S. & L.Farris, A., 2013. *Taming Text*,

Jacobson, I., 2000. *El Proceso Unificado de Desarrollo de Software*,

Jameel, S., 2014. *Latent Probabilistic Topic Discovery for Text Documents Incorporating Segment Structure and Word Order*.

Jameel, S. & Lam, W., 2013. An Unsupervised Topic Segmentation Model Incorporating Word Order. *SIGIR*, pp.203–212.

James, M., 2015. Scrum. *DZone Refcardz*, pp.1–6. Available at: https://dzone.com/refcardz/scrum [Accessed December 29, 2015].

Jiang, D., Tang, C. & Zhang, A., 2004. Cluster Analysis for Gene Expression Data : A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 16(11), pp.1370–1386.

Jiang, Y., Meng, W. & Yu, C., 2011. Topic Sentiment Change Analysis. *Machine Learning and Data Mining in Pattern Recognition*, pp.443–457.

José H. Canós, P.L. yMª C.P., Metodologías Ágiles en el Desarrollo de Software.

Joty, S., Carenini, G. & Ng, R.T., 2013. Topic Segmentation and Labeling in Asynchronous Conversations. *Journal of Artificial Intelligence Research*, 47, pp.521–573.

Jurafsky, D. & Martin, J.H., 2007. *Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition.*,

Jurgens, D. & Stevens, K., 2010. The S-Space Package : An Open Source Package for Word Space Models. *Proceedings of the ACL 2010 System Demonstrations*, (July), pp.30–35.

Kern, R., Graz, A.- & Granitzer, M., 2009. Efficient Linear Text Segmentation Based on Information Retrieval Techniques. *MEDES*.

Kleedorfer, F., Knees, P. & Pohle, T., 2008. Oh Oh Oh Whoah! *ISMIR 2008 – Session 2d – Social and Music Networks 1.*, pp.287–292.

Kniberg, H., 2009. *Kanban vs Scrum How to make the most of both*,

Kniberg, H., Jeff, P. De & Cohn, M., 2007. *Scrum y XP desde las trincheras*,

Koike, D. et al., 2013. Time Series Topic Modeling and Bursty Topic Detection. *International Joint Conference on Natural Language Processing*, (October), pp.917–921.

Kovács, F., Legány, C. & Babos, A., 2005. Cluster Validity Measurement Techniques. *Proceedings of the 6th International Symposium of Hungarian Researchers on Computational Intelligence*.

Landauer, T.K. & Foltz, P.W., 1998. An Introduction to Latent Semantic Analysis. *Discourse Processes*, pp.259–284. Available at: lsa.colorado.edu/papers/dp1.LSAintro.pdf.

Lazo-cortes, M., Ruiz-shulcloper, J. & Alba-cabrera, E., 2001. An overview of the evolution of the concept of testor. *Pattern recognition*, 34, pp.753–762.

Lee, D.D., Hill, M. & Seung, H.S., 2000. Algorithms for Non-negative Matrix Factorization. *Advances in Neural Information Processing Systems*.

Li, J. & Cardie, C., 2013. TopicSpam: a Topic-Model-Based Approach for Spam Detection. *ACL*, pp.217–221.

Lin, C., 2011. Automatically extracting polarity-bearing topics for cross- domain sentiment classification. *49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.

Lin, C. et al., 2012. Weakly-supervised joint sentiment-topic detection from text. *Knowledge and Data Engineering, IEEE Transactions*, 24(6).

Liu, B., 2010. Sentiment Analysis and Subjectivity. , pp.1–38.

Liu, T., Zhang, N.L. & Chen, P., 2014. Hierarchical Latent Tree Analysis for Topic Detection. *ECML PKDD*, pp.256–272.

Liu, Y. et al., 2010. Understanding of Internal Clustering Validation Measures. *2010 IEEE International Conference on Data Mining*.

Lloret, E., 2009. *Topic Detection and Segmentation in Automatic Text Summarization*,

Lu, M. et al., 2011. Probabilistic Latent Semantic Analysis for Broadcast News Story Segmentation. *INTERSPEECH*, (August), pp.1109–1112.

Lu, Y. et al., 2013. Health-Related Hot Topic Detection in Online Communities Using Text Clustering. *PLoS ONE*, 8(2), pp.1–9.

Manning, C., Prabhakar Raghavan & Schütze, H., 2008. *An Introduction to Information Retrieval*, Cambridge University Press.

Manning, C.D. et al., 2014. The Stanford CoreNLP Natural Language Processing Toolkit. *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp.55 – 60.

Mao, X., 2012. Automatic Labeling Hierarchical Topics. *CIKM*.

Marcheggiani, D. & Oscar, T., 2014. Hierarchical Multi-label Conditional Random Fields for Aspect-Oriented Opinion Mining. *ECIR*, pp.273–285.

Martín-Wanton, T., Gonzalo, J. & Amigó, E., 2013. An Unsupervised Transfer Learning Approach to Discover Topics for Online Reputation Management. *CIKM*, pp.1565–1568.

Massung, S. & Hockenmaier, J., 2013. Structural Parse Tree Features for Text Representation. *In Semantic Computing (ICSC), 2013 IEEE Seventh International Conference*.

Mat, H. et al., 2014. Microblog hot topic detection based on topic model using term correlation matrix. *Proceedings of the 2014 International Conference on Machine Learning and Cybernetic*, pp.13–16.

Mei, Q., Shen, X. & Zhai, C., 2007. Automatic Labeling of Multinomial Topic Models. *KDD*.

Mikolov, T., 2013. Learning Representations of Text using Neural Networks. *NIPS Deep LearningWorkshop*, pp.1–31.

Misra, H. et al., 2011. Text segmentation : A topic modeling perspective. *Information Processing and Management*, 47(4), pp.528–544. Available at: http://dx.doi.org/10.1016/j.ipm.2010.11.008.

Moens, M. & Busser, R. De, 2001. Generic Topic Segmentation of Document Texts. *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pp.418–419.

Moens, M.F., Li, J. & Seng, T., 2014. *Mining User Generated Content*,

Moghaddam, S. & Ester, M., 2013. Opinion Mining in Online Reviews : Recent Trends. *Tutorial at WWW2013*.

Nallapati, R. & Allan, J., Capturing Term Dependencies using a Language Model based on Sentence Trees. *Proceedings of the eleventh international conference on Information and knowledge management. ACM.*, pp.383–390.

Ngoc, T. & Do, Q., 2012. *A graph model for text analysis and text mining*. Universite de Lorraine.

Ott, M. et al., 2011. Finding Deceptive Opinion Spam by Any Stretch of the Imagination. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp.309–319.

Palmer, D.D. & Hearst, M.A., 1994. Adaptive Sentence Boundary Disambiguation. *Proceedings of the 4th Conference on Applied Natural Language Processing*, (2).

Panem, S. et al., 2014. Entity Tracking in Real-Time using Sub-Topic Detection on Twitter. *Advances in Information Retrieval*, pp.528–533.

Pang, B. & Lee, L., 2008. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, 2, pp.1–135.

Pavlopoulos, I. (John), 2014. *Aspect based sentiment analysis*. Athens University of Economics and Business.

Pedersen, T. & Michelizzi, J., 2004. WordNet :: Similarity - Measuring the Relatedness of Concepts Measures of Relatedness. *Intelligent Systems Demonstrations*, (Patwardhan 2003), pp.1024–1025.

Pérez, A. & Medina., J.E., 2007. A clustering algorithm based on generalized stars. *Proceedings of the 5th International Conference on Machine Learning and Data Mining in Pattern Recognition (MLDM 2007).*

Pérez, A.N. & González, K.S., 2014. *Desambiguación del sentido de las palabras*. Santa Clara: Universidad Central "Marta Abreu" de Las Villas.

Perez-Tellez, F. et al., 2010. Clustering Weblogs on the Basis of a Topic Detection Method. *Advances in Pattern Recognition*, pp.342–351.

Petkos, G., Aiello, L. & Skraba, R., 2014. A soft frequent pattern mining approach for textual topic detection. *WIMS*.

Petkos, G., Papadopoulos, S. & Kompatsiaris, Y., 2014. Two-level message clustering for topic detection in Twitter. *Proceedings of the SNOW 2014 Data Challenge*.

Pons-porrata, A. et al., 2004. Jerartop: A New Topic Detection System. *Progress in Pattern Recognition, Speech and Image Analysis*, pp.446–453.

Pons-porrata, A. & Berlanga-llavori, R., 2007. Topic discovery based on text mining techniques. *Information Processing and Management*, 43, pp.752–768.

Pons-porrata, A., Berlanga-llavori, R. & Ruiz-shulcloper, J., 2002. Temporal-Semantic Clustering of Newspaper Articles for Event Detection. *Pattern Recognition in Information Systems*.

Porrata, A.P., 2004. *Desarrollo de algoritmos para la estructuración dinámica de información y su aplicación a la detección de sucesos*. Universidad Jaume I, Castellón.

Prasad, S. et al., 2011. Emerging Topic Detection using Dictionary Learning. *CIKM*.

Purver, M., 2011. Topic Segmentation. In *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*. pp. 1–28.

Rajaraman, K. & Tan, A., 2001. Topic Detection, Tracking and Trend Analysis Using Self-organizing Neural Networks. *Advances in Knowledge Discovery and Data Mining*.

Ramage, D. & Manning, C.D., 2011. Partially Labeled Topic Models for Interpretable Text Mining. *KDD*.

Read, J. et al., 2012. Sentence Boundary Detection : A Long Solved Problem? *COLING*, (December 2012), pp.985–994.

Ren, W. & Han, K., 2014. Sentiment Detection of Web Users Using Probabilistic Latent Semantic Analysis. *Journal of Multimedia*, 9(10), pp.1194–1200.

Resnik, P., 1995. Using information content to evaluate semantic similarity in a taxonomy. *Proceedings of IJCAI*.

Reynar, J., 1998. *Topic Segmentation: Algorithms and Applications*. University of Pennsylvania.

Reynar, J.C., 1994. An automatic method of finding topic boundaries. *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pp.331–333.

Reynar, J.C., 1999. Statistical Models for Topic Segmentation. *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pp.357–364.

Reynar, J.C., Ratnaparkhi, A. & Science, I., 1997. A Maximum Entropy Approach to Identifying Sentence Boundaries. *Proceedings of the fifth conference on Applied natural language processing. Association for Computational Linguistics*, pp.16 –19.

Riedl, M. & Biemann, C., 2012a. How Text Segmentation Algorithms Gain from Topic Models. *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp.0–4.

Riedl, M. & Biemann, C., 2012b. TopicTiling: A Text Segmentation Algorithm based on LDA. In *ACL 2012 50th Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop*. Jeju Island , Korea.

Rosenberg, D. & Stephens, M., 2007. *Use Case Driven Object Modeling with UML Theory and Practice*, Apress.

Rousseeuw, P.J., 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathmatics*, 20, pp.53–65.

Sahlgren, M., 2006. *The Word-Space Model Using distributional analysis to represent syntagmatic and paradigmatic relations between words*. Stockholm University.

Salton, G., Wong, A. & Yang, C.S., 1975. A Vector Space Model for Automatic Indexing. *Information Retrieval and Language Processing, Communications of the ACM.*, 18(11), pp.613–620.

Sayyadi, H. & Raschid, L., 2013. A Graph Analytical Approach for Topic Detection. *ACM Transactions on Internet Technology*, 13(2).

Seijo, F.C., Luna, J.M.F. & Guadix, J.F.H., 2011. *Recuperación de Información. Un enfoque práctico y multidisciplinar* RAMA, ed., Madrid, Spain.

Seo, Y. & Sycara, K., 2004. *Text clustering for topic detection*, Pittsburgh, Pennsylvania.

Shafiei, M.M. & Milios, E.E., 2008. A Statistical Model for Topic Segmentation and Clustering. *Advances in Artificial Intelligence*, pp.283–295.

Sharan, R., Maron-katz, A. & Shamir, R., 2003. CLICK and EXPANDER : a system for clustering and visualizing gene expression data. *Bioinformatics*, 19(14), pp.1787–1799.

Sharma, N.R. & Chitre, V.D., 2014. 2014 Opinion Mining, analysis and its challenges.pdf. *International Journal of Innovations & Advancement in Computer Science*, p.7.

Shulcloper, J.R., 2010. *Reconocimiento lógico combinatorio de patrones: teoría y aplicaciones*.

Simon, A. et al., 2013. Leveraging lexical cohesion and disruption for topic segmentation. *International Conference on Empirical Methods in Natural Language Processing, EMNLP*.

Sonawane, S.S. & Kulkarni, A., 2014. Graph based Representation and Analysis of Text Document : A Survey of Techniques. *International Journal of Computer Applications*, 96(19), pp.1–8.

Steyvers, M. & Griffiths, T., 2004. Probabilistic Topic Models. *Handbook of latent semantic analysis*, 427(7), pp.424–440.

Stokes, N., 2004. *Applications of Lexical Cohesion Analysis in the Topic Detection and Tracking Domain*. National University of Ireland, Dublin.

Tagarelli, A. & Karypis, G., 2012. A segment-based approach to clustering multi-topic documents. *Knowledge and information systems*, 34(3), pp.563–595.

Thompson, C., 2014. USF: Chunking for Aspect Term Identification & Polarity Classification. *Proceedings of the 8th International Workshop on Semantic Evaluation*, (SemEval), pp.790–795.

Titov, I. & Mcdonald, R., 2008. Modeling Online Reviews with Multi-grain Topic Models. *Proceedings of the 17th international conference on World Wide Web. ACM.*, pp.111–120.

Toh, Z., Way, F. & Wang, W., 2014. DLIREC: Aspect Term Extraction and Term Polarity Classification System. *Proceedings of the 8th International Workshop on Semantic Evaluation*, (SemEval), pp.235–240.

Torres, C. & Arco, L., 2015a. *MONOGRAFIA - Detección de tópicos*, Santa Clara: Samuel Feijoó.

Torres, C. & Arco, L., 2015b. *MONOGRAFIA - Modelos para la representación textual*, Santa Clara: Samuel Feijoó.

Torres, C. & Arco, L., 2015c. *MONOGRAFIA - Segmentación por tópicos*, Santa Clara: Samuel Feijoó.

Torres, C. & Arco, L., 2016. Segmentación y detección de tópicos en textos de opiniones. *III Conferencia Internacional en Ciencias Computacionales e Informáticas (CICCI) Informatica 2016*.

Torres, C., Arco, L. & Amores, M., 2015. Propuesta de incorporación de técnicas de detección de tópicos a PosNeg Opinion. *Compumat*, p.10.

Turney, P.D., 2010. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, 37, pp.141–188.

Valle, K. & Ozt, P., 2011. Graph-based Representations for Text Classification. *India-Norway Workshop on Web Concepts and Technologies*.

Vulic, I., 2011. Identifying Word Translations from Comparable Corpora Using Latent Topic Models. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pp.479–484.

Vulic, I. et al., 2012. Probabilistic Topic Modeling in Multilingual Settings : A Short Overview of Its Methodology and Applications. *Proceedings of the NIPS Workshop on Cross-Lingual Technologies (xLiTe)*, pp.1–11.

Wang, H., Lu, Y. & Zhai, C., 2010. Latent Aspect Rating Analysis on Review Text Data : A Rating Regression Approach. *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp.783–792.

Wang, H., Lu, Y. & Zhai, C., 2011. Latent Aspect Rating Analysis without Aspect Keyword Supervision. *KDD*, pp.618–626.

Wang, K., Wang, B. & Peng, L., 2009. CVAP: Validation for cluster analyses. *Data Science Journal*, 8(May), pp.88–93.

Wang, X. & Mccallum, A., 2006. Topics over time: a non-Markov continuous-time model of topical trends. *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp.424–433.

Wang, X. & Wang, J., 2013. A Method of Hot Topic Detection in Blogs Using N -gram Model. *JOURNAL OF SOFTWARE*, 8(1), pp.184–191.

Wattenhofer, R. & Cselle, G., 2007. BuzzTrack : Topic Detection and Tracking in Email. *IUI*.

Wayne, C.L., 2000. Multilingual Topic Detection and Tracking : Successful Research Enabled by Corpora and Evaluation. *LREC*.

Wayne, C.L., Meade, F. & Ames, A., 2007. Topic Detection & Tracking ( TDT ) Overview & Perspective.

Wibowo, A., Handojo, A. & Halim, A., 2011. Application of Topic Based Vector Space Model with WordNet. *International Conference on Uncertainty Reasoning and Knowledge Engineering*, (5), pp.1–4.

Xiaolin, Y.I. et al., 2013. An Improved Single-Pass Clustering Algorithm Internet-oriented Network Topic Detection. *Fourth International Conference on Intelligent Control and Information Processing (ICICIP)*, (August 1992), pp.560–564.

Xie, W. et al., 2012. TopicSketch : Real-time Bursty Topic Detection from Twitter. *Data Mining (ICDM), 2013 IEEE 13th International Conference*.

Xu, T. & Oard, D.W., 2011. Wikipedia-based Topic Clustering for Microblogs. *ASIST*.

Xu, W., Liu, X. & Gong, Y., 2003. Document Clustering Based On Non-negative Matrix Factorization. *SIGIR*, pp.267–273.

Yan, X., 2013. Chinese Microblog Topic Detection Based on the Latent Semantic Analysis and Structural Property. *Journal of Networks*, 8(4), pp.917–923.

Yan, X. et al., 2013. Learning Topics in Short Texts by Non-negative Matrix Factorization on Term Correlation Matrix. *Proceedings of the 13nd SIAM International Conference on Data Mining*.

Ye, J. et al., 2006. *Protej: Biomedical Topic Detection and Tracking*,

Yin, H., Cui, B. & Lu, H., 2013. A Unified Model for Stable and Temporal Topic Detection from Social Media Data. *IEEE*, pp.661–672.

Zhang, H., Wang, C. & Lai, J., 2014. Topic Detection in Instant Messages. *13th International Conference on Machine Learning and Applications*.

Zhang, L. et al., 2013. Intelligent facial emotion recognition and semantic-based topic detection for a humanoid robot. *Expert Systems With Applications*, 40(13), pp.5160–5168. Available at: http://dx.doi.org/10.1016/j.eswa.2013.03.016.

Zhang, L. & Liu, B., 2014. Aspect and Entity Extraction for Opinion Mining. In *Data Mining and Knowledge Discovery for Big Data*. Springer Berlin Heidelberg, pp. 1–40.

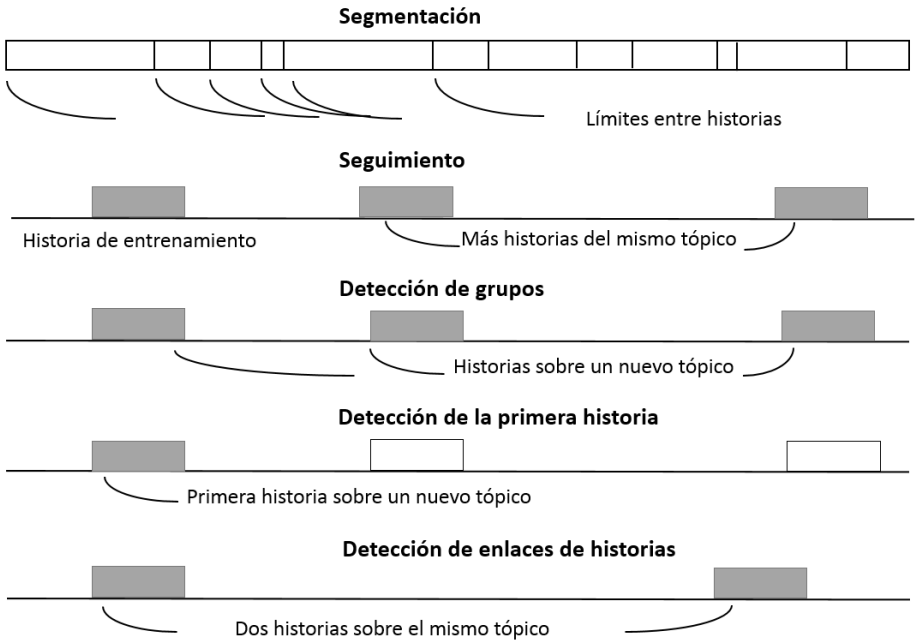
Zhang, Z., Nie, J. & Wang, H., 2015. TJUdeM : A Combination Classifier for Aspect Category Detection and Sentiment Polarity Classification. *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, (SemEval), pp.772–777.

Zhao, W.X., 2011. Comparing Twitter and Traditional Media using Topic Models. *Advances in Information Retrieval*, pp.338–349.

Zheng, D. & Li, F., 2009. Hot Topic Detection on BBS Using Aging Theory. *Web Information Systems and Mining*, pp.129–138.

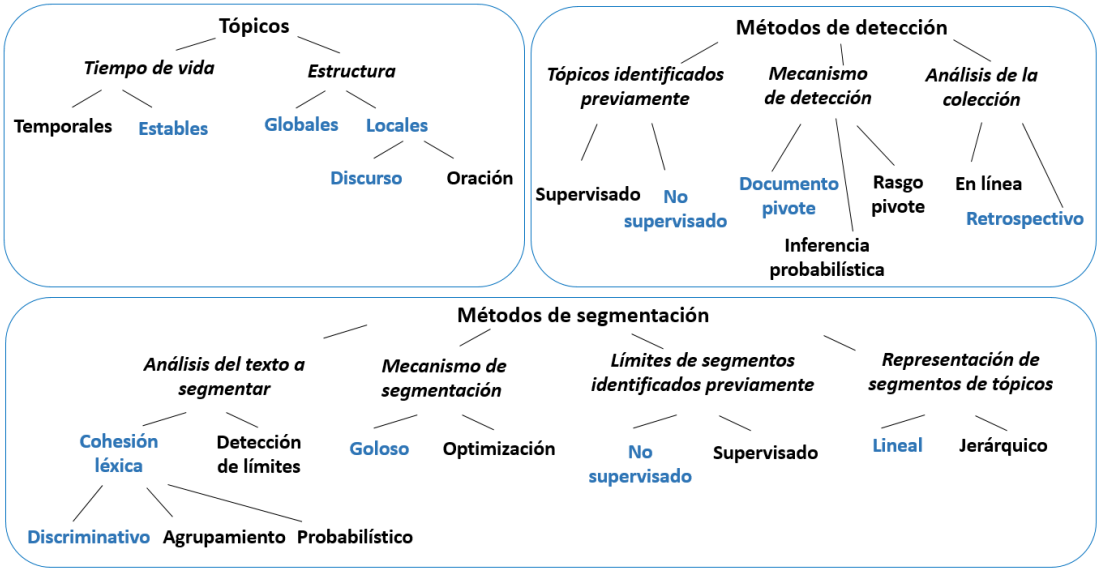
# Anexos

**Anexo 1 Tareas de TDT. Fuente:** (Wayne 2000)**.**

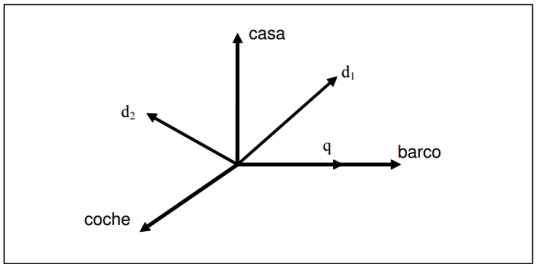



5. Segmentación de historias (Story segmentation): encontrar regiones homogéneas en el texto tópicamente.
6. Seguimiento (Tracking): encontrar historias adicionales sobre un tópico dado.
7. Detección de la primera historia (First Story Detection; FSD): reconocer el comienzo de un nuevo tópico en el flujo de historias.
8. Detección de grupos (Cluster Detection): detectar y agrupar nuevos tópicos, es decir, agrupar todas las historias tal como llegan, basándose en los tópicos que ellas presentan.
9. Detección de enlaces de historias (Story Link Detection)**:** decidir si dos historias seleccionadas aleatoriamente pertenecen al mismo tópico.

## Anexo 2 Clasificaciones de tópicos, métodos de detección y segmentación.

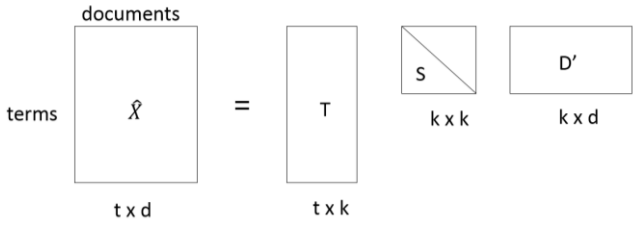


## **Anexo 3 Ejemplo de un VSM en un espacio tridimensional. Fuente:** (Seijo et al. 2011)**.**




En este ejemplo del modelo VSM en un espacio tridimensional se considera una colección que únicamente contiene los términos: barco, casa y coche; y sólo dos documentos, $d_1$ y $d_2$. Se define como: $d_1$ = "barco casa", $d_2$ = "casa coche", es decir, V= {barco, casa, coche} y D = {$d_1$, $d_2$}. Por lo tanto, el modelo vectorial de este ejemplo consistirá en un espacio tridimensional. Se escoge un esquema de pesos binario, el mismo que en el modelo booleano clásico. Conforme a este esquema, los documentos $d_1$ y $d_2$ se representan como los siguientes vectores que definen su posición en el espacio tridimensional: $d_1$ = (1, 1, 0) y $d_2$ = (0, 1, 1). Si se realiza la consulta "barco", dicha consulta q se representará en el modelo vectorial clásico como: q = (1, 0, 0), donde el 1 significa que el término "barco" está presente en la consulta, y donde los ceros representan que los términos "casa" y "coche" no figuran en ella.

## **Anexo 4 Esquema de SVD reducido de una matriz término-documento. Fuente:** (Deerwester et al. 1990)**.**




$$\hat{X} = T\,S\,D'$$

La matriz original es aproximada usando el valor k singular mayor y sus vectores singulares correspondientes.

T tiene columnas ortogonales de tamaño uno (T´T=I)

*D* tiene columnas ortogonales de tamaño uno *(D´D=I)*

*S* matriz diagonal de valores singulares

*t* es el número de filas de *X*

*d* es el número de columnas de *X*

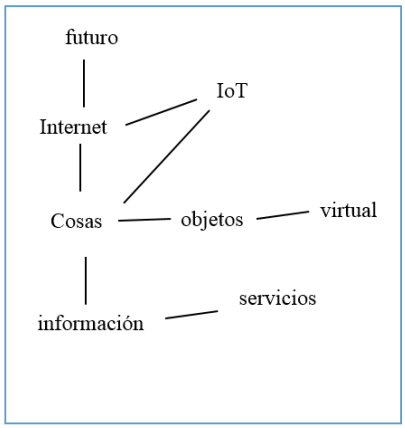
*m* es el rango de *X* ($\leq min(t, d)$)

*k* es el número escogido de dimensiones en el modelo reducido ($k \leq m$)

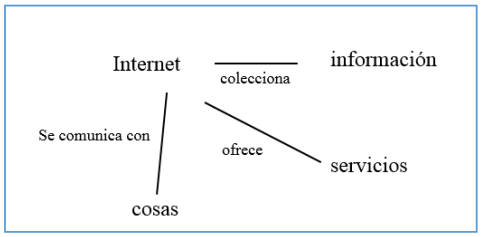
## Anexo 5 Representaciones de tres tipos de grafos para un fragmento de texto: coocurrencia de palabras claves, relaciones gramaticales y grafos conceptuales.

Como parte del futuro Internet, la Internet de las Cosas (IoT) intenta integrar, coleccionar información y ofrecer servicios a un espectro variado de cosas físicas usadas en diferentes dominios. Las "cosas" son objetos del día a día para el cual IoT ofrece una presencia virtual en Internet, ubica una identidad específica y direcciones virtuales, y adiciona capacidades de auto-organizar y comunicar con otras cosas sin la intervención humana.

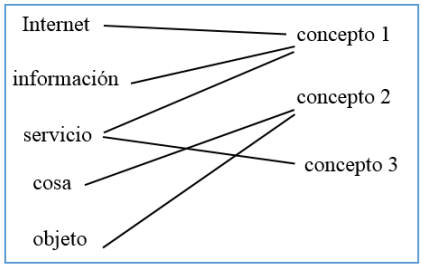
(1) Grafo de coocurrencia de palabras claves



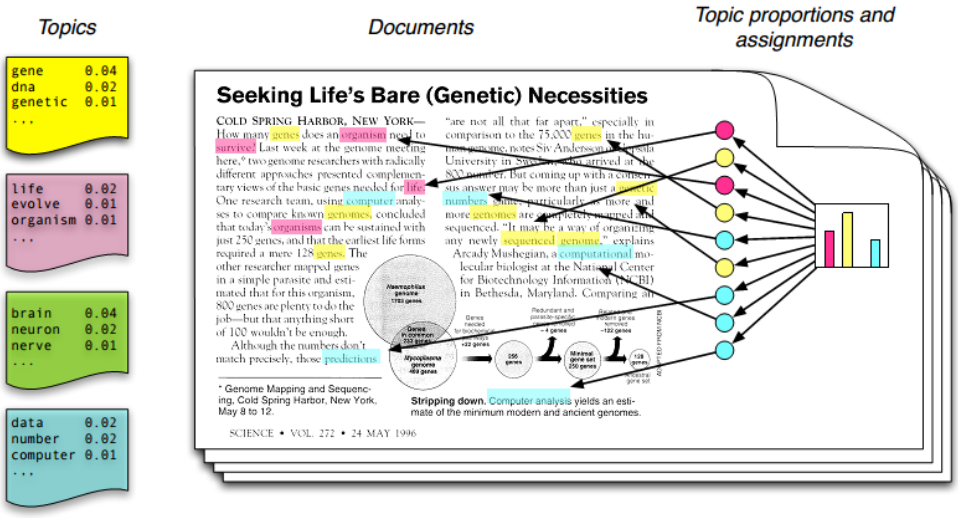

(2) Grafo de relaciones gramaticales



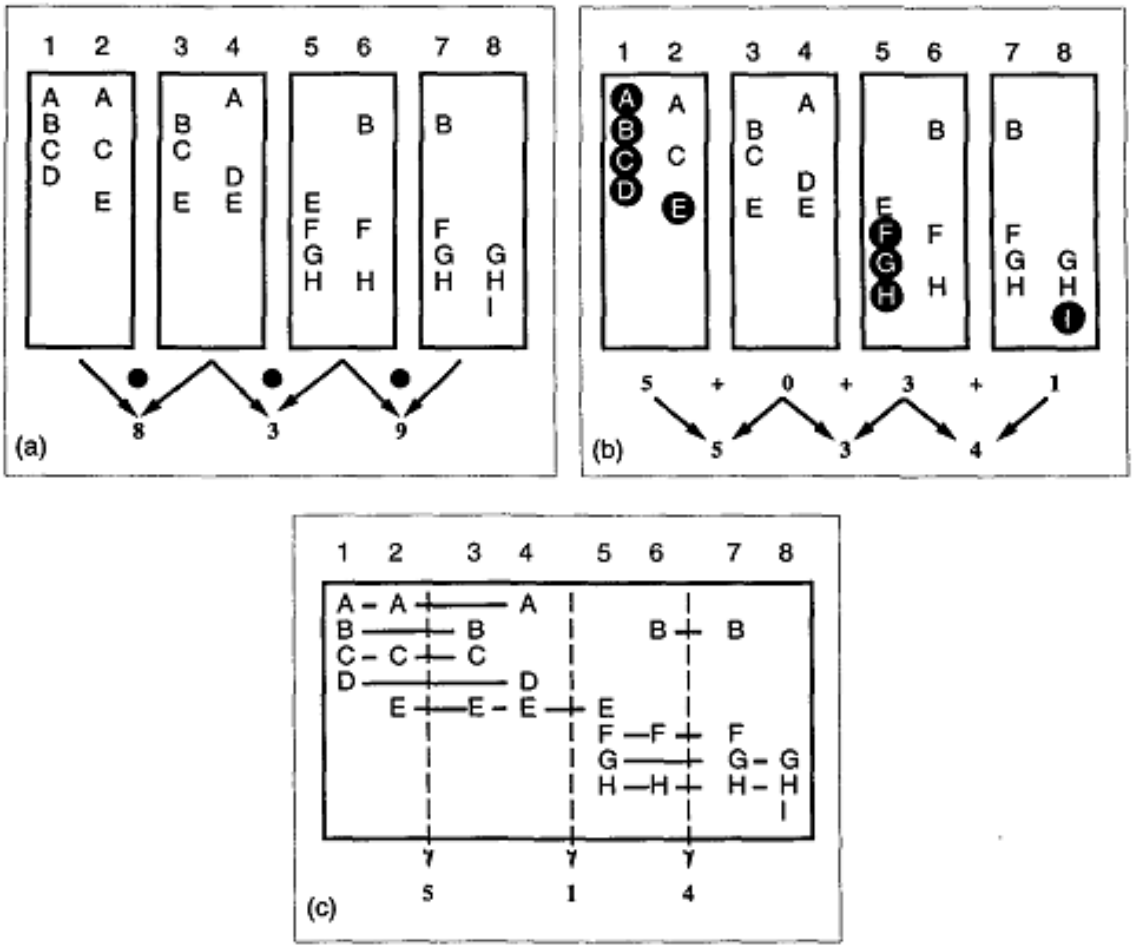

(3) Grafo bipartito de conceptos




## Anexo 6 Asignaciones de palabras de un documento a tópicos predefinidos con LDA. **Fuente:** (Blei 2011).



**Anexo 7 Formas de calcular la puntuación léxica entre los espacios entre oraciones para el algoritmo TextTiling. Fuente:** (Hearst 1997)**.**



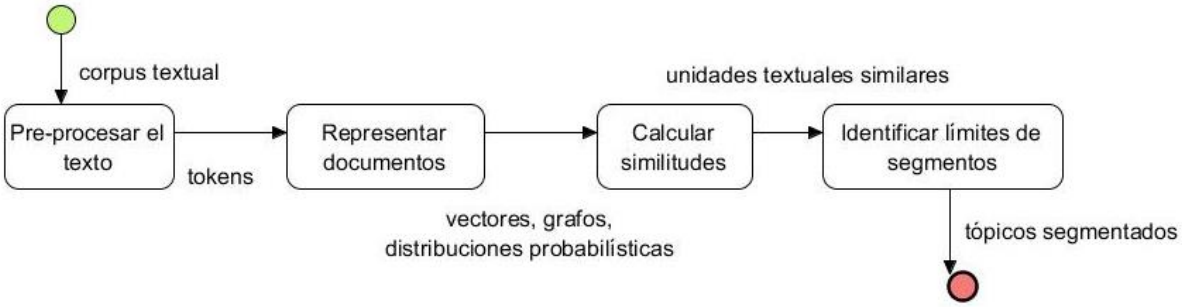

Los números indican la secuencia de oraciones, las columnas de letras significan los términos en la oración dada.

(a) Bloques: producto de vectores de conteo de palabras en el bloque de la izquierda y la derecha. (un vector contiene el número de veces que cada objeto léxico ocurre en el bloque correspondiente, el producto interior es normalizado para ubicarlo entre 0 y 1).

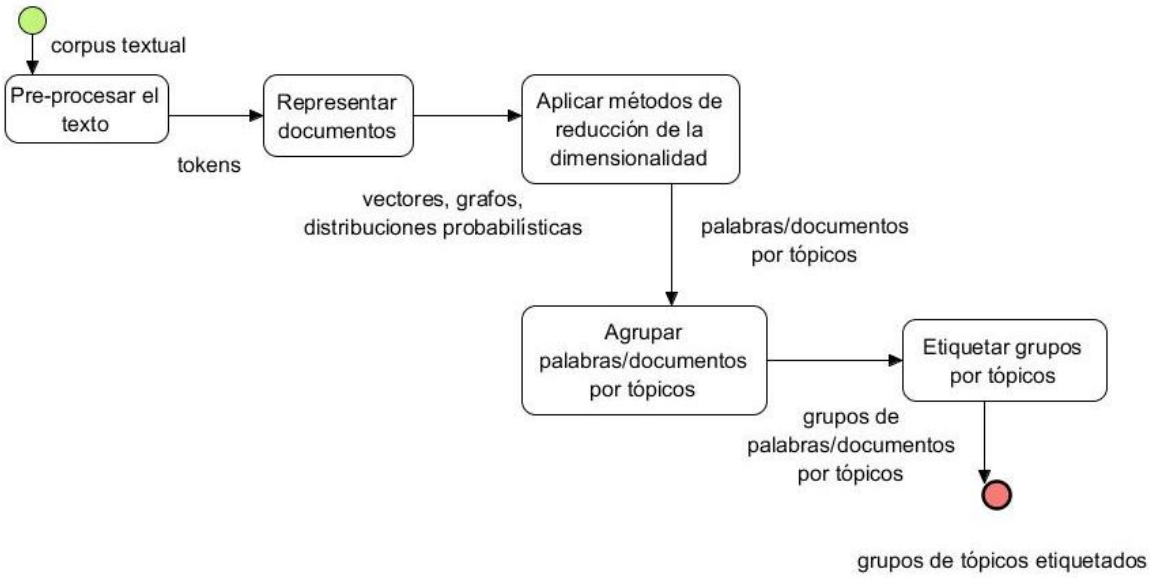
(b) Introducción del vocabulario: número de palabras que ocurren por primera vez dentro del intervalo centrado en el espacio de la oración.

(c) Cadenas: el número de cadenas activas o términos que se repiten dentro del umbral de las oraciones y amplían el espacio de las oraciones.

## Anexo 8 Resumen de las principales etapas para la segmentación por tópicos y la detección de tópicos.

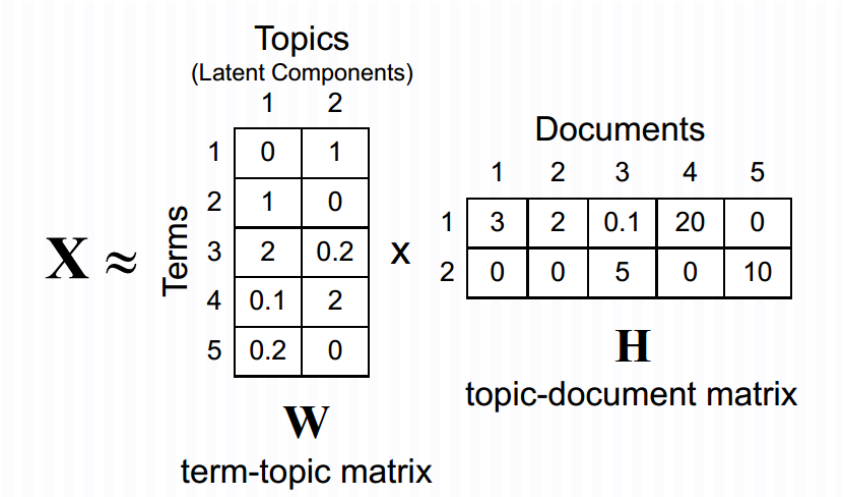



(a) Principales tareas identificadas en el proceso de segmentación.




(b) Principales tareas identificadas en el proceso de detección.

**Anexo 9 Representación del método de NMF. Fuentes:** (Carranza 2014) (Yan et al. 2013)**.**




NMF descompone la matriz de términos-documento en dos matrices no negativas de bajo orden. Primeramente un matriz de término-tópico donde cada columna representa un tópico como una combinación de términos, y luego una matriz de tópico-documento donde cada columna representa un documento como una combinación convexa de tópicos.

## Anexo 10 Salidas de los algoritmos de segmentación TextTiling y C99 para una opinión.

| TextTiling | C99 |
|---|---|
| ==========We stayed here from Nov .30 to Dec 2 and had a wonderful time .The hotel is just beautiful and the service was excellent from check in to the maid staff to the bartenders in Kitty O ' Shea ' s .<br>==========We had a room with a king bed that was very comfortable and had very nice feather pillows .<br>==========You can request other types if you have a problem with feathers .<br>==========The large flat screen TV was very nice .<br>==========Bath products were by Crabtree & Evelyn and included shampoo , conditioner , mouthwash and bady lotion .Plenty of coffee was provided .<br>==========We drank and had snacks at Kitty O ' Shea ' s .<br>==========The crowd was fun and there was live Irish folk music each night .<br>==========Good selection of beer and good Shepards Pie , Cheese Dip and Crisps !The location worked very well for us .<br>==========We walked to the Art Museum and Buddy Guys Blues Club is right across the street .<br>==========And shopping was a breeze using the shuttle !The cab ride to the Museum of Science and Industry was quite far though and cost about $15 each way .An excellent weekend getaway !========== | ==========We stayed here from Nov . 30 to Dec 2 and had a wonderful time . The hotel is just beautiful and the service was excellent from check in to the maid staff to the bartenders in Kitty O ' Shea ' s . We had a room with a king bed that was very comfortable and had very nice feather pillows . You can request other types if you have a problem with feathers . The large flat screen TV was very nice . Bath products were by Crabtree & Evelyn and included shampoo , conditioner , mouthwash and bady lotion . Plenty of coffee was provided . We drank and had snacks at Kitty O ' Shea ' s .<br>==========The crowd was fun and there was live Irish folk music each night . Good selection of beer and good Shepards Pie , Cheese Dip and Crisps ! The location worked very well for us . We walked to the Art Museum and Buddy Guys Blues Club is right across the street . And shopping was a breeze using the shuttle ! The cab ride to the Museum of Science and Industry was quite far though and cost about $15 each way . An excellent weekend getaway ! ========== |

**Anexo 11 Descripción de medidas que permiten hallar similitud semántica. Fuente:** (Pedersen & Michelizzi 2004)**.**

| Medida | Basado en | Clasificación semántica | Características |
|---|---|---|---|
| **HirstStOnge** | Cadenas léxicas | Relación | Se basa en encontrar cadenas léxicas enlazando dos sentidos de las palabras. Expresa que dos conceptos léxicos están semánticamente cerca si los synset[59] de WordNet están conectados por un camino que no es muy largo y no cambia de dirección muy a menudo. |
| **Lesk** | Diccionario | Relación | Encuentra superposiciones en los comentarios de dos synsets. La puntuación de relación es la suma de los cuadrados de los tamaños superpuestos. Lesk propuso que la relación entre dos palabras es proporcional a la extensión de las superposiciones de las definiciones de su diccionario. Los autores Banerjee y Pedersen extendieron esta noción al uso de WordNet como el diccionario para las definiciones de palabras. De esta forma utilizan la jerarquía de relaciones semánticas de WordNet. |
| **JiangConrath** | Información de contenido | Similitud | Toma la diferencia de la suma (del contenido de la información de dos conceptos A y B) y el contenido de la información de la mínima clase común[60]. Usa la noción de contenido de la información, pero en la forma de probabilidad condicional de encontrar una instancia de un synset-hijo dada una instancia de un synset-padre. |
| **Lin** | Información de contenido | Similitud | Incrementa el contenido de la información de la mínima clase común con la suma del contenido de la información de los conceptos A y B. |
| **Resnik** | Información de contenido | Similitud | El valor de Resnik es igual al contenido de la información de la mínima clase común, la clase más informativa. Esto significa que el valor será siempre mayor que o igual a 0. El límite superior en el valor es generalmente grande y varía en dependencia del tamaño del corpus usado para determinar valores de contenido de la información. De esta forma, define la similitud entre dos synset como el contenido de la información de la mínima clase común. |
| **LeacockChodorow** | Tamaño del camino entre dos conceptos | Similitud | Devuelve una puntuación que nombra cuán similar son dos sentidos de palabras, basados en el camino más corto que conecta los sentidos y la máxima profundidad de la taxonomía en la que los sentidos ocurren. |
| **Path** | Tamaño del camino entre dos conceptos | Similitud | Devuelve una puntuación que nombra cuán similar son dos sentidos de palabras, basadas en el camino más corto que conecta los sentidos en la taxonomía |

[59] Conjunto de sinónimos, representados por sustantivos, adjetivos, verbos y adverbios. Cada conjunto expresa un concepto distinto. Los synset están interconectados por medio de relaciones léxicas y semánticas-conceptuales. http://wordnet.princeton.edu/

[60] En inglés last common subsumer, se refiere al mínimo concepto común que incluye a dos conceptos en una taxonomía "es-un", por ejemplo: "nickel" y "dime" tienen como concepto común "coin", el cual es hijo del concepto "cash", y "cash" es hijo del concepto "money" (Resnik 1995). Es el concepto más específico que comparten como ancestro dos conceptos (Pedersen & Michelizzi 2004).

| | | | |
|---|---|---|---|
| | | | "es-un" (hypernym/hypnoym). Es igual al inverso del camino más corto entre dos conceptos. |
| **WuPalmer** | Tamaño del camino entre dos conceptos | Similitud | Devuelve una puntuación que nombra cuán similares son dos sentidos de palabras basados en la profundidad de los dos sentidos en la taxonomía y en su mínima clase común. |

## Anexo 12 Características tomadas de los métodos de desarrollo Scrum y Kanban.

| Característica | Método |
|---|---|
| Tiempo de las iteraciones es opcional | Kanban |
| Compromiso de trabajo por iteración | Scrum |
| Métrica por defecto velocidad | Scrum |
| Equipo multifuncional | Scrum |
| Las funcionalidades se dividen por sprint | Scrum |
| La limitación del trabajo en progreso es por el estado del trabajo | Kanban |
| Se pueden adicionar tareas | Kanban |
| Dos roles (dueño del producto- UCLV y equipo- 1 persona) | Scrum |
| Pila de producto priorizada | Scrum |

## Anexo 13 Pila de funcionalidades principales de OpinionTopicDetection.

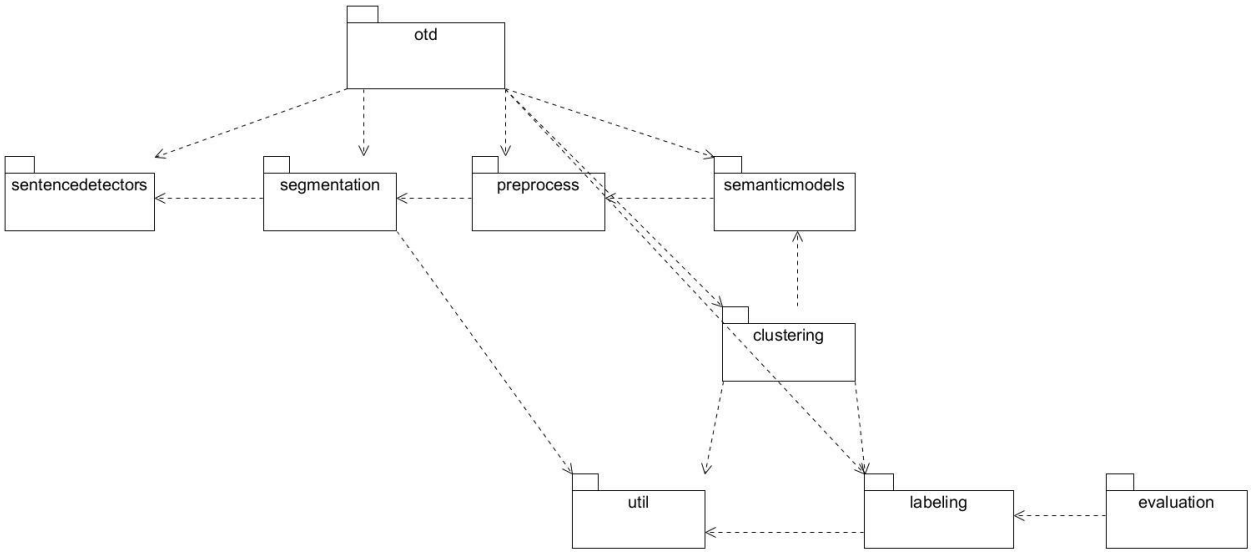
| Id | Funcionalidad | Estimación inicial[61] |
|---|---|---|
| | **Sprint:** Análisis de herramientas | |
| 1 | Estudio de herramientas de segmentación | 8 |
| 2 | Estudio de Lucene | 6 |
| 3 | Estudio de OpenNLP | 4 |
| 4 | Estudio de Stanford | 6 |
| 5 | Estudio de Mahout | 10 |
| 6 | Estudio de LingPipe | 5 |
| 7 | Estudio de Mallet | 5 |
| 8 | Estudio de SemanticVectors | 8 |
| 9 | Estudio de S-Space | 10 |
| 10 | Estudio del TopicDetectionFramework | 6 |
| 11 | Estudio de Carrot | 3 |
| 12 | Estudio de Solr | 3 |
| | **Sprint:** Segmentación | |
| 13 | Detectar oraciones con OpenNLP | 2 |
| 14 | Detectar oraciones con Stanford | 4 |
| 15 | Segmentación por tópicos con C99 | 12 |
| 16 | Segmentación por tópicos con TextTiling | 12 |
| | **Sprint:** Pre-procesamiento | |
| 17 | Análisis léxico con Lucene | 6 |
| 18 | Lematizar términos (TreeTagger y Stanford) | 5 |
| 19 | Corrector ortográfico | 6 |
| 20 | Eliminar palabras vacías | 0.5 |
| 21 | Convertir palabras a minúscula | 0.5 |
| | **Sprint:** Representación textual | |
| 22 | Representaciones textuales con S-Space para VSM y LSA | 12 |
| | **Sprint:** Agrupamiento | |
| 23 | Agrupar segmentos con algoritmos HAC de S-Space (dos variantes, dendrograma y umbral) | 12 |
| 24 | Agrupar segmentos con algoritmos Estrella de RST-DisambiguationLib | 6 |
| | **Sprint:** Etiquetamiento | |
| 25 | Identificar partes del discurso por cada término de los tópicos con TreeTagger | 2 |
| 26 | Identificar partes del discurso por cada término de los tópicos con OpenNLP | 2 |
| 27 | Hallar sustantivos más representativos por cada tópico (WS4J) | 6 |
| | **Sprint:** Integración | |
| 28 | Integración de funcionalidades para descubrir tópicos locales | 12 |
| 29 | Integración de funcionalidades para descubrir tópicos globales | 12 |
| 30 | Asociar categorías a los tópicos para evaluación | 6 |
| | **Sprint:** Evaluación | |
| 31 | Medidas externas para la evaluación | 4 |
| 32 | Preparar ficheros de validación para el agrupamiento con medidas internas | 6 |
| | **Sprint:** Útiles | |
| 33 | Detectar frases sustantivas | 4 |
| 34 | Crear índices con Lucene | 4 |
| 35 | Visualizar dendrogramas | 8 |
| 36 | Documentación javadoc | 6 |
| 37 | Aplicación desktop OpinionTD | 8 |

[61] Puntos de historia-horas/persona.

## Anexo 14 Ejemplo de un tablero de tareas para el sprint de Segmentación[62].

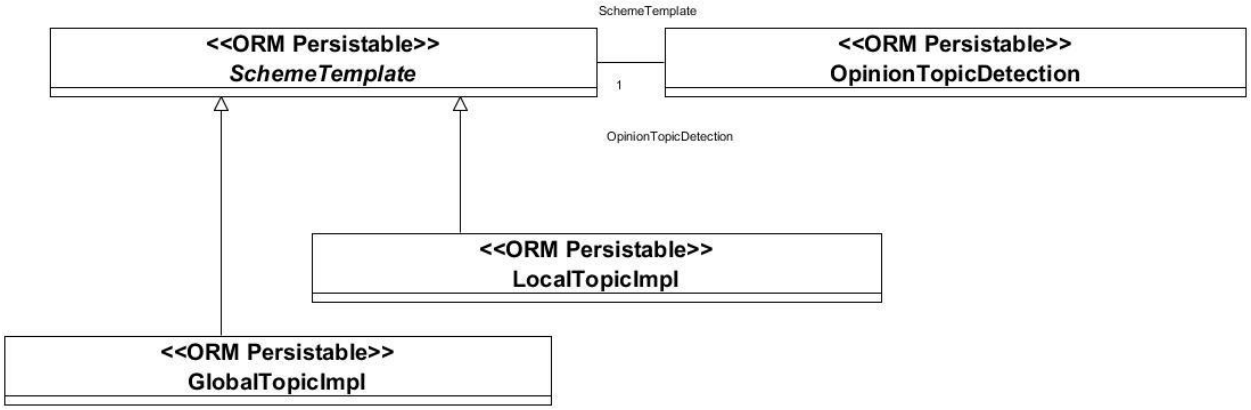
| **Pendiente** | **En desarrollo** | **Evaluando** | **Integrado** |
|---|---|---|---|
| Detectar oraciones con OpenNLP | Detectar oraciones con OpenNLP | Detectar oraciones con OpenNLP | Detectar oraciones |
| Detectar oraciones con Stanford | Detectar oraciones con Stanford | Detectar oraciones con Stanford | Detectar oraciones con Stanford |
| Segmentación por tópicos con C99 | Segmentación por tópicos con C99 | Segmentación por tópicos con C99 | |
| Segmentación por tópicos con TextTiling | Segmentación por tópicos con TextTiling | | |

## Anexo 15 Diagrama de paquetes del marco de trabajo OpinionTopicDetection.

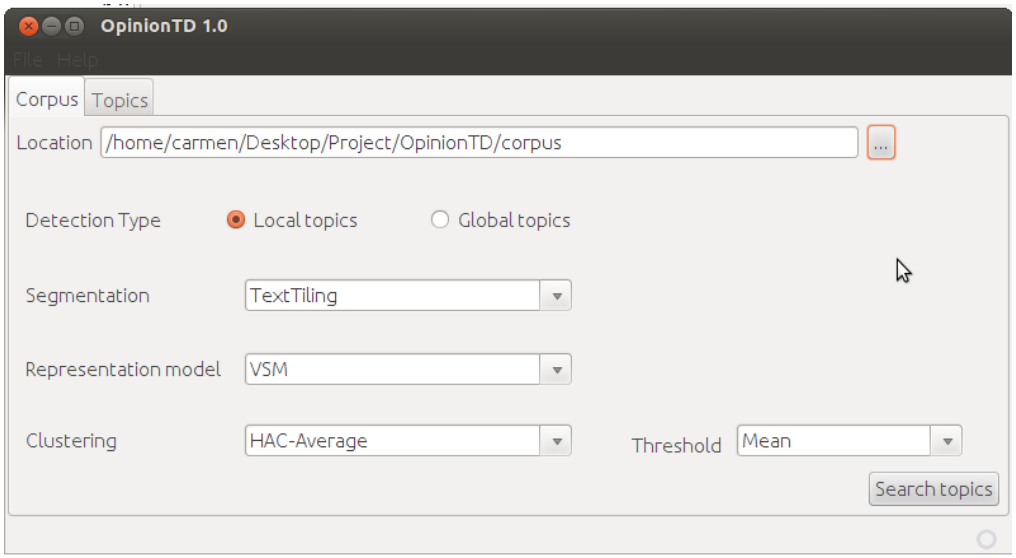



[62] Los textos en color negro significan que es la fase en la que está en ese momento y en gris la fase por la que ya pasó.

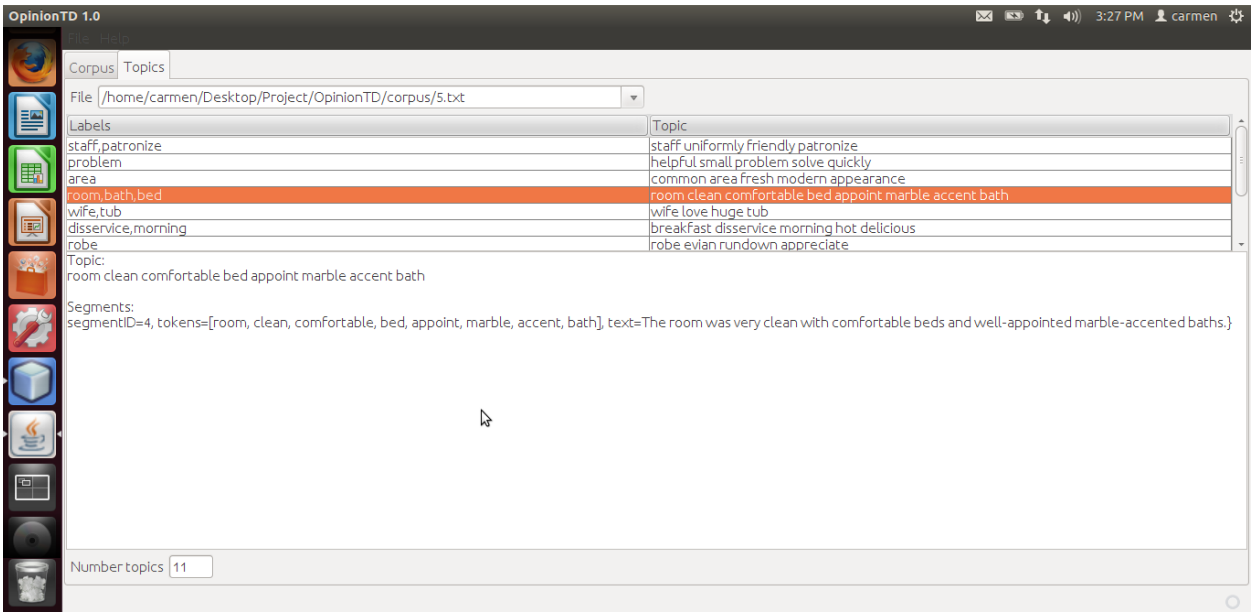
**Anexo 16 Diagrama de clases principales del paquete otd en OpinionTopicDetection.**




**Anexo 17 Interfaz de la aplicación OpinionTD.**



## Anexo 18 Interfaz de la aplicación que muestra los resultados de tópicos locales.




## Anexo 19 Tópicos encontrados para tres opiniones con HAC-Complete y umbral medio.

| Opiniones | Tópicos | Total |
|---|---|---|
| A | Label--- river,lake,customer<br>Topic--- [great, place, view, river, lake, walk, clean, comfortable, room, accommodate, staff, arrive, early, check, late, effectively, add, day, vacation, large, hotel, work, hard, take, good, care, customer]<br>Segments---<br>segmentID=0, tokens=[great, place, great, view, river, lake, walk, clean, comfortable, room, accommodate, staff], text=This was a great place to be! Great views of river and lake, walk to everything, a clean and comfortable room, and very accommodating staff.}<br>segmentID=1, tokens=[arrive, early, check, check, late, effectively, add, day, vacation], text=We arrived early and were checked in by 10:30 am and checked out late , effectively adding two days to our vacation.}<br>segmentID=2, tokens=[large, hotel, staff, work, hard, take, good, care, customer], text=It is a large hotel but the staff works hard and takes good care of the customers.} | 1 |
| B | Label--- screen,tv<br>Topic--- [large, flat, screen, tv, nice]<br>Segments---<br>segmentID=4, tokens=[large, flat, screen, tv, nice], text=The large flat screen TV was very nice.} | 12 |
| | Label--- bath,shampoo,lotion<br>Topic--- [bath, product, crabtree, evelyn, include, shampoo, conditioner, mouthwash, baby, lotion]<br>Segments---<br>segmentID=5, tokens=[bath, product, crabtree, evelyn, include, shampoo, conditioner, mouthwash, baby, lotion], text=Bath products were by Crabtree & Evelyn and included shampoo, conditioner, mouthwash and bady lotion.} | |
| | … | |
| | Label--- bed,wifi,bedroom | |

| | | |
|---|---|---|
| C | Topic--- [super, comfy, bed, ensure, good, snooze, complimentary, wifi, allow, plan, day, activity, bedroom]<br>Segments---<br>segmentID=6, tokens=[super, comfy, bed, ensure, good, snooze, complimentary, wifi, allow, plan, day, activity, bedroom], text=The super comfy bed ensured a good snooze and complimentary WiFi allowed us to plan our next day's activities from our bedroom.} | 10 |
| | Label--- mix,bonito<br>Topic--- [barman, mix, bonito]<br>Segments---<br>segmentID=10, tokens=[barman, mix, bonito], text=The barman mixes a mean mohito !} | |
| | Label--- return,check,alternative<br>Topic--- [return, week, james, chicago, hotel, fabulous, chic, chill, service, compare, check, swift, painless, concierge, super, efficient, locate, excellent, hair, salon, couple, fab, restaurant, stay, rate, top, vote, resident, visit, alternative, maintain, standard]<br>Segments---<br>segmentID=0, tokens=[return, week, james, chicago, hotel, fabulous, chic, chill, service, compare, check, swift, painless, concierge, service, super, efficient, locate, excellent, hair, salon, couple, fab, restaurant, stay, hotel, restaurant, rate, top, chicago, vote, chicago, resident], text=We have just returned from a week at the James Chicago.The hotel is fabulous, very chic and chilled.The service is beyond compare.Check in and out were swift and painless.The Concierge service is also super efficient having located an excellent hair salon and a couple of fab restaurants for us during our stay.The hotel restaurant itself is rated 5 in the top 10 in Chicago - voted by Chicago residents.}<br>segmentID=14, tokens=[visit, chicago, stay, alternative, james, maintain, standard], text=Next visit to Chicago I'll definitely be staying here and wouldn't even consider an alternative if The James can maintain these standards.} | |
| | … | |

## Anexo 20 Valores promedios obtenidos por las medidas de validación interna para tópicos locales para algoritmos HAC.

| Opiniones Largas | | | | | |
|---|---|---|---|---|---|
| | Silhouette | Dunn | R-Squared | Homogeneity | Separation |
| HAC AVERAGE | 0.388645 | 0.17 | 0.9854 | 0.6437 | 0.6694 |
| HAC COMPLETE | 0.4066 | 0.11 | 0.9953 | 0.6609 | 0.6 |
| HAC SINGLE | 0.40328 | 0.2292 | 0.7728 | 0.574 | 0.705 |
| Opiniones Medianas | | | | | |
| | Silhouette | Dunn | R-Squared | Homogeneity | Separation |
| HAC AVERAGE | 0.451 | 0.2944 | 0.718 | 0.7568 | 0.7844 |
| HAC COMPLETE | 0.4997 | 1.082 | 0.815 | 0.7518 | 0.7704 |
| HAC SINGLE | 0.4086 | 0.2208 | 0.5839 | 0.7568 | 0.7982 |
| Opiniones Pequeñas | | | | | |
| | Silhouette | Dunn | R-Squared | Homogeneity | Separation |
| HAC AVERAGE | 0.7645 | 0.7258 | 0.8265 | 0.8509 | 0.8576 |
| HAC COMPLETE | 0.7645 | 0.1055 | 0.7806 | 0.8556 | 0.8655 |
| HAC SINGLE | 0.7212 | 0.7258 | 0.7325 | 0.8592 | 0.8656 |

## Anexo 21 Valores promedios obtenidos por las medidas de validación interna para tópicos locales para variantes del algoritmo Estrella.

| Opiniones Largas | | | | | |
|---|---|---|---|---|---|
| | Silhouette | Dunn | R-Squared | Homogeneity | Separation |
| STAR EXTENDED | 0.3082 | 0.437 | 0.9058 | 0.6188 | 0.623 |
| STAR GENERALIZED | 0.2751 | 0.4941 | 0.6882 | 0.6018 | 0.632 |
| STAR CONDENSED | 0.2543 | 0.4244 | 0.8562 | 0.577 | 0.6235 |
| Opiniones Medianas | | | | | |
| | Silhouette | Dunn | R-Squared | Homogeneity | Separation |
| STAR EXTENDED | 0.4616 | 0.5189 | 0.6451 | 0.752 | 0.7789 |
| STAR GENERALIZED | 0.4616 | 0.5189 | 0.6073 | 0.752 | 0.7811 |
| STAR CONDENSED | 0.5244 | 1.09 | 0.677 | 0.8591 | 0.8394 |
| Opiniones Pequeñas | | | | | |
| | Silhouette | Dunn | R-Squared | Homogeneity | Separation |
| STAR EXTENDED | 0.7178 | 0.7258 | 0.7376 | 0.8515 | 0.8589 |
| STAR GENERALIZED | 0.7212 | 0.7258 | 0.7325 | 0.8592 | 0.8656 |
| STAR CONDENSED | 0.7314 | 0.7258 | 0.7415 | 0.8592 | 0.864 |

## Anexo 22 Umbrales encontrados con la prueba de Friedman para corpus de opiniones largas.

a) Silhouette

| **Algoritmo** | **Umbral seleccionado** | **Diferencias significativas** |
|---|---|---|
| HAC AVERAGE | max | - |
| HAC COMPLETE | max | - |
| HAC SINGLE | max | - |
| STAR EXTENDED | max | - |
| STAR GENERALIZED | max | - |
| STAR CONDENSED | max | 0.008544222130039691 |

b) RSquared

| **Algoritmo** | **Umbral seleccionado** | **Diferencias significativas** |
|---|---|---|
| HAC AVERAGE | max | - |
| HAC COMPLETE | max | 0.01233580329866013 |
| HAC SINGLE | max | - |
| STAR EXTENDED | max | - |
| STAR GENERALIZED | max | - |
| STAR CONDENSED | max | - |

c) Homogeneity

| **Algoritmo** | **Umbral seleccionado** | **Diferencias significativas** |
|---|---|---|
| HAC AVERAGE | max | - |
| HAC COMPLETE | max | - |
| HAC SINGLE | min | - |
| STAR EXTENDED | max | 0.008784631848763325 |
| STAR GENERALIZED | max | 0.04029259917491146 |
| STAR CONDENSED | max | - |

d) Separation

| **Algoritmo** | **Umbral seleccionado** | **Diferencias significativas** |
|---|---|---|
| HAC AVERAGE | mean | - |
| HAC COMPLETE | mean | 0.002341407721621058 |
| HAC SINGLE | mean | - |
| STAR EXTENDED | mean | - |
| STAR GENERALIZED | min | 0.028310162556166474 |
| STAR CONDENSED | comb | - |

## Anexo 23 Ejemplo de análisis estadístico realizado con la medida Separation aplicada al algoritmo HAC-Complete en opiniones largas para identificar umbrales.

a) Valores obtenidos con la prueba de Friedman

| Algorithm | Ranking |
|---|---|
| min | 2.45 |
| mean | 1.75 |
| comb | 2.5 |
| max | 3.3 |

Table 1: Average Rankings of the algorithms

b) Valores de p-value obtenidos con los métodos post-hoc

| $i$ | algorithms | $z = (R_0 - R_i)/SE$ | $p$ | Holm | Shaffer |
|---|---|---|---|---|---|
| 6 | mean vs. max | 3.796709 | 0.000147 | 0.008333 | 0.008333 |
| 5 | min vs. max | 2.082066 | 0.037336 | 0.01 | 0.016667 |
| 4 | comb vs. max | 1.959592 | 0.050044 | 0.0125 | 0.016667 |
| 3 | mean vs. comb | 1.837117 | 0.066193 | 0.016667 | 0.016667 |
| 2 | min vs. mean | 1.714643 | 0.086411 | 0.025 | 0.025 |
| 1 | min vs. comb | 0.122474 | 0.902523 | 0.05 | 0.05 |

Table 2: P-values Table for $\alpha = 0.05$

En este caso se selecciona el umbral medio (mean) como el umbral que presenta mayor diferencia significativa de acuerdo a los valores de los rankings obtenido por la prueba de Friedman a) y la comparación con el método de Holm del par mean vs max en b).

## Anexo 24 Datos de entrada para realizar la prueba estadística de Friedman con el objetivo de buscar diferencias significativas entre los algoritmos de agrupamiento.

| opiniones | HAC AVERAGE (max) | HAC COMPLETE (max) | HAC SINGLE (max) | STAR EXTENDED (max) | STAR GENERALIZED (max) | STAR CONDENSED (max) |
|---|---|---|---|---|---|---|
| camera1 | 0.6386 | 0.6386 | 0.6058 | 0.6052 | 0.6052 | 0.6185 |
| camera2 | 0.7054 | 0.7054 | 0.6358 | 0.7023 | 0.7023 | 0.659 |
| camera3 | 0.654 | 0.6421 | 0.5836 | 0.5971 | 0.5971 | 0.5671 |
| camera4 | 0.6463 | 0.6507 | 0.4823 | 0.5494 | 0.5245 | 0.5096 |
| camera5 | 0.7236 | 0.7236 | 0.659 | 0.6857 | 0.6819 | 0.6998 |
| hotel1 | 0.6469 | 0.6985 | 0.5895 | 0.705 | 0.6426 | 0.5625 |
| hotel2 | 0.6315 | 0.6552 | 0.5753 | 0.6207 | 0.6033 | 0.5839 |
| hotel3 | 0.6429 | 0.6813 | 0.6117 | 0.6429 | 0.6502 | 0.5856 |
| hotel4 | 0.4877 | 0.5941 | 0.4069 | 0.5111 | 0.4746 | 0.414 |
| hotel5 | 0.7202 | 0.7202 | 0.6132 | 0.6101 | 0.6101 | 0.6794 |
| restaurant1 | 0.4943 | 0.4843 | 0.3602 | 0.4012 | 0.4012 | 0.3163 |
| restaurant2 | 0.7026 | 0.7026 | 0.575 | 0.6448 | 0.6245 | 0.6126 |
| restaurant3 | 0.5926 | 0.6352 | 0.54 | 0.5432 | 0.5432 | 0.5031 |
| restaurant4 | 0.6256 | 0.6256 | 0.4506 | 0.5795 | 0.5312 | 0.39 |

| | | | | | | |
|---|---|---|---|---|---|---|
| restaurant5 | 0.5876 | 0.6088 | 0.5114 | 0.5734 | 0.5734 | 0.5162 |
| mobilephone 1 | 0.7404 | 0.7404 | 0.7214 | 0.7214 | 0.7214 | 0.7017 |
| mobilephone 2 | 0.6892 | 0.6892 | 0.7198 | 0.7243 | 0.7342 | 0.7255 |
| mobilephone 3 | 0.5737 | 0.5737 | 0.3967 | 0.4981 | 0.4503 | 0.4216 |
| mobilephone 4 | 0.6989 | 0.7238 | 0.6815 | 0.7246 | 0.6925 | 0.6874 |
| mobilephone 5 | 0.6654 | 0.7264 | 0.6076 | 0.7362 | 0.6739 | 0.6076 |

## Anexo 25 Ejemplo de análisis estadístico realizado con la medida Homogeneidad para comparar algoritmos de agrupamiento en opiniones largas.

a) Valores obtenidos con la prueba de Friedman

| Algorithm | Ranking |
|---|---|
| HAC AVG (max) | 2.2 |
| HAC COMP (max) | 1.675 |
| HAC SINGLE (max) | 5.275 |
| STAR EXT (max) | 3.15 |
| STAR GEN (max) | 3.775 |
| STAR COND (max) | 4.925 |

Table 1: Average Rankings of the algorithms

a) Valores de p-value obtenidos con los métodos post-hoc

| $i$ | algorithms | $z = (R_0 - R_i)/SE$ | $p$ | Holm | Shaffer |
|---|---|---|---|---|---|
| 15 | HAC COMP (max) vs. HAC SINGLE (max) | 6.085111 | 0 | 0.003333 | 0.003333 |
| 14 | HAC COMP (max) vs. STAR COND (max) | 5.493503 | 0 | 0.003571 | 0.005 |
| 13 | HAC AVG (max) vs. HAC SINGLE (max) | 5.197699 | 0 | 0.003846 | 0.005 |
| 12 | HAC AVG (max) vs. STAR COND (max) | 4.606091 | 0.000004 | 0.004167 | 0.005 |
| 11 | HAC SINGLE (max) vs. STAR EXT (max) | 3.591906 | 0.000328 | 0.004545 | 0.005 |
| 10 | HAC COMP (max) vs. STAR GEN (max) | 3.549648 | 0.000386 | 0.005 | 0.005 |
| 9 | STAR EXT (max) vs. STAR COND (max) | 3.000298 | 0.002697 | 0.005556 | 0.007143 |
| 8 | HAC AVG (max) vs. STAR GEN (max) | 2.662236 | 0.007762 | 0.00625 | 0.007143 |
| 7 | HAC SINGLE (max) vs. STAR GEN (max) | 2.535463 | 0.01123 | 0.007143 | 0.007143 |
| 6 | HAC COMP (max) vs. STAR EXT (max) | 2.493205 | 0.01266 | 0.008333 | 0.008333 |
| 5 | STAR GEN (max) vs. STAR COND (max) | 1.943855 | 0.051913 | 0.01 | 0.01 |
| 4 | HAC AVG (max) vs. STAR EXT (max) | 1.605793 | 0.108319 | 0.0125 | 0.0125 |
| 3 | STAR EXT (max) vs. STAR GEN (max) | 1.056443 | 0.290766 | 0.016667 | 0.016667 |
| 2 | HAC AVG (max) vs. HAC COMP (max) | 0.887412 | 0.374857 | 0.025 | 0.025 |
| 1 | HAC SINGLE (max) vs. STAR COND (max) | 0.591608 | 0.554113 | 0.05 | 0.05 |

Table 2: P-values Table for $\alpha = 0.05$

En este caso no hubo diferencias significativas porque Friedman obtuvo valor de 7.63. Por ello se seleccionó el algoritmo con menor ranking y a la vez donde el p-value fuera menor, en este caso HAC-Complete con umbral máximo.

## Anexo 26 Segmentos obtenidos con LSA y VSM para representar tópicos globales.

<table>
<tr><th>LSA</th><th>VSM</th></tr>
<tr><td rowspan="3">Label--- hotel,location<br>Topic--- [hotel, monaco, great, location, service, centrally, locate, excellent, magnificent]<br>Segments---<br>segmentID=1, path=Corpora Global/LARA-Hotel/124.txt, tokens=[hotel, monaco], text=The Hotel Monaco is the best yet.<br><br>segmentID=0, path=Corpora Global/LARA-Hotel/123.txt, tokens=[hotel, monaco, great, location, service, hotel, monaco, centrally, locate, excellent, service], text=Hotel Monaco - Great Location and Service The Hotel Monaco is centrally located and provides excellent service.<br><br>segmentID=0, path=Corpora Global/LARA-Hotel/110.txt, tokens=[magnificent, hotel, monaco], text=Magnificent Hotel Monaco!</td><td>Label--- night,weekend<br>Topic--- [pleasant, stay, seattle, monaco, night, weekend, getaway, hitch, hotel, fan, back]<br>Segments---<br>segmentID=0, path=Corpora Global/LARA-Hotel/18.txt, tokens=[pleasant, stay, seattle, monaco, night, weekend, getaway, seattle, monaco, hitch], text=Pleasant stay at the Seattle Monaco Our 2 night weekend getaway at the Seattle Monaco went off without a hitch.<br><br>segmentID=1, path=Corpora Global/LARA-Hotel/124.txt, tokens=[hotel, monaco], text=The Hotel Monaco is the best yet.<br><br>segmentID=0, path=Corpora Global/LARA-Hotel/20.txt, tokens=[fan, hotel, monaco, seattle, back, weekend, hotel, monaco, seattle], text=Fan of the Hotel Monaco Seattle Just got back from our weekend at the Hotel Monaco Seattle.</td></tr>
<tr><td>Label--- hotel<br>Topic--- [magnificent, hotel, monaco]<br>Segments---<br>segmentID=0, path=Corpora Global/LARA-Hotel/110.txt, tokens=[magnificent, hotel, monaco], text=Magnificent Hotel Monaco!</td></tr>
<tr><td>Label--- room,hotel,location<br>Topic--- [excellent, stay, room, service, prompt, food, hotel, monaco, great, location, centrally, locate]<br>Segments---<br>segmentID=0, path=Corpora Global/LARA-Hotel/115.txt, tokens=[excellent, stay], text=Excellent Stay!<br><br>segmentID=5, path=Corpora Global/LARA-Hotel/124.txt, tokens=[room, service, prompt, food, excellent], text=Room service was prompt and the food was excellent.<br><br>segmentID=0, path=Corpora Global/LARA-Hotel/123.txt, tokens=[hotel, monaco, great, location, service, hotel, monaco, centrally, locate, excellent, service], text=Hotel Monaco - Great Location and Service The Hotel Monaco is centrally located and provides excellent service.</td></tr>
</table>