Universidad de las Ciencias Informáticas
Facultad 3

Título: Valoración de técnicas de minería de texto para la detección de tópicos.

Trabajo de Diploma para optar por el título de

Ingeniero en ciencias Informáticas

Autor(es): Geidy Medina Rodríguez
Nereyda M. Rojas Rodríguez

Tutor: Dr. Ing. Ernesto González Díaz

Consultante: Ing. Ernesto Guevara Martínez

Ciudad Habana, Junio 2007

“La ciencia tiene una característica maravillosa, y es que aprende de sus errores, que utiliza sus equivocaciones para reexaminar los problemas y volver intentar resolverlos, cada vez por nuevos caminos”

Ruy Pérez Tamayo

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al <nombre área> de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

"[Insertar nombre(s) de autor(es)]"

_____

"[Insertar nombre(s) de tutor(es)]"

_____

DATOS DE CONTACTO

FACULTAD DE INGENIERÍA INDUSTRIAL	DEPARTAMENTO O CENTRO CEIS	TELEFONO	CORREO ELECTRONICO
NOMBRES Y APELLIDOS Ernesto González Díaz	DIRECCIÓN PARTICULAR Lagueruela 125, Sevillano La Habana, Cuba	TELEFONOS 41-7121 266-3905	CORREO ELECTRONICO egonzalez@ceis.cujae.edu.cu
No CARNE DE IDENTIDAD 72052913724	SEXO M	CENTRO DE TRABAJO CEIS CUJAE	OCUPACIÓN ACTUAL Profesor
RAMADE LA CIENCIA, ESPECIALIDAD Informática			
GRADO CIENTÍFICO Y FECHA EN QUE LO OBTUVO Master Ciencias 2001	OTROS TITULOS ACADEMICOS	CATEGORÍA DOCENTE Asistente	CATEGORÍA CIENTIFICA
EXPERIENCIA PROFESIONAL EN LA EDUCACIÓN SUPERIOR____11__años____ EN LA INVESTIGACIÓN CIENTÍFICA___11_años____ ENLA PRODUCCIÓN Y LOS SERVICIOS___2__años____			
CONDECORACIONES Y DISTINCIONES RECIBIDAS			

RESULTADOS RELEVANTES ALCANZADOS EN LA DOCENCIA DE PREGRADO Y POSGRADO Tutor de varios trabajos de Curso y de Diploma que han sido premiados en Jornadas Científicas Estudiantiles Participación como profesor en el curso de Formación de Profesores de Computación Básica para la Enseñanza Primaria. Profesor y Jefe de Carrera de Ingeniería Informática en la Universalización de la Enseñanza Superior en el Municipio Arroyo Naranjo. Cursos 2002-2003 y 2003- 2004. En los cursos 2001-2002 y 2002-2003 recibió la calificación de Excelente en la evaluación profesional. Profesor Principal de Inteligencia Artificial en la Universidad de las Ciencias Informáticas en el curso 2004-2005 En la actualidad se desempeña como coordinador de la Maestría Nuevas Tecnologías de la informática y las comunicaciones para la Educación
PREMIOS RECIBIDOS Mención en el XII Fórum de Ciencia y Técnica Municipio Arroyo Naranjo Mención Especial en el XIII Fórum de Ciencia y Técnica Municipio Arroyo Naranjo. 24 Concurso Científico Técnico Juvenil de las BTJ.
PARTICIPACIÓN ACTUAL Y EN LOS ULTIMOS CINCO AÑOS EN PROYECTOS DE INVESTIGACIÓN Proyecto de Informática Educativa en las Carreras de la Educación Técnica y Profesional. Específicamente la Carrera de Construcción Civil. Red Temática Internacional para la Enseñanza de la Informática Gráfica y el Diseño Asistido por Computadora en la Arquitectura. Informatización de la Carrera Ingeniería Informática para la Universalización. Aplicación de la Inteligencia Artificial en el software educativo. (Actual)

RESULTADOS RELEVANTES ALCANZADOS EN LA INVESTIGACIÓN

Proyecto de Informática Educativa para la Disciplina Obras de Ingeniería Civil en la Carrera Construcción del ISPETP

Software para la enseñanza de la asignatura Geotecnia en la Carrera Ingeniería Civil.

Sistema de Mapas Conceptuales digitalizados para la enseñanza del medio ambiente.

Modelación de los sistemas de evaluación de los sorteares educativos mediante redes bayesianas.

Experiencia profesional

Ha impartido las asignaturas de Computación I, II, III y IV en la carrera de Arquitectura, en la carrera Ingeniería informática CUJAE ha impartido las siguientes asignaturas:

Introducción a la Informática
Lógica y Algoritmos
Programación I, II y IV
Inteligencia Artificial I, III
Sistemas Operativos
Sistemas Informáticos Inteligentes II
Máquinas Computadoras I

En la Universidad de las Ciencias Informáticas impartió las siguientes asignaturas:

Inteligencia Artificial
Programación IV, I, III

Y los cursos optativos:

Programación Descriptiva
Lógica Matemática
Sistemas Basados en el Conocimiento
Sistemas Adaptativos
Lenguajes y Técnicas de Programación

Ha impartido cursos de postrado de las siguientes temáticas

Programación Orientada a Objetos (CUJAE)
Inteligencia Artificial (UCI)
Temas Avanzados de Inteligencia Artificial (UCI)
Informática Educativa (Instituto Superior Pedagógico para la Educación Técnica Profesional ISPETP, Universidad de la Habana)
Gestión del conocimiento y de la información (ISPETP)

En la maestría en Nuevas Tecnologías de la Informática y las Comunicaciones Aplicadas a la Educación ha impartido las asignaturas:

Ingeniería de Sistemas Educativos,
Tecnología de la Programación,
Arquitectura de Sistemas Computacionales,
Servicios Telemáticos para la Educación,
Multimedia Educativa
Sistemas de enseñanza en línea.
Temas de Inteligencia Artificial

Estas asignaturas las impartió en la CUJAE y el Centro Universitario de La Isla de la Juventud

Impartió el módulo titulado Modelación y presentación de proyectos usando AutoCAD 2D-3D en la pasantía Internacional “Rehabilitación del Barrio de Colón en La Habana”, con alumnos de la Universidad de Cuenca en Ecuador. Facultad de Arquitectura CUJAE

En la Especialidad Informática Operativa impartió en curso Fundamentos de la Investigación Científica (CUJAE).

En la Maestría Informática Aplicada que se desarrolla en la UCI, impartió la asignatura “Programación Avanzada en .NET”.

PARTICIPACIÓN COMO PONENTE EN EVENTOS CIENTÍFICOS

(Título de la ponencia, evento, fecha)

Empleo de la computación en el desarrollo de habilidades y conocimientos en la asignatura hidráulica agrícola.

Evento Nacional AGROMEC 1999.

Empleo de la computación en el desarrollo de habilidades y conocimientos en la asignatura hidráulica agrícola.

Convención Internacional de la Industrias Mecánicas y Electrónica METANICA 99.

Las Tecnologías Informáticas en el desarrollo de habilidades en la carrera Construcción Civil del ISPETP.

XI Reunión Científica de Profesores ISPETP 1999.

Propuesta de Software Educativo para la enseñanza de la asignatura Topografía de la Carrera Construcción Civil

IV Taller Nacional de Pedagogía Profesional.

Análisis Estadístico de la información climática en la región de Santiago de las Vegas La Habana Cuba.

Evento Nacional AGROMEC 2000

Sistema de tareas docentes para la inserción de la computación en la enseñanza de la Topografía, carrera Construcción Civil.

XII Fórum de Ciencia y Técnica Municipio Arroyo Naranjo. 2000

Caracterización de las variables climáticas en la región de Santiago de Las Vegas.

II Fórum Tecnológico Especial “Suelos y manejo del agua”. 2000.

Análisis Estadístico de la información climática en la región de Santiago de las Vegas La Habana Cuba.

Convención Internacional de la Industrias Mecánicas y Electrónica METANICA 2000.

La informática como medio de enseñanza y herramienta de trabajo en la Carrera Construcción del ISPETP.

II Taller Nacional sobre Didáctica Universitaria Universidad de la Habana 2000.

La Tecnología Educativa en el desarrollo de la asignatura Hidráulica Agrícola (Riego y drenaje).

Evento Nacional AGROMEC 2001.

La dimensión ambiental de la carrera Construcción Civil

III Convención Internacional sobre Medio Ambiente y Desarrollo. 2001.

Proyecto de Informática Educativa para la disciplina Obras de Ingeniería Civil en la Carrera Construcción del ISPETP.

V Taller Nacional de Pedagogía Profesional 2002.

Proyecto de Informática Educativa para la disciplina Obras de Ingeniería Civil en la Carrera Construcción del ISPETP.

IV Simposio Iberoamericano de Pedagogía Profesional. 2002

La enseñanza de la Informática Gráfica a través de la modelación espacial en la carrera Arquitectura.

III Taller Nacional sobre Didáctica Universitaria Universidad de la Habana 2002.

Aplicación de los Mapas Conceptuales en la Gestión del conocimiento. Presentación del mapa conceptual de Medio Ambiente.

Congreso Internacional CUBA-RIEGO 2003.

Sistema de Mapas Conceptuales para el aprendizaje de los conceptos fundamentales del medio ambiente.

Concurso Científico Técnico Juvenil de las BTJ 2003

Sistema de Mapas Conceptuales para la enseñanza de los conceptos fundamentales del medio ambiente.

Fórum Tecnológico Nacional de Informática Educativa La Habana 2004.

Los mapas conceptuales en el desarrollo de los procesos docentes en la educación superior. IV Taller Nacional de Didáctica de la Educación Superior Universidad de la Habana 2004

Los mapas conceptuales en el desarrollo de los procesos docentes en la educación superior. Taller Científico Metodológico para la Universalización Universidad de La Habana 2004.

Las redes bayesianas en la modelación de los módulos de evaluación de los softwares educativos. Taller Inteligencia Artificial UCICIENCIA 2005, UCI La Habana 2005

Tendencias y Tecnologías actuales en la Programación web, Proyecto web Universidad de La Habana 2005.

IV Congreso Iberoamericano de Reconocimiento de Patrones IRCAP 2005, La Habana

Noviembre 2005

Modelación del Estudiante en Software Educativo con Técnicas de Inteligencia Artificial, Taller de Inteligencia Artificial, UCIENCIA 2006,

Seminario Nacional para el Perfeccionamiento de las Sedes Municipales Universitarias del MES. Septiembre 2006

III Congreso Nacional de Reconocimiento de Patrones RECAP 2006, Sociedad Cubana de Matemática Computación. UCI Octubre 2006

OTROS DATOS QUE CONSIDERE PUEDAN RESULTAR DE INTERES.

En el año 2003 fungió como coordinador por la CUJAE de la red alfa para la enseñanza de la informática grafica y el diseño asistido por computadora. En esta red participaban tres Universidades españolas, una de Uruguay una de Chile y la CUJAE.

Ha cursado estudios de nivel medio de Ingles, Francés, portugués y alemán, y postgrados de Pedagogía Profesional, Evaluación Escolar, Metodología de la Investigación Científica, Estadística Matemática, Diseño de Experimentos, Educación y Sociedad, Informática Educativa, Tecnología Educativa, Problemas Sociales de la Ciencia y la Tecnología, Computación e Infotecnología.

En el curso 2002-2003 fungió como profesor principal de la disciplina computación en la carrera de Arquitectura.

Ha impartido conferencias especializadas sobre Informática Educativa e Inteligencia Artificial, en La Universidad de las Ciencias Informáticas, el Instituto Superior Pedagógico para la Educación Técnica y Profesional y el Instituto de Investigaciones de Riego y Drenaje.

Ha sido tribunal, tutor y oponente de mas de 35 tesis de pregrado de las carreras Ingeniería informática, Licenciatura en Educación, especialidad Construcción Civil, e Ingeniería Civil. Ha sido tribunal, tutor y oponente de mas de 10 tesis de la maestría Informática Aplicada que se desarrolla en el CEIS. Ha sido tutor y tribunal de varias tesinas de los diplomados de Informática Aplica y de la maestría en Nuevas Tecnologías de la Informática y las Comunicaciones en la Educación en la CUJAE y en el Centro Universitario de la Isla de la Juventud.

PUBLICACIONES EN LOS ULTIMOS CINCO AÑOS (ARTICULOS, MONOGRAFÍAS Y LIBROS) (Título, revista, fecha; si es libro editorial, ISBN))

Empleo de la computación en el desarrollo de habilidades y conocimientos en la asignatura hidráulica agrícola. Memorias de la Convención Internacional METANICA 99.

Análisis Estadístico de la información climática en la región de Santiago de las Vegas La Habana Cuba. Memorias de la Convención Internacional de la Industrias Mecánicas y Electrónica METANICA 2000.

Esquema Metodológico para la elaboración de proyectos de informática educativa.

Publicado en el sitio web monografías.com/educación.

Diseño y concepción de mapas conceptuales.

Publicado en el sitio web monografías.com/educación.

Sistema de mapas conceptuales para la enseñanza de los conceptos fundamentales del medio ambiente. Publicado en el sitio web monografías.com/educación.

Aplicación de los Mapas Conceptuales en la Gestión del conocimiento. Presentación del mapa conceptual de Medio Ambiente.

CD del Congreso Internacional CUBA-RIEGO 2003.

Los mapas conceptuales en los procesos docentes de la educación superior CD del evento IV Taller Nacional de Didáctica de la Educación Superior Universidad de la Habana 2004.

Modelación de los módulos de evaluación de los software educativos empleando redes bayesianas. CD Evento del Taller de Inteligencia Artificial del evento UCIENCIA 2005

Definiciones Fundamentales de Multimedia. Articulo publicado en el CD Reporte de Investigaciones del CEIS 2007.

Validación de la Metodología OMMMA-L para el análisis y diseño de multimedia a través de un caso de estudio. . Articulo publicado en el CD Reporte de Investigaciones del CEIS 2007.

Validación de la Metodología RMM para el análisis y diseño de multimedia a través de un caso de estudio. . Articulo publicado en el CD Reporte de Investigaciones del CEIS 2007.

Ha participado en el análisis, diseño e implementación de sistemas para las siguientes entidades.

Intranet de la empresa ECASOL- 2003

Sistema para el control de las ventas a bordo ECASA -2004

Sistema para el diagnostico arquitectónico urbano en el centro histórico de la ciudad de Camagüey- Plan Maestro de la Oficina del Historiador de la Ciudad de Camagüey – 2005

Multimedia para la enseñanza del Acondicionamiento Ambiental en la Carrera Arquitectura CUJAE- 2005

Sistema de control y aseguramiento de la calidad de ensayos reactivos Inmunoensayo 2006

Diseño e implementación de un software para la declaración, facturación y recepción de mercancías, Empresa Energoimport 2006

AGRADECIMIENTOS

A Fidel y a la Revolución cubana por permitirme estudiar en la mejor escuela del país.

A Ernesto González por ser un gran tutor y ayudarme en todo lo que necesité.

A Ernesto Guevara por dedicar parte de su tiempo a mis inquietudes.

A mis abuelos María Ana y Ernesto por confiar en mí y apoyarme en todos los momentos.

A mis padres Elsa y Gerónimo por darme la vida.

A mi hermana, eres un tesoro para mí. Te quiero mucho, mucho.

A toda mi familia por estar siempre orgullosos de mí.

A Francesco por apoyarme siempre en los momentos buenos y malos.

A Pedro por ayudarme cuando más lo necesité.

A Alberto Manso Blanco por darme tantos momentos de felicidad.

A Yasser Abdel Cruzata por quererme tanto y tener tanta paciencia conmigo.

A Denia (Mimi) y a Yanisleidy (Baúl) por ser mis amigas siempre y darme sus manos en los momentos más difíciles.

A María y a María Elena por tenerme como hija y hermana respectivamente.

A los mejores amigos (Yelenis, Yudaika, Alexander, Lisneidy, Mildrey, Annelis, Yusmaidy, Lisbet).

A Nery por ser tan buena y ayudarme en todo lo que necesité.

Geidy

A Fidel y a la Revolución cubana por permitirme estudiar en la mejor escuela del país.

A Ernesto González por ser un gran tutor y ayudarme en todo lo que necesité.

A mis padres, gracias por apoyarme en los caminos que he emprendido, por darme la fuerza necesaria para seguir adelante, por creer en mi ,por darme la vida y tantas cosas…
Están dentro de mi corazón.

A mis hermanos por ser la luz que me ilumina cada día por muy nublado que esté.

A Michel por amarme tanto y ayudarme en todo lo que necesité.
A Maikel por estar siempre a mi lado compartiendo mis tristezas y alegrías apoyándome en todo momento, por ser amigo incondicional.

A Enrique por escuchar mis problemas y darme energía para vencer todos los obstáculos de la vida e iluminarme para llevar a cabo mis objetivos y metas.

A tía María, a tío Miguel, a mi familia entera por preocuparse tanto por mí; por quererme, por tenderme la mano en los momentos difíciles y aconsejarme tanto.

A Beatriz Fuentes y Pascual Verdecia, por su ayuda, por enseñarme tantas cosas, porque de una u otra forma depositaron muchos granitos de arena en mi vida.

A todos mis profes de la Universidad de Moa, Mirelis, Rafael y Lores por el camino recorrido, que sin ellos no habría llegado hasta aquí.

A mis amigos (Katy, Yoyi, Mary, Yamilka, Yudaika, Susana, Jean).

A Geidy por ser mi rayito de luz.

Nereyda

DEDICATORIA

A mi gran tesoro: Mi abuela porque siempre confió en mi, por su amor y apoyo en cada instante de mi vida.

Geidy

A mi madre: Georgina que siempre esta a mi lado brindándome su apoyo y su fe en mí.
Y en especial a mi padre Rubén, todo mi éxito se lo dedico a él.

Nereyda

RESUMEN

En este trabajo, se presentan y evalúan diferentes algoritmos de Agrupamiento y de Clasificación, para la determinación de su eficiencia y aplicabilidad en problemas de detección y seguimiento de tópicos a partir de varios corpus de texto preparados a tal efecto. El presente trabajo permite conocer cuál de los algoritmos anteriormente mencionados es el más eficiente a partir de resultados obtenidos con la herramienta WEKA para la detección y seguimiento de tópicos, además se expone cómo se realizó la preparación del corpus de texto. Este trabajo puede ser muy útil debido a que no se cuenta hoy con un estudio similar y cada día aumentan más los volúmenes de texto que abordan una misma temática y es necesario procesar con vista a obtener la mejor información posible de los mismos.

PALABRAS CLAVES:

Procesamiento del lenguaje natural, minería de texto, minería de datos, algoritmos de agrupamiento y algoritmos de clasificación.

Tabla de Contenidos

INTRODUCCIÓN

En aplicaciones donde existe un flujo continuo de documentos se requiere de mecanismos automáticos que operando a la misma velocidad que el flujo, organicen y filtren la información para su posterior estudio por parte de los usuarios. Una de estas aplicaciones consiste en la Detección y el Seguimiento automático de sucesos en flujos de noticias digitales, también conocida como TDT (Topic Detection and Tracking).

El principal problema planteado consiste en determinar si un documento entrante informa sobre un nuevo tópico o suceso o forma parte de otros sucesos recogidos por el sistema. La tarea de detección es una abstracción experimental del agrupamiento de noticias[1]. El objetivo de un sistema de detección es agrupar y clasificar las noticias que abordan un mismo suceso o tópico. Es preciso tener muy en cuenta que el conjunto de noticias cambia en el tiempo, pues es necesario modificar el agrupamiento y clasificación de estas a medida que se van publicando nuevas noticias para mantenerlo actualizado.

Para los problemas de Detección y Seguimiento de tópicos se emplean técnicas de la Minería de Texto, como son los algoritmos de agrupamiento y clasificación. Las medidas de calidad para estos algoritmos fueron definidas por DARPA (Defense Advanced Research Projects Agency) para investigar las aproximaciones desarrolladas en la Detección y Seguimiento de tópicos. Se encuentran dentro de estas medidas, la medida F1 y Coste de detección. Dentro de las principales tareas de la Detección y Seguimiento de tópicos se pueden mencionar la Segmentación de noticas, la Detección de tópicos, la Detección de la primera noticia y la Detección de enlaces.

[1] http://www.deli.deusto.es/wiki/index.php/TA/NIST_MT

La **situación problemática** del presente trabajo se puede resumir en:

- Existen varios algoritmos para la minería de textos, pero no se cuenta con un estudio comparativo de los mismos a partir de su eficiencia y aplicabilidad a problemas de detección de tópicos.

El **problema** que genera dicha situación es:

Necesidad de realizar una valoración de las diferentes técnicas de minería de texto a problemas de detección de tópicos.

El **objeto de estudio** en este trabajo es:

Las diferentes técnicas de la inteligencia artificial para la Minería de textos.

El **campo de acción** presente en este trabajo es:

Los algoritmos para clasificación y agrupamiento de textos.

El **objetivo general** de este trabajo es:

- Valorar diferentes técnicas de minería de textos para la detección de tópicos a partir del análisis de dichas técnicas con un corpus de texto.

Para cumplir con el objetivo de este trabajo se trazaron las siguientes **tareas**:

- Realizar una revisión del estado del arte de los trabajos enmarcados en la Minería de Texto.
- Realizar una búsqueda bibliográfica de las diferentes técnicas de la I.A para la Minería de Texto.
- Seleccionar las técnicas a valorar.
- Estudiar dichas técnicas.

En el desarrollo del trabajo se emplearon los siguientes Métodos Científicos:

Métodos Teóricos:

- **Dialéctico para la obtención del conocimiento**: Es el método fundamental que guía la investigación, lo que permite que la investigación se oriente por vías científicas de demostración del problema y de obtención de las conclusiones a partir de los resultados de los experimentos.

- **Análisis-síntesis:** Permite inicialmente descomponer el problema en sus partes componentes y posteriormente volver a integrarlas, en el trabajo este método se emplea en el análisis de los diferentes algoritmos con vistas a seleccionar los que se van a evaluar, y de las diferentes herramientas que se pudieran emplear, así como en la obtención de conclusiones a partir de los resultados de los experimentos.

- **Inductivo-deductivo**: Para ir de lo general a lo particular y viceversa evidenciándose en el trabajo en el momento de realizar las valoraciones de los diferentes algoritmos seleccionados y en la aplicación de los conceptos fundamentales a el problema particular del trabajo.

- **Analógico:** Para hacer comparaciones entre problemas similares de manera tal que permita encontrar soluciones a partir de estas similitudes, en la realización de este trabajo se hicieron valoraciones de varios algoritmos para la detección y seguimiento de tópicos a partir de los mismo corpus de textos, lo que permite obtener conclusiones sobre la eficiencia y el coste de detección de estos algoritmos a partir de analogías.

- **Sistémico:** Enfoca el problema como un todo, este método permite enfocar los distintos componentes del problema y las diferentes tareas a realizar de manera integrada e interrelacionadas unas con otras.

Métodos Empíricos:

Recolección de información y **análisis documental**: Para lo cual se desarrolló una búsqueda bibliográfica y se consultaron varios especialistas y expertos en la temática. , además de la búsqueda y preparación de conjuntos de noticias de diferentes tópicos y agencias de noticias.

Método experimental: Análisis, diseño y desarrollo de los experimentos que permitieron llegar a las conclusiones finales del trabajo.

Este documento de tesis se divide en **dos** capítulos:

- **Fundamentación Teórica**: Se presentará el fundamento teórico: Marco teórico y modelo teórico. Consiste en un análisis crítico de investigaciones anteriores y de fuentes con enfoques, teorías y modelos relacionados con el estudio del estado del arte de la Inteligencia Artificial, la Minería de Texto y de los algoritmos de Clasificación y Agrupamiento. Se describen los métodos, procedimientos y técnicas utilizadas.

- **Modelo, propuesta concreta de la tesis:** En este capítulo se detallan los algoritmos de Clasificación y Agrupamiento. Se presentan también los métodos utilizados por los diferentes algoritmos. Se proponen en esta sección los modelos, metodologías y procedimientos. Se realiza un análisis de los resultados concluido el trabajo de diploma y teniendo en cuenta la caracterización y análisis crítico de la investigación se sugiere aplicar encuestas u otras herramientas para la validación de las propuestas de este capítulo.

CAPITULO I

Capítulo 1: Fundamentación Teórica

Introducción

Las noticias representan un dominio de información ideal para el estudio de la detección y seguimiento de nuevos sucesos. Un sistema de Detección y Seguimiento de Tópicos (TDT, siglas en inglés de *Topic Detection and Tracking*) investiga métodos para la organización de las noticias en tópicos y clasifica y organiza nuevas noticias para un usuario interesado en realizar un seguimiento de los sucesos de actualidad que se obtienen de diversas fuentes en línea.

En la actualidad existen muchas aplicaciones prácticas donde se necesita la detección y seguimiento de noticias, especialmente para comerciantes, financieros, analistas de los medios de comunicación y editores de periódicos digitales en línea, los cuales coleccionan, interpretan y muestran las noticias procedentes de varias fuentes. Basta pensar, por ejemplo, en un analista político que tiene que leer diariamente un gran número de cables para identificar cuáles de ellos se refieren al tópico que desea abordar en su comentario. Hoy en día, el volumen de noticias en línea en Internet es enorme (de hecho, la mayoría de las agencias de noticias y periódicos del mundo proveen sus noticias no sólo en papel sino también en Internet) y, por tanto, se hace necesario el desarrollo de herramientas eficientes y eficaces que sean capaces de procesarlas. Un sistema que organice los sucesos y detecte los nuevos sucesos que ocurran sería útil para aquellas aplicaciones cuyos datos de entrada sean noticias y donde la decisión a tomar sea detectar si ha ocurrido un nuevo suceso o identificar las noticias que conforman un suceso al que hay que darle seguimiento.

TDT es una nueva línea de investigación compuesta, en sus inicios, por tres subproblemas principales: la segmentación y reconocimiento del habla a partir del flujo de noticias procedentes de la radio y la TV; la detección de nuevos sucesos en el flujo de noticias segmentadas o no, y el seguimiento del desarrollo de un suceso a partir de una muestra de noticias sobre el mismo suceso identificada por el usuario.

Las investigaciones en TDT comenzaron en 1996 [Alla, 98]. El proyecto TDT es una iniciativa patrocinada por DARPA (*Defense Advanced Research Projects Agency*) dentro del programa TIDES (*Translingual Information Detection, Extraction and Summarization*) para investigar las aproximaciones desarrolladas en la detección y seguimiento de nuevos sucesos en un flujo de noticias (habladas o escritas). Las tareas

TDT y las aproximaciones para su evaluación fueron desarrolladas en un esfuerzo conjunto entre DARPA, la Universidad de Massachusetts, el Instituto Tecnológico para el Lenguaje de la Universidad de Carnegie Mellon y los Sistemas Dragón. Durante un año se hizo un estudio piloto para definir el problema claramente, desarrollar las bases de la investigación y evaluar la habilidad de las tecnologías actuales para solucionar el problema. Los resultados finales del estudio se expusieron en un taller en 1997, elaborándose un informe final llamado TDT1 [Alla, 98]. El propósito de ese estudio fue desarrollar aún más las tecnologías requeridas para segmentar, detectar y seguir información en una cadena continua de noticias; así, las noticias viejas pueden ser seguidas y las nuevas, detectadas, aunque provengan de distintas fuentes.

Las investigaciones en TDT han continuado desarrollándose y en este período se han realizado seis evaluaciones: en 1998, 1999, 2000, 2001, 2002 y 2003[1]. Estos esfuerzos han dado lugar a algoritmos para el descubrimiento y seguimiento de sucesos y tópicos en un flujo de noticias para los idiomas inglés, chino mandarín y árabe.

Conceptos Preliminares.

1.1 Inteligencia Artificial.

Algunas definiciones de inteligencia artificial plantean:

- Estudio de los mecanismos de la inteligencia y las tecnologías que lo sustentan. [Newell, 91].
- Intento de reproducir (modelar) la manera en que las personas identifican, estructuran y resuelven problemas difíciles [Pople, 84].

  Son ciertas herramientas de programación, entendiendo por Herramientas:
  - ❖ Lenguajes: LISP, PROLOG
  - ❖ Entornos de desarrollo: shells
  - ❖ Arquitecturas de alto nivel: nodo y arco, sistemas de producciones
- La interesante tarea de lograr que las computadoras piensen […] *máquinas con mente*, en su amplio sentido natural."[Haugeland, 1985].
- "La automatización de actividades que vinculamos con procesos de pensamiento humano, actividades tales como toma de decisiones, resolución de problemas, aprendizaje…" [Bellman, 1978].

[1] http://www.nist.gov/speech/tests/tdt.html

- "El estudio de las facultades mentales mediante el uso de modelos computacionales." [Charniak y MacDermott, 1985].
- "El estudio de los cálculos que permiten percibir, razonar y actuar." [Winston, 1992].
- "El arte de crear máquinas con capacidad de realizar funciones que cuando son hechas por personas requieren de inteligencia." [Kurzweil, 1990].
- "El estudio de cómo lograr que las computadoras realicen tareas que, por el momento, los humanos hacen mejor." [Rich y Knight, 1991].
- "Un campo de estudio que se enfoca a la explicación y emulación de la conducta inteligente en función de procesos computacionales." [Schalkoff, 1990].
    - "La rama de la ciencia de la computación que se ocupa de la automatización de la conducta inteligente." [Luger y Stubblefield, 1993].

Las autoras de este trabajo, a partir del análisis realizado, consideran que la Inteligencia Artificial es el estudio de las técnicas computacionales que permitan dar solución a problemas para los cuales no existen algoritmos o los que existen son de complejidad no polinomial.

1.2 Lingüística computacional y procesamiento de textos.

La lingüística computacional es la ciencia que trata de la aplicación de los métodos computacionales en el estudio del lenguaje natural [Gelbukh y Bolshakov, 1999]. Esta ciencia es una combinación de dos ciencias más grandes; la lingüística, que estudia las leyes del lenguaje humano, y la inteligencia artificial (ver figura 1.1).El problema u objetivo más importante de la lingüística computacional es la comprensión del lenguaje, es decir, la transformación del lenguaje hablado o escrito a una representación formal del conocimiento, como por ejemplo una red semántica. La solución tradicional de este problema consiste en construir un procesador lingüístico constituido por diferentes módulos independientes (ver figura 1.2):

- El módulo morfológico se encarga de reconocer las palabras. Básicamente, convierte las cadenas de letras a una entrada de un diccionario, y pone las marcas de tiempo, género y número.
- El módulo sintáctico reconoce oraciones. Este módulo convierte las cadenas de palabras marcadas a una estructura gráfica, en donde se hacen explícitas algunas relaciones entre las palabras de la oración.
- El módulo semántico reconoce la estructura completa del texto y lo convierte a una "red semántica".

La lingüística computacional se encarga de otras cosas, adicionalmente a la comprensión del lenguaje. Algunas de estas otras áreas de investigación de la lingüística computacional se muestran en la (figura 1.2). La más grande de estas áreas, y tal vez la más importante, es el procesamiento automático de textos. El procesamiento automático de textos considera una gran diversidad de tareas (ver figura 1.3), desde muy simples, como la separación de palabras, hasta muy complejas como algunas tareas de minería de texto.

Sin lugar a dudas, la lingüística computacional en su conjunto enfrenta uno de los más grandes retos de la ciencia computacional: lograr que las computadoras sean nuestros verdaderos ayudantes en la ocupación principal de la raza humana, pensar y comunicar. Además, las tareas de la lingüística computacional tienen una gran utilidad práctica inmediata, ya que se relacionan con: la toma de decisiones, la búsqueda e intercambio de conocimiento, y toda clase de operaciones relacionadas con la publicación y uso de los documentos. Así pues, sin temor a equivocarnos, podemos decir que los países que disponen de buenas herramientas para el análisis y generación de textos tienen, en nuestro mundo competitivo, una gran ventaja económica, tecnológica y hasta militar sobre los demás países.

1.3 Procesamiento del lenguaje natural:

Un objetivo principal de la Inteligencia Artificial lo constituye sin lugar a dudas el Procesamiento del Lenguaje Natural por computadora. El Procesamiento del Lenguaje Natural (PLN) es una parte esencial de la Inteligencia Artificial que investiga y formula mecanismos automáticos efectivos que faciliten la interrelación hombre-máquina y permitan una comunicación mucho más fluida y menos rígida que los lenguajes formales y sistemas de menús utilizados tradicionalmente. Todo el sistema de Procesamiento del Lenguaje Natural intenta simular un comportamiento lingüístico humano; para ello debe tomar conciencia tanto de las estructuras propias del lenguaje, como de un conocimiento general acerca del universo de discurso. De esta forma, una persona que participe en un diálogo sabe cómo pueden combinar las palabras para formar una oración, conoce los significados de las mismas, sabe cómo éstas afectan el significado global de la oración y tienen un conocimiento del mundo en general que permite participar de la conversación.

1.3.1 Lenguaje:

Lenguaje es el empleo de la palabra para expresar ideas, comunicarse, establecer relaciones entre los seres humanos. Un lenguaje es un conjunto de palabras, su pronunciación y los métodos para combinarlas en frases y oraciones, generalmente infinito y que se forma mediante combinaciones de palabras definidas en un diccionario terminológico previamente establecido. Las combinaciones posibles deben respetar un conjunto de reglas sintácticas establecidas, a ello se le conoce con el nombre de Sintaxis. Además, las palabras deben tener determinado sentido, deben ser comprendidas por un grupo humano en un contexto dado, a ello se le denomina Semántica. El ser humano en el transcurso de su desarrollo histórico ha utilizado los lenguajes para expresar sus pensamientos, emociones, sentimientos y para establecer comunicación entre grupos, etnias, naciones y sociedades humanas en su conjunto. Esta función del lenguaje se realiza mediante señales: gráficas, sonoras, lumínicas, y la escritura.

1.3.2 Lenguaje Natural:

Se le denomina al lenguaje escrito o hablado usado por una comunidad que es precisamente lo contrario a un lenguaje para establecer comunicación con una computadora, mediante la entrada de datos, o la programación de su funcionamiento.

1.3.3 Lenguaje Formal:

Se le denomina lenguaje Formal a un lenguaje artificial o sea creado por el hombre que esta formado por símbolos y fórmulas y que tiene como objetivo fundamental formalizar la programación de computadoras.

1.4 Minería de Datos.

Bajo el nombre de minería de datos se engloban un conjunto de técnicas encaminadas a la extracción de "conocimiento" procesable implícito en las bases de datos de determinadas entidades entre ellas: las empresas. Las bases de la minería de datos se encuentran en la inteligencia artificial y en el análisis estadístico. Mediante los modelos extraídos utilizando técnicas de minería de datos se aborda la solución a problemas de predicción, clasificación y segmentación. Un proceso típico de minería de datos parte de la selección del conjunto de datos, tanto en lo que se refiere a las variables dependientes, como a las variables objetivo, como posiblemente al muestreo de los registros disponibles.

Los modelos obtenidos por técnicas de minería de datos se aplican incorporándolos en los sistemas de análisis de información de las empresas, e incluso, en los sistemas transaccionales. En este sentido cabe destacar los esfuerzos del Data Mining Group, que está estandarizando el lenguaje PMML (Predictive Modelling Markup Language), de manera que los modelos de minería de datos sean interoperables en distintas plataformas, con independencia del sistema con el que han sido construidos. Los principales fabricantes de sistemas de bases de datos y programas de análisis de la información hacen uso de este estándar.

Las técnicas de minería de datos se aplicaban sobre información contenida en almacenes de datos. De hecho, muchas grandes empresas e instituciones han creado y alimentan bases de datos especialmente diseñadas para proyectos de minería de datos en las que centralizan información potencialmente útil de todas sus áreas de negocio, etc. [1]

1.4.1 Estado del arte de la Minería de Datos.

El concepto de Minería de Datos surgió hace más de 10 años. La aplicación y desarrollo de la minería en múltiples ramas como los negocios, finanzas, ingeniería, banca, salud, sistemas de energía y meteorología; así como el entorno altamente competitivo de las empresas, que requieren mantener y ganar nuevos clientes; ha llevado a que el interés por este campo se haya incrementado.

Las técnicas de Minería de Datos han madurado con el paso de los años y algunos de los factores que han contribuido a su desarrollo han sido el surgimiento de grandes cantidades de datos en las entidades, el desarrollo de las técnicas de aprendizaje automático, la posible presencia de incertidumbre en los datos y el rápido crecimiento del manejo de sistemas de bases de datos.

Los algoritmos son más o menos eficientes y pueden manipular datos complicados. Las herramientas son cada vez más potentes, permitiendo desarrollar aplicaciones de minería más completas.

[1] http://es.wikipedia.org/wiki/Miner%C3%ADa_de_datos

1.4.2 Herramientas de la Minería de Datos.

IBM

IBM tiene un producto de minería de datos llamado Intelligent Miner, desarrollado por una subsidiaria alemana de IBM. Intelligent Miner contiene un conjunto de algoritmos y permite exportar modelos de minería en Predictive Modeling Markup Language (PMML). [TangM, 2005]. PMML es un lenguaje estándar basado en XML el cual es desarrollado por Data Mining Group (DMG), un grupo líder en desarrollo y venta de estándares para minería de datos. Los archivos PMML pueden ser cargados por la base de datos con propósitos de predicción.

Oracle

ORACLE 10g incorpora el paquete ODM (Oracle Data Mining), lo que simplifica el proceso de extracción de conocimiento ya que los datos no tienen que ser movidos para realizar el análisis. De esta forma todas las operaciones de preparación, limpieza, creación de modelos e implementación permanecen en la base de datos lo que resulta de gran importancia para mejorar la productividad, automatización e integración de los proyectos de Minería.

SQL Server 2000 de Microsoft

La primera versión del paquete Analysis Services que se utiliza para realizar proyectos de Minería en SQL Server, se incorpora en el SQL Server 2000. De esta forma SQL Server se introduce en el mercado de las compañías que se dedican al proceso de descubrir y extraer conocimiento en grandes volúmenes de información, a través de su especificación OLE DB para Minería de Datos. OLE DB para Minería de Datos es un estándar de la industria que define un lenguaje de minería de datos con el estilo de SQL y hace factible el manejo de componentes, especialmente características de predicción. SQL Server 2000 contiene además dos algoritmos de minería de datos: Microsoft Clustering y Árboles de Decisión de Microsoft.

SQL Server 2005

Microsoft SQL Server 2005 Analysis Server establece nuevas facilidades para realizar Minería de Datos:

- Explorar y manipular datos, así como diseñar y editar modelos.
- El procesamiento de los modelos de una misma estructura de minería ocurre en paralelo, en una sola lectura de los datos.
- Proporciona más de 12 visores de resultados para los algoritmos que ayudarán a comprender mejor los patrones encontrados en el proceso de minería.
- Proporciona gráficos de elevación, de beneficios y una matriz de clasificación que permite establecer una comparación de lo real con lo previsto; para contrastar y comparar la calidad de los modelos.
- Posee un lenguaje para la creación de consultas de minería (DMX) similar al SQL que facilita la tarea de creación de aplicaciones de minería de datos.
- Cuenta con los algoritmos de minería: Naive Bayes, Clustering, Clústeres de Secuencia, Árboles de Decisión, Redes Neuronales, Series Temporales, Reglas de Asociación, Regresión Logística, y Regresión Lineal y minería de textos.
- Marco de desarrollo para agregar nuevos algoritmos y también para construir visores propios para los modelos generados. [Crivat, 2005] [Iyer, 05] [MacLennan, 2004] [Netz, 2005] [Tang, 2005] [TangM, 2005].

1.4.3 Actividades dentro de un proyecto de Minería de Datos.

Dentro de la Minería de Datos se incluyen actividades en las que se tiene claro el objetivo desde un inicio, pertenecientes a la Minería de Datos directa (MDD) y otro grupo en el que no se sabe a ciencia cierta qué resultado se quiere obtener, conocido como Minería de datos indirecta (MDI) [Berry, 2000].

Tabla 1.1 Actividades de la Minería de Datos Directa y de la Minería de Datos Indirecta

Minería de Datos Directa	**Minería de Datos Indirecta**
1. Clasificación	2. Determinar grupos afines o reglas de asociación
3. Estimación	4. Clustering
5. Predicción	6. Descripción y visualización

A continuación se describen las actividades de cada grupo y a la vez se presenta un enfoque práctico de los resultados que pueden obtenerse con las técnicas de cada tipo de actividad. [Rosete, 2004].

Clasificación (MDD)

La clasificación consiste en examinar características de un objeto (registro) y asignarle una clase predefinida. En este caso las salidas son clases que son valores discretos. Esta tarea se realiza de muchas maneras, el punto en común en cada caso es la construcción de un modelo para hacer la clasificación. Ejemplos:

- Asignar palabras claves a documentos.
- Clasificar los préstamos que brinda un banco por riesgo (alto, medio, bajo).
- Clasificar transacciones fraudulentas o no.

Estimación (MDD)

La estimación es similar a la clasificación, pero sus salidas son valores continuos. En algunos casos puede hacerse previo a la clasificación. Ejemplo:

- Determinar la probabilidad de que una transacción sea fraudulenta. En este caso, luego se puede clasificar usando umbrales.
- Asignar un valor entre 0 y 100 a los préstamos que sean más aptos según sea el riesgo que asume el banco al hacerlo.
- Determinar el número de minutos que juega un determinado jugador de baloncesto antes que se agote.
- Determinar el valor con que cerrarán determinadas acciones en la bolsa de valores.

Predicción (MDD)

La predicción es similar a cualquiera de los anteriores, pero la salida (sea esta discreta o continua) no ha ocurrido. Su peculiaridad es que la variable que se estima o la clase que se asigna se corresponden con un fenómeno que ocurrirá en el futuro. Ejemplos:

- Determinar si un usuario pedirá de nuevo determinado servicio.
- Determinar si un usuario comprará un producto que se le está haciendo marketing.

- Determinar si un usuario solicitará servicios telefónicos agregados a partir de analizar sus gastos infiriendo si estos son por uso de Internet.

Determinar grupos afines o reglas de asociación (MDI)

Esta actividad de Minería de Datos (MDI) tiene como objetivo encontrar fenómenos que ocurren de conjunto sin que quede claro el tipo de relación causal que ocurre entre ellos. Ejemplo:

- Identificar y agrupar productos que se compran juntos. Este análisis permite a los dueños o a los que administran la política comercial, presentar los productos juntos y/o establecer políticas de marketing combinadas.

Clustering (MDI)

Clustering significa agrupamiento, consiste en segmentar un grupo diverso en subgrupos. Para esto se toman los valores de diferentes variables para un determinado fenómeno y se crean grupos según el grado de semejanza entre ellos. Esta búsqueda de semejanza se realiza calculando distancia por métodos muy parecidos a los usados en el razonamiento basado en casos [Rich, 1994] [Shapiro, 1990].
El Agrupamiento se considera una técnica de la MDI porque los grupos que se obtienen no tienen un significado a priori. No se conoce la cantidad ni el significado de los grupos que se obtienen antes de correr el algoritmo. A los grupos se le da un significado después de obtenido con la ayuda de los expertos del dominio del negocio. Ejemplos:

- Agrupar los usuarios según los productos que compran.
- Agrupar al personal de una empresa según su edad, nivel profesional, coeficiente de inteligencia, indicadores de salud.

Descripción y visualización (MDI)

Una última técnica de MDI que es muy importante es la de descripción y visualización. Esta ayuda a entender mejor los problemas. Siendo así, su aplicación permite enfocar las demás actividades de MD.

También permite encontrar explicaciones a fenómenos o al menos elaborar hipótesis iníciales para el trabajo. "una imagen vale más que 100 palabras".

1.4.4 Aplicaciones de la Minería de Datos.

La integración de las técnicas de minería de datos en las actividades del día a día se está convirtiendo en algo habitual. Los negocios de la distribución y la publicidad dirigida han sido tradicionalmente las áreas en las que más se han empleado los métodos de minería, ya que han permitido reducir costes o aumentar la receptividad de ofertas. Pero éstas no son las únicas áreas a las que se pueden aplicar. De hecho, podemos encontrar ejemplos en todo tipo de aplicaciones: financieras, seguros, científicas (medicina, farmacia, astronomía, informática psicología, etc.), políticas económicas, sanitarias o demográficas, educación, policiales, procesos industriales y un largo etcétera.

Aplicaciones financieras y banca:

- Obtención de patrones de uso fraudulento de tarjetas de crédito.
- Determinación del gasto en tarjeta de crédito por grupos.
- Identificación de reglas de mercado de valores a partir de históricos.

Análisis de mercado, distribución y, en general, comercio:

- Análisis de la cesta de la compra (compras conjuntas, secuenciales, ventas cruzadas, señuelos, etc.).
- Análisis de la fidelidad de los clientes.
- Estimación de stocks, de costes, de ventas, etc.

Seguros y salud privada:

- Determinación de los clientes que podrían ser potencialmente caros.
- Predicción de qué clientes contratan nuevas pólizas.
- Identificación de patrones de comportamiento para clientes con riesgo.
- Identificación de comportamiento fraudulento.

Educación:

- Selección o captación de estudiantes.
- Detección de abandonos y de fracaso.

- Estimación del tiempo de estancia en la institución.

Procesos industriales:

- Predicción de fallos y accidentes.
- Extracción de modelos de coste.
- Extracción de modelos de producción.

Medicina:

- Identificación de patologías. Diagnóstico de enfermedades.
- Detección de pacientes con riesgo de sufrir una patología concreta.
- Recomendación priorizada de fármacos para una misma patología.

Telecomunicaciones:

- Establecimiento de patrones de llamadas.
- Modelos de carga en redes.
- Detección de fraude.

Informática:

- Inteligencia Artificial: Mediante un sistema informático que simula un sistema inteligente, se procede al análisis de los datos disponibles. Entre los sistemas de Inteligencia Artificial se encuadrarían los Sistemas Expertos y las Redes Neuronales.
- Sistemas Expertos: Son sistemas que han sido creados a partir de reglas prácticas extraídas del conocimiento de expertos. Principalmente a base de inferencias o de causa-efecto.
- Sistemas Inteligentes: Son similares a los sistemas expertos, pero con mayor ventaja ante nuevas situaciones desconocidas para el experto.
- Redes neuronales: Genéricamente, son métodos de proceso numérico en paralelo, en el que las variables interactúan mediante transformaciones lineales o no lineales, hasta obtener unas salidas. Estas salidas se contrastan con los que tenían que haber salido, basándose en unos datos de prueba, dando lugar a un proceso de retroalimentación mediante el cual la red se reconfigura, hasta obtener un modelo adecuado.

1.4.5 Tendencias de la Minería de Datos.

En la breve historia de la minería de datos, se han cumplido algunas expectativas y se han dejado abiertas otras muchas. En particular, se espera una minería de datos más automática, más sencilla, con más

fiabilidad, con patrones más novedosos y más eficiente. De hecho, según autores, se pueden destacar todavía más retos. Por ejemplo, Han y Kamber [Han y Kamber, 2001] afirman que para que la minería de datos sea completamente aceptada como una tecnología, se deben resolver algunos problemas principalmente relacionados con la eficiencia y la escalabilidad, la interacción con el usuario, la incorporación de conocimiento de base, las técnicas de visualización, la evolución de lenguajes de consultas de minería de datos estandarizados y mejorar el tratamiento de datos complejos, entre otros.
Uno de los principios de la minería de datos es que tiene que trabajar de forma eficiente y efectiva con grandes bases de datos.

La Minería de Datos ha sufrido transformaciones en los últimos años de acuerdo con cambios tecnológicos, de estrategias de marketing, la extensión de los modelos de compra en línea, etc. Los más importantes de ellos son:

- La importancia que han cobrado los datos no estructurados (texto, páginas de Internet, etc.)
- La necesidad de integrar los algoritmos y resultados obtenidos en sistemas operacionales, portales de Internet, etc.
- La exigencia de que los procesos funcionen prácticamente en línea (por ejemplo, que frente a un fraude con una tarjeta de crédito, ésta pueda ser cancelada casi al instante).

1.5 Minería de texto

La minería de texto es la más reciente área de investigación del procesamiento de textos. Ella se define como el proceso de descubrimiento de patrones interesantes y nuevos conocimientos en una colección de textos, es decir, la minería de texto es el proceso encargado del descubrimiento de conocimientos que no existían explícitamente en ningún texto de la colección, pero que surgen de relacionar el contenido de varios de ellos [Hearst y Kodratoff, 1999].
Este proceso consiste de dos etapas principales: una etapa de pre-procesamiento y una etapa de descubrimiento [Tan, 1999]. En la primera etapa, los textos se transforman a algún tipo de representación estructurada o semi-estructurada que facilite su posterior análisis, mientras que en la segunda etapa las representaciones intermedias se analizan con el objetivo de descubrir en ellas algunos patrones interesantes o nuevos conocimientos. La figura 1.4 ilustra este proceso.

Dependiendo del tipo de métodos usados en la etapa de pre – procesamiento es el tipo de representación del contenido de los textos construida; y dependiendo de esta representación, es el tipo de patrones descubiertos. La figura 1.5 muestra los tres tipos de estrategias empleadas en los actuales sistemas de minería de texto.

Figura 1. 5. Estado del arte de la Minería de texto.

La minería de texto o Text Mining es una herramienta que proviene del área del procesamiento automático de textos y que permite localizar y extraer la información más significativa y esencial de los documentos, así como información y conocimiento implícito y oculto en grandes corpus textuales electrónicos, estructurados o no estructurados, como mensajes de correos electrónicos, discursos, artículos, entre otros. Debido a esto, en ocasiones se asocia con el espionaje.

Funciona a partir de una telaraña semántica, que tiene como objetivo construir toda una estructura de metadatos, información sobre la estructura y significado de los datos almacenados e incluirlos en los documentos de forma que sean navegables, identificables y entendibles por las máquinas, por lo que es una herramienta eficaz para gestionar el conocimiento. “Se enfoca en el descubrimiento de patrones interesantes y nuevos conocimientos en un conjunto de textos, es decir, su objetivo es descubrir tendencias, desviaciones y asociaciones en la gran cantidad de información textual disponible”, es decir, facilita realizar análisis y se instituye como un área emergente de la minería de datos. Elimina la información duplicada y detecta información similar o relacionada con la existente. La minería de textos utilizada en las Ciencias de la Información pudiera explotarse como herramienta en los nuevos métodos de resumen porque permite la decodificación y análisis del lenguaje natural e interfaces en la lengua materna de cada dominio, traducción automática, procesamiento de voz, generación de texto, etcétera. Todas estas cualidades de la minería de texto son la razón que fundamenta la propuesta de esta herramienta como perspectiva metodológica para la realización de resúmenes documentales.

Las perspectivas metodológicas de la minería de texto aplicables en las instituciones de información son disímiles, porque su rango de acción no sólo se desarrolla en el trabajo con el texto, sino que además explora otros sectores como el procesamiento de voz, decodificación de imágenes, construcción de corpus documentales, representación y graficación de términos mediante herramientas de ponderación asociadas, entre otros.

1.5 .1 Técnicas de minería de texto.

La minería de texto es el proceso encargado del descubrimiento de conocimiento que no existe en el texto, pero que surge al relacionar el contenido de varios textos.
La minería de texto se divide en dos etapas que son el pre-procesamiento y una etapa de descubrimiento. Dependiendo del tipo de métodos utilizados en la etapa de pre-procesamiento se genera una representación distinta del contenido del texto.

1.5 .1 Técnicas clásicas:

Las técnicas clásicas en minería de texto se estructuran básicamente en tres etapas:

- Etapa de pre-procesamiento: Es el proceso mediante el cual los textos se transforman en algún tipo de representación estructurada que facilite su análisis.
- Etapa de representación: La representación depende de la técnica de pre-procesamiento utilizada y determinarán cuál será el algoritmo de descubrimiento a utilizar.
- Etapa de descubrimiento: Son algoritmos que a partir de una representación estructurada de la información, son capaces de descubrir regularidades en los textos.

Como se puede observar, todas las etapas están muy interrelacionadas, así pues, la primera etapa condiciona el descubrimiento de los patrones que la minería de texto puede realizar.
Las técnicas más usadas en minería de texto son los vectores de temas que muestran el nivel temático del texto, la secuencia de palabras que permite descubrir patrones en el texto y las tablas de datos que permite descubrir interrelaciones entre entidades.

1.5 .2 Herramientas para Minería de Texto.

Realizan la minería de datos (MD) o minería de textos (Data Mining, Text Mining) a partir de los datos que se recopilan en la organización con los sistemas de búsqueda, recuperación, filtrado y almacenamiento, tanto de información interna como externa. Las herramientas de MD se orientan a obtener información sobre posibles comportamientos futuros a partir de datos presentes o pretéritos.

Algunos sistemas que se emplean para hacer minería de texto son: SMART, ANES, SIM-SUM, KADS, Classifier, Parse r, Text Classifier, Text Recognizer, la plataforma ILC, NEURODOC, SDOC, HENOCH, entre otros. Todos estos sistemas permiten extraer la información relevante de un documento, agregan y comparan información automáticamente, clasifican y organizan los documentos según su contenido y organizan los depósitos para la búsqueda y recuperación de la información, pero la elección del sistema que permitirá hacer minería de texto estará determinada por la misión, visión y objetivos de la institución de información, así como las tecnologías disponibles para su implementación.

Listado de algunas aplicaciones relacionadas con minería de datos (data mining)

- WekaMetal – Extensión de meta-aprendizaje de Weka.
- Weka-Parallel – Procesos paralelos para Weka
- Weka Visualization tools - Usando PMML, VisWiz, y ROCOn.
- Weka on Text – Software para la Minería de Texto.
- Semi-Supervised and Collective- Clasificación usando Weka.
- Mathematica- Interface para Weka.
- weka4WS – Distribuidor de Minería de Datos.

1.5 .3 Herramienta de Minería de texto (WEKA).

Weka contiene una extensa colección de algoritmos de Máquinas de conocimiento desarrollados por la universidad de Waikato (Nueva Zelanda) implementados en Java; útiles para ser aplicados sobre datos mediante los interfaces que ofrece o para embeberlos dentro de cualquier aplicación. Además posee las herramientas necesarias para realizar transformaciones sobre los datos, tareas de clasificación, regresión, clustering, asociación y visualización. Weka está diseñado como una herramienta orientada a la extensibilidad por lo que añadir nuevas funcionalidades es una tarea sencilla.

Sin embargo, y pese a todas las cualidades que Weka posee, tiene un gran defecto y éste es la escasa documentación orientada al usuario que tiene junto a una usabilidad bastante pobre, lo que la hace una herramienta difícil de comprender y manejar sin información adicional.

Weka está programado en Java, es independiente de la arquitectura, ya que funciona en cualquier plataforma sobre la que haya una máquina virtual Java disponible.

Weka es un conjunto de librerías JAVA para la extracción de conocimientos desde bases de datos. Es un software que ha sido desarrollado bajo licencia GPL lo cual ha impulsado que sea una de las suites más utilizadas en el área en los últimos años.

La versión 3.4.7 incluye las siguientes características:

- Diversas fuentes de datos (ASCII, JDBC).
- Interfaz visual basado en procesos/flujos de datos (rutas).
- Distintas herramientas de minería de datos: reglas de asociación (a priori, Tertius, ...), agrupación/segmentación/conglomerado (Cobweb, EM y k-medias), clasificación (redes neuronales, reglas y árboles de decisión, aprendizaje Bayesiona) y regresión (Regresión lineal,SVM...).
- Manipulación de datos (pick & mix, muestreo, combinación y separación).
- Combinación de modelos (Bagging, Boosting ...)
- Visualización anterior (datos en múltiples gráficas) y posterior (árboles, curvas ROC, curvas de coste...).
- Entorno de experimentos, con la posibilidad de realizar pruebas estadísticas (t-test).

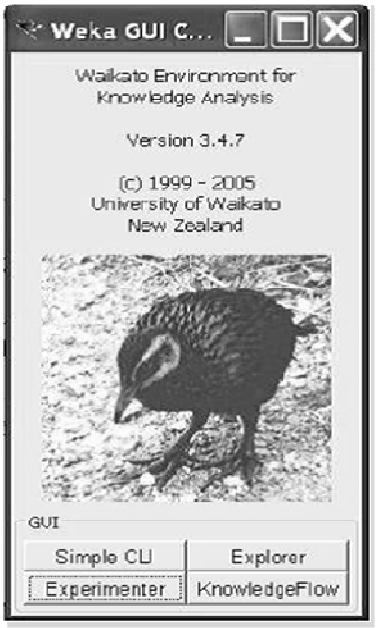

Figura 1.6. Ventana inicial de Weka.

Como se puede ver en la parte inferior de la Figura 1.6, Weka define 4 entornes de trabajo

- **Simple CLI:** Entorno consola para invocar directamente con java a los paquetes de Weka
- **Explorer:** Entorno visual que ofrece una interfaz gráfica para el uso de los paquetes
- **Experimenter:** Entorno centrado en la automatización de tareas de manera que se facilite la realización de experimentos a gran escala.
- **KnowledgeFlow**: Permite generar proyectos de minería de datos mediante la generación de flujos de información.

Explorer.

En esta sección se explicará el entorno Explorer, ya que permite el acceso a la mayoría de las funcionalidades integradas en Weka de una manera sencilla.

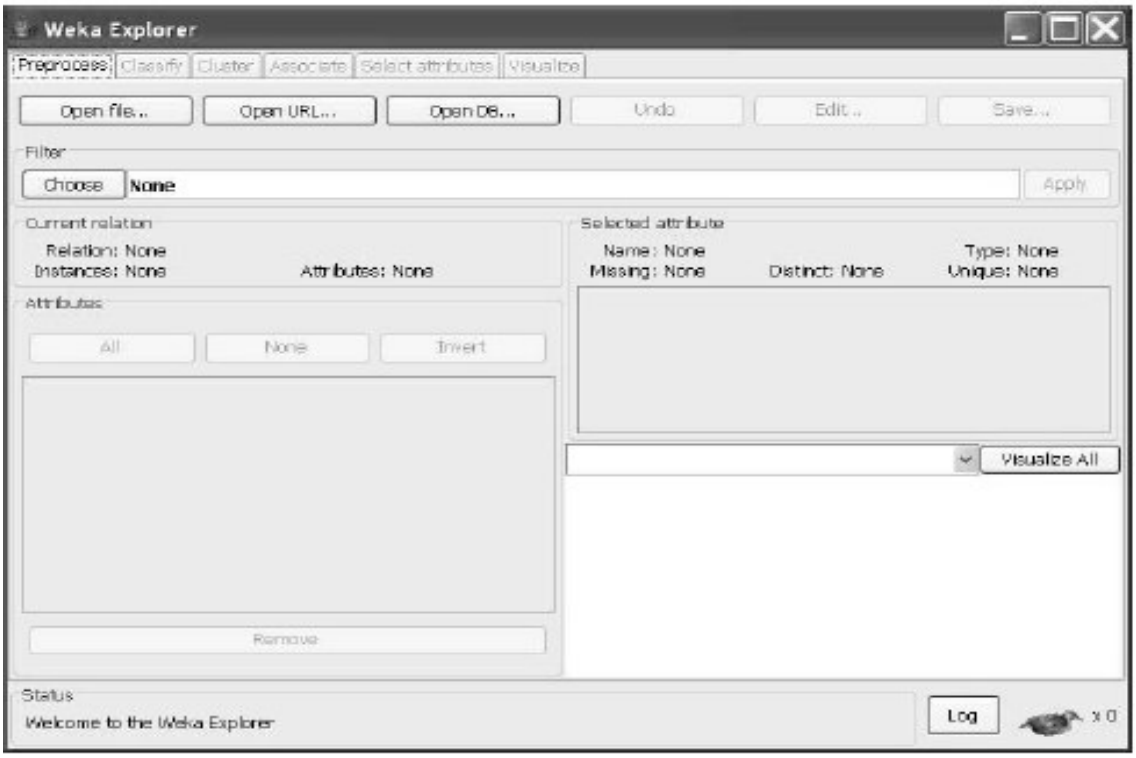

Figura 1.7. Ventana del Explorador

Como se puede observar existen 6 sub-entornos de ejecución:

- **Preprocess:** Incluye las herramientas y filtros para cargar y manipular los datos.
- **Classification:** Acceso a las técnicas de clasificación y regresión.
- **Cluster:** Integra varios métodos de agrupamiento.
- **Associate:** Incluye una pocas técnicas de reglas de asociación.
- **Select Attributes:** Permite aplicar diversas técnicas para la reducción del número de atributos.

- **Visualize:** En este apartado podemos estudiar el comportamiento de los datos mediante técnicas de visualización.

1.5 .4 Ventajas de la herramienta Weka.

El programa WEKA es una herramienta que permite realizar minería de texto con una interfaz gráfica. Además, permite una comparación con los distintos métodos que se utilizan para el pre-procesamiento, clasificación de información, clustering y meta-aprendizaje. WEKA proporciona una plataforma para evaluar un problema con distintas combinaciones de algoritmos y poder extraer conocimiento interesante. Esta herramienta es libre por lo que puede ser utilizada por todos los usuarios, y está validado en proyectos de Minería de Datos y Minería de Texto.

1.6 Corpus de Texto.

Se le denomina corpus de textos a varios documentos de un mismo tópico. Por ejemplo: Un estudiante desea estudiar sobre la asignatura de Inteligencia Artificial, necesitará consultar diferentes documentos que aborden el tópico Inteligencia Artificial.

1.7 Definiciones de suceso y tópico.

En el contexto de esta tesis, una **noticia** es un artículo periodístico o un segmento de transmisión de un medio de comunicación con un enfoque coherente [TDT, 03], en otras palabras, son las narraciones en las que se determina la ocurrencia de sucesos, de hechos.

A lo largo de las investigaciones en TDT se han dado diversas definiciones de sucesos y tópicos. La determinación de los límites de estos conceptos es una tarea extremadamente difícil y arbitraria. El Consorcio de Datos Lingüísticos (LDC, siglas en inglés de *Linguistic Data Consortium*) para facilitar esta tarea ha identificado ciertos tópicos generales y proporcionado un conjunto de reglas para ello [TDT, 03].

Un **suceso** se definió en el estudio piloto TDT1 como algún hecho que ocurre en un instante de tiempo concreto. Por ejemplo, la erupción del Monte Pinatubo el 15 de junio de 1991 es un suceso. Los sucesos pueden ser inesperados, tal como la erupción de un volcán o esperados como una elección política [Alla y TDT2, 98]. Posteriormente, la definición de *suceso* fue extendida para incluir la componente espacial del mismo: *un suceso es algo que ocurre en un instante de tiempo y lugar específicos*. Elecciones específicas, accidentes concretos, crímenes y desastres naturales específicos son ejemplos de sucesos.

En el proyecto TDT2 se amplió la noción de *suceso* a la de *tópico*. Con este propósito un **tópico** se define como una actividad o suceso de especial relevancia, junto con todos los sucesos, hechos o actividades directamente relacionados con él. Esta definición es la que se mantiene en la actualidad [TDT, 03].

El conjunto de acciones conectadas que tienen un enfoque común se define como **actividad** (campañas electorales, investigaciones, labores de socorro en desastres) [Papk, 99], [TDT, 03].

Varias noticias se consideran que abordan el mismo tópico siempre que se conecten directamente al suceso asociado. Así, por ejemplo, una noticia acerca de la búsqueda de los supervivientes de una caída de un avión o una noticia acerca del entierro de las víctimas de ese accidente aéreo, se considerarán que son noticias sobre el suceso de la caída de ese avión. Esto marca una diferencia con el estudio piloto TDT, donde se consideraba que los sucesos consiguientes eran sucesos separados. Obviamente existen límites para esta inclusión. Por ejemplo, noticias sobre la sustitución de la directiva de la línea aérea como consecuencia de las investigaciones sobre el accidente, probablemente no se considerarán noticias sobre el accidente aéreo. Como parte del esfuerzo por hacer más amplia la noción de tópico, los tópicos incluyen, además, las noticias que tengan un enfoque coherente alrededor del tópico, aún cuando no hay ningún suceso subyacente claro [TDT2, 98].

1.7.1 Principales tareas

Las tareas básicas definidas en el estudio TDT son las siguientes [TDT, 03]:

- Segmentación de noticias (*Story segmentation*).
- Detección de tópicos (*Topic Detection*).
- Detección de la primera noticia (*First Story Detection*, FSD – siglas en inglés).
- Seguimiento de tópicos (*Topic Tracking*).
- Detección de enlaces (*Link Detection*).

La figura 1.8 ilustra las nociones básicas de cada tarea [Wayn, 00].

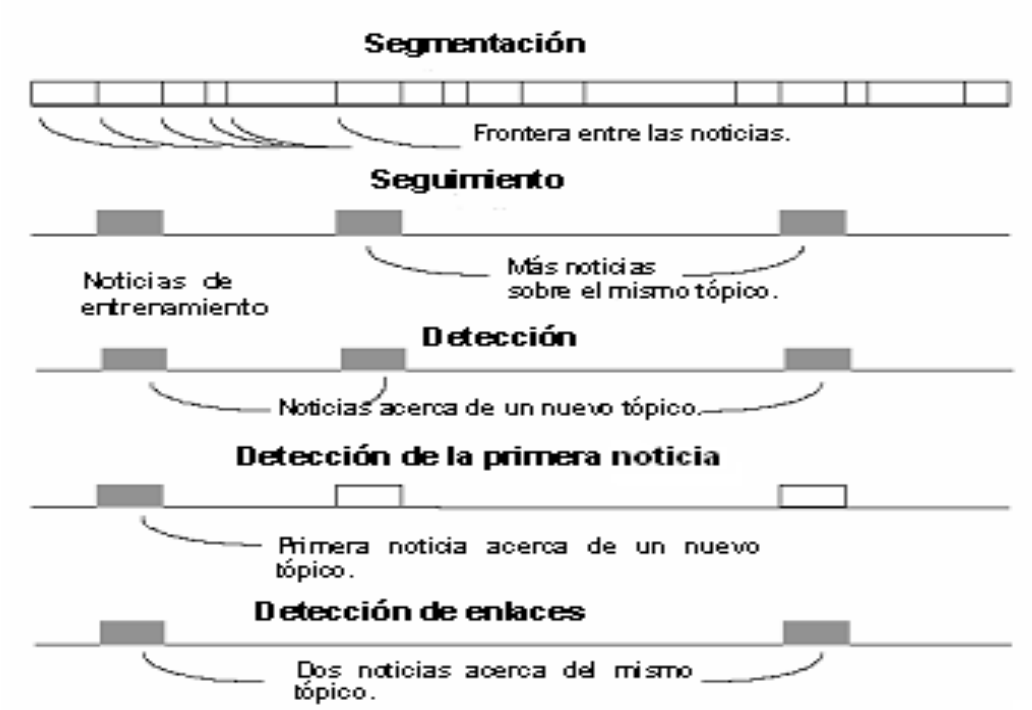

Fig.1.8. - Tareas TDT.

La tarea de ***segmentación*** consiste en segmentar un flujo continuo de textos (incluyendo los hablados) en sus noticias constituyentes, es decir, localizar correctamente las fronteras entre las noticias adyacentes para todas las noticias del corpus. Debido a que las noticias que provienen de textos se proporcionan de forma segmentada, esta tarea, sólo se aplica a las fuentes de audio (radio y televisión).

La tarea de ***detección*** está caracterizada por la carencia de conocimiento sobre el tópico que se desea detectar. La detección es, entonces, el problema de identificar en un flujo de noticias aquéllas que pertenecen a un nuevo tópico o a uno no identificado previamente. En otras palabras, es una tarea de aprendizaje no supervisado, es decir, sin muestra de entrenamiento etiquetada [Alla, 98].

La tarea de detección es una abstracción experimental del agrupamiento de noticias. El objetivo de un sistema de detección es agrupar las noticias que abordan el mismo tópico. Existen dos modos de operación de un sistema de detección de nuevos tópicos: *inmediato* (detección en línea) y *retardado* (detección retrospectiva) [Yang, 98], [Papk, 99]. En el modo inmediato se asume una aplicación en tiempo real que indica si el documento actual aborda o no un nuevo tópico antes de explorar el próximo

documento. En el modo retardado las decisiones de clasificación se realizan cada cierto intervalo de tiempo. Por ejemplo: el sistema puede coleccionar las noticias al finalizar el día y proporcionar los nuevos sucesos que han acontecido ese día.

La *detección retrospectiva* se define como la tarea de identificar todos los tópicos de un corpus de noticias. Este tipo de detección recibe como entrada el corpus completo y obtiene como salida una partición del corpus en grupos de noticias, donde cada grupo representa un tópico. Se asume que cada noticia aborda a lo sumo un tópico. Por lo tanto, cada noticia pertenecerá a un solo grupo [Alla, 98].

La *detección en línea* se define como la tarea de identificar nuevos tópicos en un flujo de noticias. Cada vez que llega una noticia, una decisión de Sí o No debe ser tomada antes de procesar las noticias posteriores. Esta decisión indica si la noticia aborda o no un nuevo tópico. Este tipo de detección recibe como entrada el flujo de noticias en orden cronológico, simulando la llegada en tiempo real de un tópico y obtiene como salida un agrupamiento de las noticias en tópicos.

Ambas formas de detección no tienen conocimiento previo de los tópicos a detectar, aunque pueden tener acceso a las noticias anteriores, de modo que se pueden utilizar para contrastar y determinar cuándo se produce un nuevo tópico.

El agrupamiento de las noticias puede dividirse en dos fases: detectar cuándo aparece un nuevo tópico y colocar las noticias que abordan tópicos previamente analizados en los grupos apropiados. La primera fase es precisamente, la detección de la primera noticia.

La ***detección de la primera noticia*** se define, por tanto, como la tarea de detectar la primera noticia que aborda un tópico previamente desconocido. Esta tarea es la que alerta a un usuario en un sistema TDT cuando un nuevo tópico ocurre. La detección de la primera noticia consiste en observar un flujo de noticias y etiquetar cada noticia como "*primera*" *o* "*no primera*", indicando de esta manera si es la primera noticia que aborda un nuevo tópico [Alla, 00]. Esta tarea está muy relacionada con la de detección. La correcta detección de la primera noticia sobre un tópico dará lugar a que el sistema de detección funcione mejor. Por ello, se ha separado como una tarea independiente.

El problema del ***seguimiento de tópicos*** se define como la asociación automática de las noticias con los tópicos conocidos por el sistema. Un tópico se define como *conocido por el sistema* si tiene asociado un conjunto de noticias que lo abordan. El objetivo de este problema es recuperar las noticias posteriores que pertenecen al tópico de interés. El seguimiento de tópicos es un problema de clasificación supervisada,

donde el usuario del sistema conoce a priori los tópicos que tiene interés de seguir. Por lo tanto, se necesita dividir al corpus de estudio en dos partes: un conjunto de entrenamiento que incluirá las noticias que abordan el tópico que se desea seguir y un conjunto de prueba formado por las noticias que el sistema de seguimiento determinará si abordan o no el tópico de interés.

El problema del seguimiento de tópicos puede ser considerado, además, como un problema de categorización de textos [Yang, 00] con las siguientes restricciones:

- Cada tópico de interés está definido por un conjunto de instancias positivas (documentos) que son manualmente identificadas antes de que el seguimiento comience; ningún otro conocimiento está disponible.
- Tan pronto como un nuevo documento llega, el sistema toma una decisión binaria con respecto a cada tópico definido.
- Cuando se entrena para un tópico, las estimaciones de relevancia para otros tópicos se asumen desconocidas. El usuario de un sistema de seguimiento sólo proporciona un pequeño número de documentos relevantes para el tópico de interés y ninguno para el resto de los tópicos que no son de interés.
- Todo documento que precede al documento que está siendo evaluado puede ser usado como dato de entrenamiento. Sin embargo, sólo las instancias previamente identificadas como positivas están etiquetadas; el resto de los documentos no lo están a pesar de que alguno puede ser realmente una instancia positiva.

Por último, la tarea de ***detección de enlaces*** detecta si dos noticias están "enlazadas" por el mismo tópico. Dos noticias están *enlazadas* si abordan el mismo tópico. A diferencia de las otras tareas de TDT, la detección de enlaces no estuvo motivada por una aplicación hipotética, sino que ella constituye el núcleo a partir del cual se construyen otras tareas de TDT. La tarea de detección de enlaces centra su atención en la comparación de dos noticias.

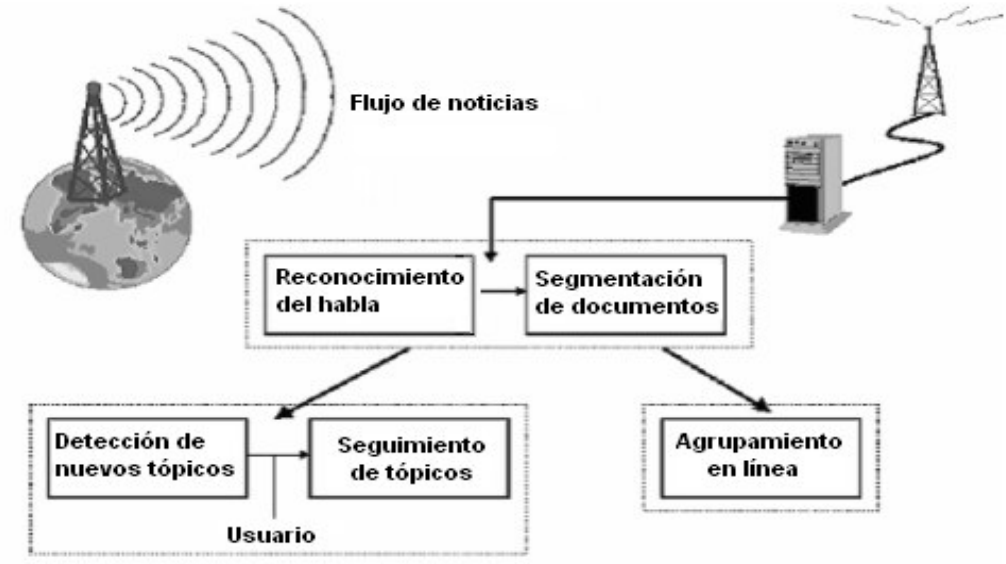

Fig. 1.9.- Arquitectura de un sistema de detección y seguimiento de tópicos.

Una vez analizadas cada una de las tareas de TDT, en la figura 1.9 [Alla, 00] se muestra la arquitectura de un sistema de detección y seguimiento de tópicos. Las noticias provenientes de varias fuentes son seguidas por un sistema. Si la fuente es la radio o la televisión, un reconocedor automático del habla convierte la señal de audio a texto y, luego, mediante técnicas de segmentación se encuentran las fronteras de las noticias (documentos). El sistema de detección sigue el flujo de noticias y alerta al usuario sobre los nuevos tópicos. Si el documento que aborda el nuevo tópico es de interés para el usuario, se inicia un proceso de seguimiento del tópico mediante la creación de un clasificador que tomará decisiones en línea acerca de los siguientes documentos que aparecen en el flujo de noticias. Adicionalmente, el usuario puede proporcionar el tópico que desea seguir, brindando alguna información para especificar dicho tópico.

1.8 Algoritmos para la Detección y Seguimiento de tópicos.

Para los problemas de Detección y Seguimiento de tópicos se utilizaron los siguientes algoritmos:

1.8.1 Algoritmos de Agrupamiento.

Un algoritmo de agrupamiento (en inglés, clustering) es un procedimiento de agrupación de una serie de vectores según criterios habitualmente de distancia; se tratará de disponer los vectores de entrada de forma que estén más cercanos aquellos que tengan características comunes. Un algoritmo de clustering permite extraer representantes de un conjunto de datos, que pueden ser posteriormente usados para transmisión, para eliminación de ruido o con una fase posterior de calibración, para clasificación de vectores en diferentes conjuntos.

Algunos algoritmos de clustering conocidos y usados son el k medias, el ISODATA y el Mapa autoorganizado de Kohonen. [2]

1.8.2 Algoritmos de Clasificación.

El proceso de crear una clasificación automática de textos consiste en descubrir variables que sean útiles en la discriminación de los textos que pertenecen a clases pre-existentes distintas. En particular, los clasificadores (programas que ejecutan algoritmos de clasificación) son entrenados en un grupo de documentos, previamente clasificados y etiquetados acorde a algún criterio particular (tema, materia, origen, etc.), conformando una clase. De esta manera, el objetivo de estos clasificadores es decidir en qué categoría debe ir cada texto nuevo, partiendo de un esquema de clasificación previo [Figuerola, Zazo y Berrocal, 2000]. También se dice que la clasificación o categorización automática de documentos puede ser entendida como una tarea en la cual, en base a la identificación por medios matemático-estadísticos, un documento nuevo es asignado a una clase particular de documentos pre-existentes [Jurafsky y Martin, 2000].[2]

[2] http://es.wikipedia.org/wiki/Algoritmo_de_agrupamiento

1.8 Conclusión:

- La inteligencia artificial es una de las ciencias de la computación más prometedoras que existen en la actualidad.
- La lingüística computacional constituye una de las tendencias más complejas de la inteligencia artificial y a la vez de las más importantes debido a la necesidad de comprender el lenguaje y generar posibles del mismo de los sistemas inteligentes.
- El procesamiento del lenguaje natural es la rama ingenieril de la lingüística computacional
- Existen diversos algoritmos para los problemas de Detección y Seguimiento de tópicos.

CAPÍTULO II

Capítulo2: Diferentes técnicas y algoritmos para el agrupamiento y clasificación de textos.

Introducción

Para la Detección y Seguimiento de tópicos se utilizan diferentes técnicas de la Minería de Texto como son los algoritmos de agrupamiento y clasificación. Los algoritmos de agrupamiento son aquellos que consisten en la segmentación de un grupo diverso de noticias en subgrupos. Los grupos que se obtienen no tienen un significado a priori por lo que no se conoce la cantidad ni el significado antes de conocer el algoritmo. Los algoritmos de clasificación consisten en examinar características de un objeto (registro), asignándole una clase predefinida o sea que decide en qué categoría debe ir cada texto nuevo.

Los algoritmos de agrupamiento y clasificación analizados fueron seleccionados cuidadosamente después de haber realizado un estudio previo de sus características generales y específicas, entre los que se encuentran: Single-Pass, 1-NN, K-NN, K-Means Incremental y Variante del Single-Pass (Scatter/Gather), K-Means y Expectativa Máxima. En este capítulo se explica la metodología utilizada en la construcción de los corpus de texto,

Posteriormente se procede al cálculo de las medidas de la calidad, F1 y Coste de Detección, donde se tomaron en cuenta diferentes aspectos como la Relevancia, Precisión, Probabilidad de omisiones y Probabilidad de falsas alarmas.

2.1 Metodología para la confección de un corpus de texto.

2.2.1 Corpus de texto utilizados.

En la selección de las palabras claves se determina además la frecuencia con que aparece esa palabra, la línea y el párrafo donde se encontraba.

Para la realización de esta tesis se utilizaron 4 corpus de texto, TDT-A, TDT-B, TDT-C, TDT-D, de los cuales escogimos 5 noticias con sus respectivos tópicos.

TDT-A

Tópico	Titular
Conflicto	Milosevic dice aceptar las condiciones del G-8.
Espionaje	Dimite el ministro de Defensa portugués por filtrar un informe sobre el espionaje.
Bombardeo	La OTAN hace una evaluación triunfal de los 70 días de bombardeo.
Pinochet	El Gobierno chileno dice que Pinochet debe volver porque hay garantías de que será juzgado.
Bombardeo	Muere un general serbio en un bombardeo.

TDT-B

Tópico	Titular
Asesinato	Ocalan implica a su ex mujer en el asesinato de Olof Palme.
Elecciones	Mandela pide a Sudáfrica votar hoy sin violencia en las elecciones de su despedida.
Mitch	Centroamérica recibirá 9.000 millones de dólares parapara superar los estragos del “Mitch”.
Campaña	La oposición pide al presidente mexicano si hubo dinero negro en las en su campaña.
Elecciones	Thabo Mbeki nuevo presidente electo de Sudáfrica.

TDT-C

Tópico	Titular
Conflicto	India descarta el uso de armas nucleares en el conflicto de Cachemira.
Violencia	La violencia se extiende a varias regiones del país.

Narcotráfico	El espionaje de EE UU vincula a dos altos políticos de México con el narcotráfico.
Cuba	Cuba reclama a Washington 181.000 millones por su “agresión” desde 1959.
Balseros	EE UU deporta a 99 balseros cubanos en un mismo día.

TDT-D

Tópico	**Titular**
Chávez	El presidente Chávez decreta la enseñanza militar en niños y jóvenes en Venezuela.
Guerrilla	La guerrilla colombiana de ELN pospone la liberación de los rehenes.
Guerrilla	La guerrilla colombiana liberara a ocho rehenes en medio de violentos ataques.
Papa	El Papa anuncia su deseo de viajar a Tierra Santa y a Irak en el 2000.
Chávez	La ministra de Hacienda venezolana rompe con Chávez.

Se establecieron los diferentes tópicos para posteriormente analizar un grupo significativo de noticias de 4 agencias de noticias que fue el criterio que se utilizó para obtener cada corpus de texto, para a partir de ahí asignar a cada noticia un tópico y seguidamente hacer un resumen y determinar las palabras claves de las noticias y llevarlas a un formato txt, donde cada una aparece con un identificador.

Ejemplo:

CNN TDT-A 01: |gobierno:a:1:9 1 29#Chile:n:1:4 1 23# jusgado:n:5:2 1 2%6 1 2%7 1 17%8 1 2%9 1 3#juicio:n:1:3 1 12#civil:a:1:6 1 29# Pinochet :np:1:6 1 14#continuar:v:1:2 1 18#concluir:v:1:7 1 15#proceso:n:1:9 2 2#preparación:n:1:5 1 10#experto:n:1:9 2 9#prueba:n:1:9 1 16#desarrollar:v:1:4 1 11#cumplir:v:3:4 2 7%5 1 8%9 1 6#mandatario:a:1:2 1 32#defensa:n:2:2 1 26%7 1 9#sentencia:n:1:9 1 23#demanda:n:1:6 1 28#menor:a:1:3 1 13#juzgado:n:1:3 1 22#malversación:n:1:2 1 12#venezolano:a:1:2 1 8#formulación:n:1:5 1 13#juicio:n:2:2 1 4%9 1 #ministro:n:1:3 1 5#presidencia:n:1:3 1 16#cargo:n:2:5 1 15%6 1 18#presunto:a:1:2 1 11#instrucción:n:1:4 1 6#peculado:n:1:2 1 15#sumario:n:1:4 1 8# #prisión:n:1:4 2 8#sostener:v:2:2 1 29%7 1 7#reanudar:v:1:8 1 4#lectura:n:1:6 1 10#abogado:n:1:8 1 13#dictar:v:2:4 1 19%9 1 21#comenzar:v:1:6 1 4#preventivo:a:1:4 2 9#público:n:4:2 1 27%6 1 3%7 1 10%9 1 4#acusado:n:1:4 1 26#oral:a:1:9 1 9#presentación:n:2:6 1 25%9 1 14#medidodía:n:1:7 1 2 #promoción:n:1:9 1 12#escribir:v:1:6 1 16#orden:n:1:4 1 21#agravado:a:1:2 1 13#proseguir:v:1:6 1 20#presidente:n:1:2 1 7#fase:n:1:4 1 3#firme:a:1:9 1 27#intervención:n:1:8 1 10#tardar:v:1:9 2 4#diferencia:n:1:4 2 12

CNN TDT-A 01: Nombre del fichero de la noticia.

gobierno: Término.

a, n, v, np: Categoría gramatical (adjetivo, sustantivo, verbo, sustantivo predicado).
1: frecuencia, cantidad de veces que aparece en el texto.
9 1 29: 9(Párrafo) 1(Oración dentro del párrafo) 29(Palabra dentro de la oración).

2.2 Medidas de evaluación de la calidad.

En la metodología de evaluación de TDT se han definido un conjunto de medidas para evaluar la calidad de los sistemas de detección de tópicos. En ellas se comparan los grupos obtenidos por el sistema con un conjunto de clases obtenidas manualmente por un experto. Dos de las medidas más utilizadas son: la *medida F1* [Rijs, 79] y el *Coste de Detección* [TDT2, 98].

La medida F1 es ampliamente utilizada en los Sistemas de Recuperación de Información. Esta medida combina los factores de *precisión* y de *relevancia* en la Recuperación de Información.

En TDT existen, además, dos tipos de errores: los errores por omisión (*misses*) y las falsas alarmas (*false alarms*) [TDT2, 98]. En la detección de nuevos tópicos, los errores por omisión ocurren cuando el sistema no detecta un nuevo tópico y las falsas alarmas ocurren cuando el sistema indica erróneamente que un documento describe un nuevo tópico. Las falsas alarmas y las omisiones se calculan sobre la base de los tópicos conocidos (identificados manualmente) y los tópicos identificados por el sistema.

Sea la tabla 2.1 siguiente [Alla, 00]:

	Relevantes	No Relevantes
Recuperados	*a*	*B*
No Recuperados	*c*	*D*

donde los documentos recuperados son los que el sistema ha clasificado como instancias positivas de un tópico y los relevantes son los que manualmente se han declarado como relevantes a un tópico dado. Así, se define para cada tópico las siguientes medidas de calidad [Yang, 99]:

- **Relevancia**: es el número de asignaciones correctas hechas por el sistema dividido por el total de asignaciones correctas.

  $$Relevancia = \frac{a}{a+c}$$ si $a + c > 0$. En otro caso, está indefinida.

- **Precisión**: es el número de asignaciones correctas hechas por el sistema dividido por el número total de asignaciones del sistema.

  $$Precisión = \frac{a}{a+b}$$ si $a + b > 0$. En otro caso, está indefinida.

- **Probabilidad de omisiones**: $P_m = \frac{c}{a+c}$ si $a + c > 0$. En otro caso, está indefinida.

- **Probabilidad de falsas alarmas**: $P_{fa} = \frac{b}{b+d}$ si $b + d > 0$. En otro caso, está indefinida.

Nótese que todas las medidas anteriores están comprendidas entre 0 y 1.

Una forma alternativa de calcular las medidas de precisión y relevancia consiste en considerar el tamaño del grupo obtenido por el sistema (n_j), el tamaño de la clase construida manualmente (n_i) y el tamaño de la intersección de ambos (n_{ij}) [Pons, 02]:

$$Relevancia(i,j) = \frac{n_{ij}}{n_i}$$

$$Precisión(i,j) = \frac{n_{ij}}{n_j}$$

Tradicionalmente, estas dos medidas se combinan en una sola para dar un indicador global de la calidad de un sistema de recuperación. La medida combinada más utilizada en los Sistemas de Recuperación de la Información es la medida F1, que se define como [Pons, 02]:

$$F1(i,j) = 2 \cdot \frac{Relevancia(i,j) \cdot Precisión(i,j)}{Relevancia(i,j) + Precisión(i,j)} = 2 \cdot \frac{n_{ij}}{n_i + n_j}$$

Hasta aquí se han visto varias medidas que indican el grado de emparejamiento entre cada grupo generado por el sistema y los construidos manualmente. Para obtener una medida global del sistema será necesario asociar cada grupo del sistema con la clase manual (tópico) que maximice la medida a evaluar. Para ello se utiliza la siguiente función:

$$\sigma(i) = \arg\max_j \{F1(i,j)\}$$

Una medida de calidad global puede ser calculada de dos formas: macro-promediada (macro-averaging) o micro-promediada (micro-averaging). Existe una distinción importante entre estas dos formas. La micro-promediada le da el mismo peso a cada documento y, por tanto, se considera un promedio por documento, es decir, un promedio sobre todos los pares documento/tópico. Por otra parte, la macro-promediada da un peso similar a cada tópico sin tener en cuenta su frecuencia, por lo que se considera un promedio por tópico [Yang, 97].

Así, la *medida F1 micro-promediada* se calcula, de la siguiente forma:

$$microF1 = 2 \cdot \frac{microP \cdot microR}{microP + microR}$$

$$microP = \frac{1}{N_{tópicos}} \sum_{i=1}^{N_{tópicos}} \frac{n_{ij}}{n_{\sigma(i)}}$$

$$microR = \frac{1}{N_{tópicos}} \sum_{i=1}^{N_{tópicos}} \frac{n_{ij}}{n_i}$$

La medida *F1 macro-promediada* se calcula como la media de la medida F1 evaluada en cada par óptimo clase-grupo:

$$macroF1 = \frac{1}{N_{tópicos}} \sum_{i=1}^{N_{tópicos}} F1(i, \sigma(i))$$

Para evaluar la efectividad de la detección, en las evaluaciones de TDT, se usa la función de coste de detección, la cual combina los errores por omisión y las falsas alarmas. El coste de detección entre un tópico i y un grupo j obtenido por el sistema se define como [TDT2, 98]:

$$C_{DET}(i, j) = coste_{f_a} \cdot P_{f_a}(i, j) \cdot (1 - P_{tópico}) + coste_m \cdot P_m(i, j) \cdot P_{tópico}$$

El coste de detección también puede calcularse a partir del tamaño del grupo obtenido por el sistema (n_j), el tamaño de la clase construida manualmente o tópico (n_i), el tamaño de la intersección de ambos (n_{ij}) y el número total de documentos de la colección (N_{docs}). Así, las probabilidades de que el sistema produzca una falsa alarma y un error por omisión se calculan como sigue [Pons, 02]:

$$P_{f_a}(i, j) = \frac{n_j - n_{ij}}{N_{docs} - n_i}$$

$$P_m(i, j) = \frac{n_i - n_{ij}}{n_i}$$

Los parámetros Ptópico (probabilidad a priori de que un documento sea relevante a un tópico), costefa y costem se determinan empíricamente a partir de un corpus de entrenamiento. En la evaluación de TDT-B se usó Ptópico = 0.02 y las constantes costefa=costem = 1 [TDT2, 98], mientras que en las evaluaciones posteriores de TDT se usó costefa = 0.1 y costem = 1[TDT, 03].

Nuevamente, para definir una medida global del coste de detección, cada tópico debe hacerse corresponder con el grupo que produce el mínimo coste de detección, mediante la función:

$$\sigma(i) = \arg\min_{j}\{C_{DET}(i,j)\}$$

En TDT2 también se definen varias formas de calcular el coste de detección global según sea la forma de calcular las probabilidades P_{fa} y P_m. Así, si el método asigna igual peso a cada decisión por cada noticia y acumula los errores por todos los tópicos se denomina *story-weighted* (o micro-promediado). En este método las probabilidades de falsas alarmas y de omisiones se calculan de la siguiente forma:

$$P_m = \frac{1}{N_{docs}} \sum_{i=1}^{N_{tópicos}} (n_i - n_{i\sigma(i)})$$

$$P_{fa} = \frac{\sum_{i=1}^{N_{tópicos}} (n_{\sigma(i)} - n_{i\sigma(i)})}{\sum_{i=1}^{N_{tópicos}} (N_{docs} - n_i)}$$

Si, por el contrario, se acumulan los errores de forma independiente en cada tópico y se promedian los errores de todos los tópicos, considerando que tienen el mismo peso, el método se denomina *topic-weighted* (o macro-promediado). Aquí, las probabilidades se calculan así:

$$P_m = \frac{1}{N_{tópicos}} \sum_{i=1}^{N_{tópicos}} \frac{n_i - n_{i\sigma(i)}}{n_i}$$

$$P_{fa} = \frac{1}{N_{tópicos}} \sum_{i=1}^{N_{tópicos}} \frac{n_{\sigma(i)} - n_{i\sigma(i)}}{N_{docs} - n_i}$$

Para la detección de tópicos se prefiere utilizar el método *topic-weighted*, atendiendo al número pequeño de tópicos y a su naturaleza heterogénea [TDT2, 98]. En las evaluaciones actuales de TDT se utiliza exclusivamente este método.

Debido a que los costes de omisiones y de falsas alarmas, así como la probabilidad a priori de un tópico varían según la aplicación, el coste de detección ha sido normalizado mediante la siguiente fórmula:

$$(C_{DET})_{Norm} = \frac{C_{DET}}{\min\{coste_m \cdot P_{tópico}, coste_{fa} \cdot (1 - P_{tópico})\}}$$

Adicionalmente a estas medidas, se utiliza la curva denominada DET (*Detection Error Tradeoff*) [Mart, 97], la cual visualiza la relación entre el porcentaje de los errores por omisión y de las falsas alarmas en

las evaluaciones de la detección de nuevos tópicos. En la curva DET se da un tratamiento uniforme a los errores por omisión y a las falsas alarmas y, en lugar, de graficar las probabilidades de ambos tipos de errores, se grafican las desviaciones normales que corresponden a esas probabilidades. La curva *y=-x* representa el comportamiento aleatorio del sistema.

El uso de la escala de desviación normal provoca que la curva se desplace hacia el cuadrante inferior izquierdo cuando la eficiencia del sistema es alta, lo cual facilita las comparaciones. Cuando se comparan dos curvas DET de las aproximaciones *A* y *B* al problema de detección de nuevos tópicos, se define que la aproximación *A* es mejor que la *B* cuando todos los puntos de la curva DET de *A* están por debajo de la curva DET de *B* [Papk, 99]. Ejemplos de curvas DET se muestran en la figura 2.1 [Papk, 99]. .

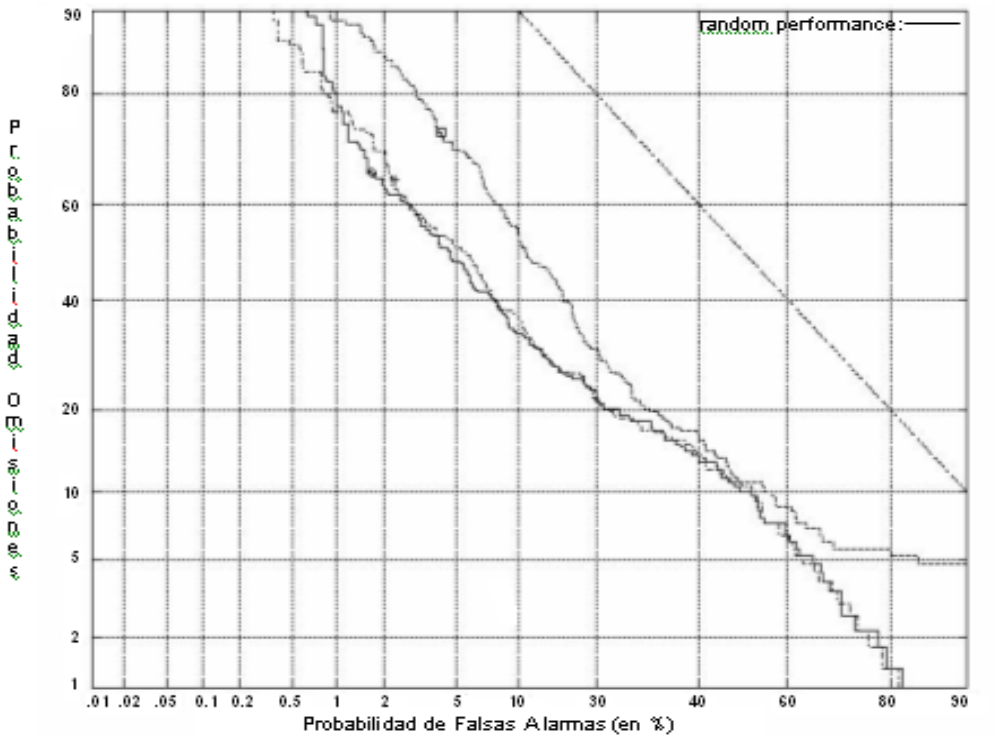

Fig. 2.1.- Ejemplos de Curvas DET.

2.3 Estructura general de un Sistema de Detección

En esta tesis, se considera que un sistema de detección de tópicos genérico está caracterizado por los siguientes elementos:

- Un modelo de representación de los documentos, en este caso, de las noticias.
- Una medida de semejanza para comparar noticias, de tal modo que determine cuándo tratan el mismo suceso o tópico.
- El tratamiento de las propiedades temporales de las noticias.
- Un algoritmo de agrupamiento o clasificación que permita agrupar las noticias en tópicos.

En los siguientes epígrafes se describen las principales propuestas en la literatura en cada uno de estos elementos.

2.3.1 Modelos de representación

Para la representación de los documentos es muy común utilizar el *modelo vectorial*. El modelo vectorial [Ragh, 86], [Salt, 89] está basado en que cada documento de la colección está representado por un vector *n*-dimensional (*n* es la cardinalidad del conjunto de términos de indexación elegido para toda la colección de documentos), en el que cada componente representa el peso del término asociado a esa dimensión. Este peso representa un estimado (usualmente estadístico, aunque no necesariamente) de la utilidad del término como descriptor del documento, es decir, de la utilidad para distinguir ese documento del resto de los documentos de la colección [Gree, 00].Un término recibe un peso de 0 en los documentos en los cuales éste no ocurre. Normalmente los términos muy comunes y los poco frecuentes son eliminados y las formas diferentes de una palabra son reducidas a su forma canónica. La mayoría de los vectores de documentos son dispersos.

Sea ζ una colección formada por N documentos. Un documento se representa, entonces, según el modelo vectorial, como un vector $d = (w_1, w_2, ..., w_n)$, donde n es el número total de términos de ζ y w_i es el peso del i-ésimo término en el documento d ($w(t_i,d)$).

La longitud de un documento d, que denotaremos como $len(d)$, es la cantidad total de términos del documento.

Una representación alternativa al modelo vectorial que está teniendo un gran auge es el basado en los *modelos de lenguaje*. Un modelo de lenguaje *M* permite estimar la probabilidad de observar o generar una frase *s* con dicho modelo, denotado *P*(*s*|*M*). Aplicado a un Sistema de Recuperación de la Información, la semejanza entre una consulta *Q* (vista como una secuencia de términos) y cada documento *d* (visto como una distribución probabilística sobre los términos de la colección) se asocia a la probabilidad de generar la consulta *Q* con el modelo de lenguaje representado por cada documento *d*, denotado P(*Q*| *d*).

En la práctica los modelos de lenguajes utilizados en estas aplicaciones asumen la independencia de los términos, al igual que el modelo vectorial, y por lo tanto pueden representarse como unigramas, donde $P(t_i \mid d)$ representa la probabilidad de observar el término t_i con el modelo del documento *d*. De esta forma, la semejanza entre la consulta y cada documento puede calcularse basándose en la distribución multinomial del siguiente modo:

$$P(t_1 \cdots t_k \mid d) = \prod_{i=1}^{k} P(t_i \mid d) \qquad (1)$$

El cálculo de las probabilidades P(t_i| *d*) suele estimarse a partir de la frecuencia media del término t_i en el documento (máxima verosimilitud), y aplicando alguna técnica de suavizado para evitar las probabilidades nulas. El método de suavizado más utilizado es la interpolación lineal, que se expresa como sigue:

$$P(t \mid d) = \lambda \frac{TF(t,d)}{len(d)} + (1-\lambda) \cdot P(t) \qquad (2)$$

donde *TF*(*t*, *d*) es la cantidad de veces que ocurre el término *t* en el documento *d*, *P*(*t*) es la probabilidad del término *t*, que se calcula a partir de una colección suficientemente extensa (*background*) y λ es el parámetro de suavizado que debe ajustarse experimentalmente.

2.3.2 Esquemas de pesado de términos

Existen diversas técnicas para asignar pesos a los términos, entre las que podemos mencionar las siguientes:

- *Booleano*, donde los pesos $w_i \in \{0,1\}$ indican la presencia o ausencia del término t_i en el documento.

- *Frecuencia de un término* o *TF* (*Term Frequency*) [Salt89]. Cada término tiene una importancia proporcional a la cantidad de veces que aparece en un documento, denotado *TF*(*t*,*d*). El peso de un término *t* en un documento *d* es *w*(*t*,*d*) = *TF*(*t*,*d*).

  Hay que señalar que es muy importante normalizar de alguna manera la frecuencia de un término en un documento para moderar el efecto de las altas frecuencias (por ejemplo, el término *la* que aparece 20 veces no es más importante que el término *telecomunicaciones* que aparece 4 veces) y para compensar la longitud del documento (en documentos más largos, previsiblemente aparecerá más veces cada término). El propósito de la normalización es lograr que el peso o importancia de un término no dependa de la frecuencia de su ocurrencia relativa con los otros términos. Pesar un término por la frecuencia absoluta obviamente tiende a favorecer los documentos más extensos sobre los menos extensos.

  Existen dos tipos de normalización de las frecuencias:

  - *Normalización de la frecuencia del término*: El *TF* se divide por la frecuencia máxima de todos los términos, logrando que el peso esté entre 0 y 1. Algunas variantes de este tipo de normalización son:

    - La *Normalización Aumentada de la Frecuencia*, que consiste en normalizar entre 0.5 y 1 mediante la fórmula:

      $$w(t,d) = 0.5 + \frac{0.5 \cdot TF(t,d)}{\max_{d' \in \zeta}(TF(t,d'))}$$

    - La *Frecuencia del Término Logarítmica*, que consiste en $1 + \log TF(t,d)$. Este método reduce la importancia de la frecuencia absoluta de un término en aquellas colecciones con gran variabilidad en la longitud de los documentos. Además reduce el efecto de un término con una inusual alta frecuencia dentro de un documento.

    - Normalizar en función del *promedio* de las frecuencias:

      $$w(t,d) = \frac{1 + \log(TF(t,d))}{1 + \log(\underset{d' \in \zeta}{avg}(TF(t,d')))}$$

- *Normalización por la longitud del vector*. Consiste en dividir cada frecuencia por la longitud del documento. Otra variante es la *Normalización del Coseno*, que consiste en dividir cada *TF* por la norma euclidiana del vector.

Un análisis detallado de estos métodos de normalización puede verse en [Gree, 00].

- *TF-IDF.* Mientras el factor *TF* tiene que ver con la frecuencia de un término en un documento, el *IDF* (*Inverse Document Frequency*) tiene que ver con la frecuencia de un término en un conjunto de documentos. Así, la importancia de un término es inversamente proporcional al número de documentos que lo contiene:

  $w(t,d) = TF(t,d) \cdot IDF(t)$

  $$IDF(t) = \log\left(\frac{N}{df(t)}\right)$$

  donde $df(t)$ es el número de documentos que contienen a t en la colección ζ.

  Es decir, mientras menos documentos contengan al término *t* mayor es su *IDF*(*t*). Por el contrario, si todos los documentos de la colección contienen al término *t* entonces *IDF*(*t*) es cero. El factor *TF*(*t*,*d*) contribuye a mejorar la relevancia y el factor *IDF*(*t*) contribuye a mejorar la precisión, pues representa la especificidad del término, distinguiendo los documentos en los que éste aparece de aquellos en los que no aparece. El *IDF*(*t*) es útil como indicador de la bondad del término *t* como discriminador de documentos. Esto expresa la idea intuitiva de que un término que ocurra en toda la colección no es útil para distinguir documentos relevantes de los no relevantes. Por ejemplo, en una colección de documentos que hablan sobre Ciencia de la Computación el término *Computadora* probablemente se mencionará en todos ellos y, por tanto, no será bueno para discriminarlos. Por otra parte, si la colección de documentos abarca una temática muy general, entonces sí será útil para distinguir a los documentos que hablen sobre Ciencia de la Computación.

  La combinación del *TF* con el *IDF* da mayor importancia a los términos que ocurren frecuentemente en el documento e infrecuentemente en la colección.

  Cuando se trabaja en la detección en línea, existe una restricción importante: no puede usarse ninguna información sobre las noticias posteriores a la que se está procesando en ese momento. Esto trae como consecuencia que haya que analizar cómo crece el vocabulario del corpus y cómo

se actualizan dinámicamente los pesos de los términos y la normalización de los vectores de documentos. Existen dos aproximaciones a este problema:

- Obtener un vocabulario fijo y unos pesos estáticos de los términos a partir de un corpus retrospectivo similar, y usar éstos para formar los nuevos grupos del corpus en línea. A los términos nuevos que estén fuera del vocabulario fijado se les da un peso constante o se usa algún tipo de método de suavizado de los pesos de los términos.
- Actualizar incrementalmente el vocabulario y el peso de los términos cada vez que un nuevo documento es procesado. Un análisis empírico muestra que una actualización dinámica de los pesos *IDF* puede ser efectiva en la recuperación de documentos después de procesar un número suficiente de documentos [Call, 96].

- El pesado *ltc* es una variante del esquema *TF-IDF* que fue implementada en el sistema SMART 11.0 [Salt, 89] y se define como:

$$w(t,d) = (1 + \log_2(TF(t,d)) \cdot \frac{IDF(t)}{\|d\|}$$

donde $IDF(t) = \frac{N}{df(t)}$ y $\|d\|$ es la norma euclidiana del documento *d*.

- El pesado de *Okapi tf* es otra variante del esquema *TF-IDF* que fue introducida por Robertson en el sistema *Okapi* [Robe, 95] y se define como:

$w(t,d) = TF_{comp}(t,d) \cdot IDF_{comp}(t)$

donde:

$$TF_{comp}(t,d) = \frac{TF(t,d)}{TF(t,d) + 0.5 + 1.5\frac{len(d)}{\underset{d' \in \zeta}{avg}(len(d'))}} \qquad (3)$$

$$IDF_{comp}(t) = \frac{\log(\frac{N}{df(t)})}{\log(N+1)} \qquad (4)$$

El factor TF_{comp} diferencia las distintas ocurrencias de un término en un documento. Así, le asigna un mayor peso a la primera ocurrencia del término y cada vez menos peso a sus restantes ocurrencias. Además, el cociente de la longitud del documento y el promedio de las longitudes de los documentos de la colección garantiza que una ocurrencia de un término tenga más peso en los documentos pequeños que en los más extensos. Por otra parte, el IDF_{comp} es el logaritmo de la frecuencia inversa del término en la colección, normalizado entre 0 y 1.

2.3.3 Procesamiento de los documentos

Como se ha mencionado anteriormente, las palabras muy frecuentes tienen poco poder discriminante y, por otra parte, las palabras raras carecen de significado estadístico. Es necesario, por tanto, elevar de alguna manera el poder de significación de los términos útiles, aplicando técnicas de indexación. La figura 2.2 [Rijs, 79] muestra las técnicas de indexación más utilizadas.

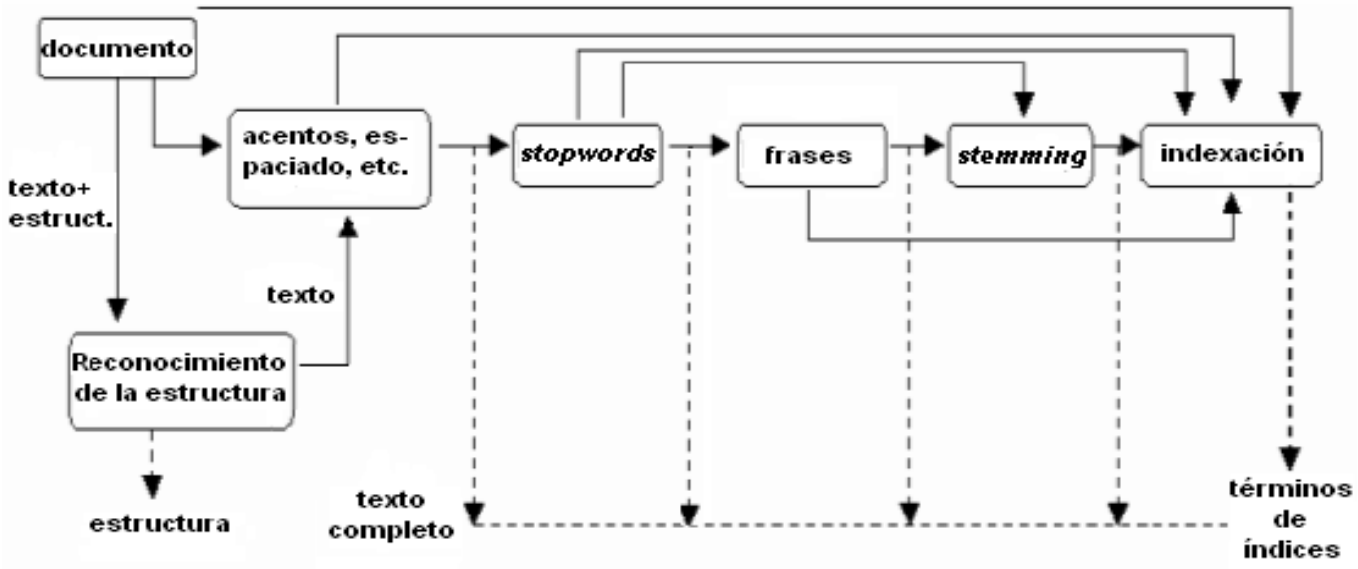

Fig. 2.2.- Técnicas de indexación.

La primera técnica utilizada suele ser la identificación de los signos de puntuación y espaciados, la eliminación de los acentos (uno de los eternos problemas en el procesamiento del lenguaje natural en español), la reducción de las mayúsculas, el reconocimiento del formato del documento (por ejemplo, si es una página HTML se eliminan las etiquetas), el reconocimiento de las palabras, etc. Cuando termina esta etapa tenemos el texto plano y las palabras identificadas en él.

La lista de palabras vacías (*stopwords*), también llamada lista de parada o antidiccionario, es una relación de términos considerados como valores no indexables, usados para eliminar potenciales términos de indexación. Los términos de la lista de parada están carentes de todo significado a la hora de recuperar información, como, por ejemplo, el artículo "*la*" no posee ninguna funcionalidad en la recuperación de documentos, ya que en todos los documentos de la base de datos aparecerá este término de forma casi segura y no resalta nada el contenido del documento almacenado. Así, cada término potencial de indexación es comprobado previamente, verificándose su presencia en la lista de parada y es descartado si se encuentra en ella. Esta lista está formada por las preposiciones, conjunciones, artículos, pronombres, así como aquellas palabras que no son discriminatorias por su elevada frecuencia de aparición en la colección de documentos. Con la eliminación de las palabras vacías se logra una reducción del documento entre un 30 y un 50% [Rijs, 79].

Otra de las técnicas de indexación es la identificación de estructuras multipalabra como, por ejemplo, frases sustantivas, nombres propios de personas, lugares, organizaciones, etc.

Los algoritmos de extracción de raíces o lemas (*stemming* o lematización) se encuentran orientados a obtener un único término a partir de diferentes palabras que constituyen esencialmente variaciones morfológicas con un mismo significado. Por ejemplo, se puede considerar la obtención del término *niño* a partir de *niños* y *niñita*. En el caso de los verbos, por ejemplo, se obtiene el infinitivo *amar* a partir de *amo* y *amará*. El resultado del algoritmo de lematización debe ser una misma forma canónica para las diferentes variaciones morfológicas de una palabra, que no tiene por qué ser, necesariamente, la raíz lingüística. Este proceso comprende la eliminación de los plurales, de ciertos prefijos y sufijos, de las conjugaciones verbales y su reducción al infinitivo, etc.

No necesariamente siempre se aplican todas las técnicas analizadas anteriormente. Por ejemplo, una vez que un elemento de indexación ha pasado el filtro de las palabras vacías, puede ir directamente al índice.

2.3.4 Medidas de semejanza.

Para determinar si dos noticias abordan o no el mismo tópico es necesario definir una medida de semejanza que exprese el grado de parecido entre ellas. Es muy usual en los sistemas de detección usar la *medida del coseno* o variantes de ella. Esta medida se define de la siguiente forma:

$$sem(d_i, d_j) = cos(d_i, d_j) = \frac{d_i \cdot d_j}{\|d_i\| \|d_j\|} = \frac{\sum_{k=1}^{n} (w_k^i \cdot w_k^j)}{\sqrt{\sum_{k=1}^{n} w_k^{i\,2} * \sum_{k=1}^{n} w_k^{j\,2}}}$$

donde w_k^i es la *k*-ésima componente del vector del documento d_i.

Otra medida de semejanza es la *suma pesada*, la cual fue utilizada en el sistema *InQuery*, y se define como:

$$sem(d_i, q_j) = \frac{\sum_{k=1}^{n} w_k^i \cdot w_k^j}{\sum_{k=1}^{n} w_k^j}$$

donde q_j representa el vector de una consulta y d_i, el vector de un documento. Esta función de semejanza tiende a obtener mejores resultados en la medida que la dimensionalidad de q_j sea menor y que sea combinada con el esquema de pesado *TF-IDF* [Alla, 00]. En el modelo de centroides, los vectores de las consultas son precisamente los centroides de los grupos.

El *centroide o representante de un grupo* es un documento, no necesariamente un documento real de la colección, que representa de alguna manera a todos los documentos del grupo. Existen varias formas

de calcular el centroide de un grupo, por ejemplo: la media, la mediana o la suma de todos los vectores de documentos del grupo. El centroide de un grupo, podría ser también el documento que es más similar al resto de los documentos del grupo.

En los sistemas que utilizan modelos de lenguaje para representar los documentos, la semejanza entre dos documentos puede calcularse de varias formas. La más sencilla es considerar uno de los documentos como una consulta (secuencia de términos) y el otro como modelo (unigramas), y aplicar las fórmulas (1) y (2) para obtener $P(d_i \mid d_j)$. Para lograr que la semejanza sea simétrica, habría que calcular también $P(d_j \mid d_i)$ y promediar ambas. En el caso de que se compare un documento con un centroide de un grupo, se toma el centroide como modelo y el documento como consulta.

Otra forma de calcular la semejanza entre dos documentos es medir la divergencia entre sus modelos. El método más utilizado es la divergencia de *Kullback-Leibler* (una variante de la entropía cruzada), que se define como sigue:

$$D(d_i \,\|\, d_j) = \sum_t P(t \mid d_i) \cdot \log \frac{P(t \mid d_i)}{P(t \mid d_j)} \qquad (5)$$

Se puede definir ahora una medida de semejanza simétrica, negando la suma de las divergencias entre los modelos:

$$sem(d_i, d_j) = -(D(d_i \| d_j) + D(d_j \| d_i))$$

2.3.5 Tratamiento de las propiedades temporales.

Un suceso es algo que ocurre en un instante de tiempo y lugar específicos. Las referencias temporales, junto con las palabras claves, nos permiten reconocer y diferenciar unos sucesos de otros (por ejemplo, ‘elecciones de 1999’, ‘elecciones de 2002’). Para localizar los tópicos o sucesos se requiere conocer cuándo se han producido. De ahí que el conocimiento de las propiedades temporales de las noticias juegue un papel muy importante en un sistema de detección de tópicos [Llid, 02].

En un sistema de detección, las propiedades temporales pueden considerarse de varias formas:

- Implícitamente mediante un orden cronológico del flujo de noticias.

- Definiendo una ventana temporal y comparando cada noticia sólo con las que pertenecen a dicha ventana. Esta ventana puede ser de tamaño fijo o definirse a partir de la fecha de publicación de las noticias.
- Incluyendo en la función de semejanza parámetros que tengan en cuenta estas propiedades temporales.

2.3.6 Algoritmo de agrupamiento

En los últimos 30 años, los análisis de agrupamientos (cluster) se han aplicado fuertemente a muchas áreas tales como la medicina (clasificación de enfermedades,), química (agrupamiento de compuestos), estudios sociales (clasificación de estadísticas), entre otros.
Su principal objetivo es identificar estructuras o subclases de los objetos en las bases de datos espaciales y que tengan algún sentido. La ventaja principal de usar esta técnica es que las estructuras o los agrupamientos interesantes se pueden encontrar directamente en los datos sin tener ningún conocimiento de fondo.
A pesar de que no existe una definición general de un agrupamiento, se han desarrollado algoritmos que encuentran diferentes clases de clústeres: esféricos, lineales, irregulares, entre otros. Motivados por su amplia gama de aplicaciones los investigadores han desarrollado técnicas para agrupar datos de diferentes tipos: binarios, nominales y otras clases de variables discretas, variables continuas, similaridades y disimilaridades. [Kaufman y Rousseeuw, 1990].

Los algoritmos de agrupamiento que se utilizan en un sistema de detección en línea tienen que cumplir los siguientes requisitos:

- Deben agrupar las noticias que abordan un mismo tópico en el mismo grupo.
- Deben ser algoritmos no supervisados, debido a que la tarea de detección está caracterizada por la carencia de conocimiento sobre el tópico que se desea detectar y, por tanto, no se dispone de muestra de entrenamiento etiquetada.

- Deben ser incrementales, pues las noticias son procesadas secuencialmente en la medida que se van publicando. Cada noticia debe ser colocada en un tópico existente o formar uno nuevo.
- No deben ser costosos computacionalmente, para poder ser capaces de agrupar miles de documentos en poco tiempo.

Los sistemas de detección actuales utilizan, por lo general como algoritmos de agrupamiento el *Single-Pass* [Hill, 68], el *K-Means incremental* [Lars, 99], el de los *K vecinos más cercanos* (*K-NN*) y variantes del *Scatter/Gather* [Cutt, 92], [Cutt, 93].

Divisiones de Algoritmos de agrupamiento.

Existen tres divisiones principales de algoritmos de agrupamiento: el agrupamiento particional, agrupamiento jerárquico y agrupamiento basado en localidades (figura 2.3). El agrupamiento particional (Partitional), desarrolla una partición de los datos tal que los objetos en un grupo son más similares a algunos que ellos a los objetos en otros grupos [Kennedy et al, 1998]. Este método construye k particiones de los datos, donde cada partición representa un grupo o cluster. Cada grupo tiene al menos un elemento y cada elemento pertenece a un solo grupo.
También, crea una partición inicial e iteran hasta un criterio de paro. Los más populares utilizan k-medias y k-medioides (por ejemplo, PAM, CLARA y CLARANS). El agrupamiento jerárquico (hierarchical), combina grupos pequeños en grupos grandes, o particiona los grupos grandes. En el agrupamiento basado en localidades (locality-based), los objetos del grupo se basan en las relaciones locales, y por consiguiente la base de datos puede examinarse en un paso.

Clasificación de los algoritmos de agrupamiento.

Se habla de algoritmos directos, constructivos o heurísticos cuando no optimizan ninguna función criterio. Si usan una función criterio a optimizar se habla de algoritmos indirectos o por optimización [Ester et al, 1996].
Por otro lado, según la filosofía empleada para la construcción de agrupamientos se distinguen: algoritmos aglomerativos o incrementales (bottom-up) los cuales parten de patrones aislados y tienden a unir agrupamientos de acuerdo a algún umbral fijado (por ejemplo, AGNES). Los algoritmos divisivos o decrementales (top-down) se generan a partir de agrupamientos ya establecidos y tienden a crear nuevos agrupamientos más homogéneos. (Por ejemplo, DIANA). Por su parte los algoritmos mixtos como su nombre indica, incorporan procesos de creación y mezcla de nuevos agrupamientos.

Existen otros métodos de agrupamiento; Los métodos basados en densidades en los que se agrupan objetos mientras su densidad (número de objetos) en la "vecindad" este dentro de un cierto umbral (por ejemplo, DBSCAN, DENCLUE). Lo métodos basados en celdas en los cuales se divide el espacio en celdas a diferentes niveles (por ejemplo, STING, CLIQUE); y los métodos basados en modelos donde se debe encontrar un modelo para cada cluster que mejor ajuste los datos de ese grupo (por ejemplo, COBWEB, AutoClass).

Técnicas de MDE basadas en Agrupamiento.

El clustering o agrupamiento es el proceso de particionar un conjunto de datos (u objetos) en un conjunto de subclases significativas llamadas grupos (clusters). Un grupo es una colección de objetos de datos que son similares a otros y así pueden ser tratados colectivamente como un grupo.
El agrupamiento es una forma de clasificación no supervisada en la que, a diferencia de la supervisada, no se conocen las etiquetas de las clases (no hay clases predefinidas) y puede que tampoco se conozca el número de grupos.
Un buen método de agrupamiento produce grupos de alta calidad en los cuales la similaridad intra-clases (esto es, dentro del grupo) es alta y la similaridad inter-clase (entre las clases) es baja. La medida de similaridad se define usualmente por proximidad en un espacio multidimensional.

2.4 Principales aproximaciones en la detección de tópicos

En este epígrafe se analizarán las principales aproximaciones existentes en la detección de tópicos. De cada una de ellas se estudiarán sus características principales, hipótesis de partida y limitaciones.

Los sistemas de detección de tópicos reportados en la literatura podríamos clasificarlos en:

- Los que usan como modelo de representación de documentos el vectorial. Ejemplos de ellos son: CMU, UMASS, el sistema de Papka, UPENN, IBM, Iowa, el sistema de Kurt y el de Brants.
- Los que usan modelos probabilísticos. Ejemplos de ellos son: Dragón, BBN y TNO.

A continuación se presentarán las características principales de cada uno de ellos.

2.4.1 El sistema CMU

El sistema CMU, desarrollado por la Universidad Carnegie Mellon [Yang, 98], [Yang, 99], [Carb, 99] se basa en las siguientes ideas:

- Las noticias que discuten el mismo tópico tienden a ser temporalmente próximas, lo que sugiere el uso de una medida que combine la semejanza léxica con la proximidad temporal para el agrupamiento de documentos.
- Un intervalo de tiempo entre noticias que abordan un mismo tópico indica la existencia de tópicos diferentes; por ejemplo, diferentes accidentes de aviones, diferentes terremotos, etc.
- El cambio significativo del vocabulario y los rápidos cambios de la distribución de las frecuencias de los términos son típicos de las noticias que abordan un nuevo tópico, lo que indica la importancia de la actualización dinámica del vocabulario del corpus y los pesos de los términos.

Para representar los documentos CMU utiliza el modelo vectorial con el método de asignación de pesos a los términos *ltc*. Los vectores de documentos se truncan a sólo los *k* términos más pesados, ignorando el resto de los términos (el valor de *k* es empíricamente seleccionado) [Yang, 99].

Como función de semejanza entre los documentos se utiliza la medida del coseno y como medida de semejanza entre grupos se usa el coseno entre sus correspondientes prototipos o centroides. El centroide de un grupo se define como la suma normalizada de los vectores de los documentos de dicho grupo.

CMU realiza los dos tipos de detección de tópicos: retrospectiva y en línea. Para la detección retrospectiva se emplea el *Scatter/Gather* [Cutt, 92], [Cutt, 93] con el método *Fractionation* para la selección de los centroides y la estrategia de agrupamiento *Group-average* [Murt, 83], [Will, 88]. Para la detección en línea, se usa el algoritmo *Single-Pass* [Hill, 68].

Detección retrospectiva [Yang, 98], [Yang, 99]

Para dividir la colección en partes al aplicar el algoritmo *Scatter-Gather*, se tiene en cuenta el orden cronológico de los documentos. Esto no se hace por problemas de eficiencia sino para explotar la propiedad de proximidad temporal de las noticias que abordan un mismo tópico. El algoritmo consiste en los siguientes pasos:

1. Ordenar las noticias en orden cronológico y usar esto como la partición inicial del corpus. Cada grupo es unitario.

2. Dividir la partición actual en partes consecutivas y no solapadas de tamaño fijo.
3. Aplicar el algoritmo jerárquico aglomerativo a cada parte hasta que el número de grupos en cada parte sea reducido por un factor de reducción ρ.
4. Eliminar las fronteras entre las partes preservando el orden de los grupos formados. Considerar esto como la nueva partición del corpus.
5. Repetir los pasos 2, 3 y 4 hasta que se alcance un número prefijado de grupos en la partición actual del corpus.
6. Periódicamente (una vez cada *k* iteraciones del paso 5) se reagrupan las noticias dentro de cada grupo de la partición actual usando el algoritmo jerárquico, provocando, por tanto, que crezca el número de grupos.

El paso 6 es una modificación que hace el sistema CMU al algoritmo *Scatter/Gather*. Este paso es útil, pues los subconjuntos de noticias que abordan el mismo tópico dentro de partes diferentes son generalmente agrupadas junto con noticias similares en un nivel más bajo y sólo después, en un nivel más alto del árbol de grupos se unen. Este reagrupamiento reduce el sesgo que provoca la división en partes y contribuye, por tanto, a la formación de mejores grupos.

Como puede verse, este algoritmo trabaja con varios parámetros: el número de grupos por partes (400), el factor de reducción ρ (0.5), el umbral de semejanza mínima para que dos grupos se unan (0.2), el número de términos de los prototipos de los grupos (100) y el número de iteraciones entre los reagrupamientos (5). Los números entre paréntesis indican los valores utilizados en los experimentos realizados sobre la colección TDT-A.

El algoritmo *Scatter/Gather* tiene una complejidad cuadrática. Por utilizar la estrategia *Group-average* tiende a crear grupos esféricos y de tamaños iguales, lo que trae como consecuencia que cuando se aplica a una colección de documentos que no sigue esta distribución el algoritmo pierde eficacia. Para realizar la división de la colección en partes en la aplicación de *Fractionation*, utiliza la proximidad temporal de las noticias basada en su ordenamiento cronológico.

Detección en línea [Yang, 98], [Yang, 99], [Carb, 99]:

Para la representación de los documentos en su sistema de detección en línea, CMU combina las dos aproximaciones para la actualización dinámica de los pesos de los términos, es decir, comienza utilizando

los pesos *IDF* obtenidos a partir de un corpus retrospectivo y, luego, los actualiza cada vez que procesa un nuevo documento. En este caso, el peso *IDF* incremental se define como:

$$IDF(t,p) = \log_2 \frac{N(p)}{n(t,p)},$$

donde *p* es el tiempo actual, *N*(*p*) es el número de documentos procesados hasta el momento actual (incluyendo los documentos del corpus retrospectivo) y *n*(*t*,*p*) es el número de documentos hasta el momento actual que contienen al término *t*.

Para la detección en línea, se define una ventana temporal que contiene las *m* noticias previas. Para cada documento procesado se calcula su valor de semejanza con los centroides de los grupos que tienen al menos un documento en su ventana temporal.

Teniendo en cuenta todo lo anterior, el algoritmo de detección de CMU basado en el *Single-Pass* trabaja de la siguiente forma [Yang, 99]:

1. Se fijan los parámetros: tamaño de la ventana temporal (*m*) y el umbral de novedad (t_n).
2. Sea *d* el nuevo documento. Actualizar los pesos *IDF*.
3. Se calcula la semejanza entre *d* y cada centroide $\vec{c}$ de los grupos existentes:

$$sem(d,\vec{c}) = \begin{cases} \left(1 - \frac{i}{m}\right) \cdot \cos(d,\vec{c}) & si\ c\ tiene\ algún\ documento\ en\ la\ ventana \\ 0 & en\ otro\ caso \end{cases}$$

   donde *i* es el número de documentos entre el documento *d* y el documento más reciente del grupo *c* en la ventana.

4. Sea *maxsem* la mayor semejanza obtenida en el paso anterior. Para controlar las decisiones en la detección se utiliza el umbral de novedad t_n. Si *maxsem* > t_n entonces se añade el documento *d* al grupo más similar en la ventana y se actualiza su centroide. Si no, se crea un nuevo grupo con el documento *d*.
5. Mover la ventana temporal hacia delante e ir al paso 2.

En los experimentos realizados sobre la colección TDT-A en la detección en línea se utilizaron un tamaño de ventana de 250 documentos y un umbral de novedad de 0.16.

Nótese que las propiedades temporales de las noticias son incorporadas a la decisión en la detección limitando las comparaciones entre los documentos a aquellos que se encuentran en una ventana temporal de tamaño fijo y definiendo una función de semejanza que toma en consideración la posición de los documentos en esta ventana temporal.

2.4.2 El sistema UMASS

El sistema de detección UMASS [Alla, 00] fue desarrollado en la Universidad de Massachussets y se basa en el algoritmo de agrupamiento del vecino más cercano (*1-NN*). Para representar los documentos utilizan el modelo vectorial y el esquema de pesado *TF-IDF* con los *IDF* estimados a partir de un corpus retrospectivo. Para realizar las comparaciones emplean la tradicional medida del coseno.

El algoritmo de detección trabaja como sigue: cada vez que se procesa un nuevo documento, se compara con todos los documentos de los grupos existentes. Se selecciona el documento más similar y si esta semejanza es mayor que un umbral especificado se declara que el nuevo documento aborda el tópico representado por el grupo al que pertenece el documento más similar. Por el contrario, si la semejanza máxima no supera este umbral el documento forma un nuevo grupo y, por tanto, aborda un nuevo tópico.

2.4.3 El sistema de Papka

Ron Papka [Papk, 98], [Papk,99], también de la Universidad de Massachusetts, propone un sistema de detección que está basado en el algoritmo *Single-Pass*. Para la representación de los documentos utiliza el modelo vectorial y una variante del esquema de pesado *Okapi tf* con los *IDF* estimados a partir de un corpus auxiliar y a partir de ellos formula un conjunto de clasificadores. En esta aproximación se introduce un modelo de umbral que toma en consideración la adyacencia temporal de las noticias.

La formulación del clasificador tiene tres etapas:

1. Selección de los rasgos.

   Para la selección de los rasgos del clasificador se toman los *n* términos más frecuentes en el documento (del cual se formula el clasificador) exceptuando los pertenecientes a la lista de parada. Aquí *n* es la dimensionalidad del clasificador y constituye un parámetro del sistema.

2. Asignación de los pesos.

Para el cálculo de los pesos de los términos se usa una variante del esquema de pesado *Okapi tf.* El clasificador q_i es un vector de pesos correspondiente a cada uno de los rasgos seleccionados en la etapa anterior, donde el peso del término *t* en el instante de tiempo *i* se calcula como:

$$w(t, q_i) = TF_{comp}(t, d_i)$$

donde d_i es el documento del cual se formula el clasificador en el instante *i*.

El documento d_j que llega en el instante de tiempo *j* se representa como un vector, donde el peso de cada término t_k, que aparece también en el clasificador, se calcula como sigue:

$$w(t_k, d_j) = 0.4 + 0.6 \cdot TF_{comp}(t_k, d_j) \cdot IDF_{comp}(t_k)$$

Para el cálculo de TF_{comp} e IDF_{comp} se utilizan las fórmulas (3) y (4). Sin embargo, dado que en la detección en línea $df(t_k)$ es desconocido, éste se estima usando un corpus auxiliar de un dominio similar. Si el término no aparece en el corpus auxiliar se asume $df(t_k) = 1$.

3. Estimación del umbral.

   Como medida de semejanza entre el clasificador q_i y el documento d_j se utiliza la suma pesada.

   Se asume que el documento d_j aborda el tópico representado por el clasificador q_i si su semejanza supera un cierto umbral. Si el clasificador es creado en el instante de tiempo *i*, esto es, cuando el último documento relevante de entrenamiento llega, entonces el umbral del clasificador para el documento que llega en el instante de tiempo *j* es:

   $$umbral(q_i, d_j) = 0.4 + \theta \cdot (s_{opt} - 0.4)$$

   donde s_{opt} es el valor de semejanza para el clasificador que cuando se aplica a los documentos de entrenamiento etiquetados con el tópico optimiza la función de coste de detección usada en TDT. El parámetro θ controla el modelo del umbral. Por ejemplo, si $\theta = 1$ el umbral es s_{opt}. El umbral del clasificador se estima encontrando un valor que separa los documentos relevantes de los no relevantes mientras optimiza la medida de efectividad usada para la evaluación.

Para el problema de la detección de nuevos tópicos, se utiliza un algoritmo similar al *Single-Pass* utilizando la representación de los textos explicada anteriormente. Este algoritmo procesa secuencialmente cada nuevo documento en el flujo de información de la siguiente forma [Papk, 99]:

1. Formular la representación del clasificador para el documento.
2. El umbral inicial del clasificador es la semejanza entre el clasificador y el documento del cual se formuló.
3. Re-estimar el umbral cuando cada nuevo documento llega. Aquí se usa un modelo de umbral que incorpora la componente del tiempo, es decir, incorpora las relaciones temporales entre las noticias. Se explota el hecho de que los documentos cercanos en el flujo de noticias son más propensos a abordar el mismo tópico que aquellos que están más lejanos. Cuando ocurre un nuevo tópico, existen usualmente varios documentos por día que lo abordan. En la medida que el tiempo pasa, el cubrimiento de los viejos tópicos es desplazado por los nuevos. Por tanto, dado el clasificador formulado en el instante de tiempo *i*, su umbral para el documento que llega en el instante de tiempo *j* es:

   $$umbral\,(q_i, d_j) = 0.4 + \theta \cdot (sem(q_i, d_i) - 0.4) + \beta \cdot (j - i)$$

   donde $sem(q_i, d_i)$ es la semejanza entre el clasificador y el documento del cual se formuló, el valor de $(j - i)$ es el número de días entre el documento d_j y la formulación del clasificador q_i. Los valores θ y β controlan la decisión en la clasificación de nuevos tópicos.
4. Comparar el nuevo documento con los clasificadores existentes en memoria.
5. Si ninguna semejanza entre el documento y los clasificadores existentes superan el umbral se etiqueta al documento como que contiene un nuevo tópico. En caso contrario, no se etiqueta al documento. Aquí se asume que si un clasificador tiene una decisión positiva, el documento aborda el tópico representado por el clasificador.
6. (Opcional) Añadir el documento a la lista de documentos de los clasificadores que tuvieron una decisión positiva.
7. (Opcional) Reformular los clasificadores existentes utilizando sus listas actualizadas de documentos. El peso del rasgo t_k es el promedio de las componentes *TF* en ese rasgo en sus documentos.
8. Añadir el nuevo clasificador a la memoria.

En los experimentos utilizados en el corpus de evaluación TDT-A se obtuvieron los mejores resultados para $\theta = 0.225$, $\beta = 0.000008$ y $n = 400$. Los resultados de este sistema se describirán en el epígrafe 2.5 junto a los de los otros sistemas analizados.

2.4.4 El sistema UPENN

Este sistema de detección [Schu, 99] fue desarrollado en la Universidad de Pennsylvania. Utiliza una representación vectorial de los documentos del tipo *TF-IDF* y como función de semejanza la medida del coseno. A diferencia de otros sistemas, no usa los lemas (*stems*) de las palabras ni ningún esquema de normalización.

Su sistema de detección está basado en el algoritmo jerárquico aglomerativo con la estrategia *Single-link* [Murt, 83], [Will, 88] y utiliza como parámetro el período de aplazamiento (*deferral period*), definido como el número de ficheros (cada uno contiene múltiples noticias) que el sistema puede procesar antes de asignarle un identificador de tópico a las noticias contenidas en los ficheros.

El algoritmo funciona como sigue. Inicialmente cada noticia forma un grupo unitario. Dos grupos se unen si la semejanza entre una noticia de un grupo y otra del otro grupo supera un umbral (determinado a partir de un conjunto de prueba). Luego, cada noticia se compara con todas las noticias precedentes (incluyendo las de los períodos de aplazamiento anteriores). Si la semejanza entre dos noticias supera el umbral, entonces sus grupos se unen. Por supuesto, los grupos de períodos de aplazamiento anteriores no pueden unirse, pues el identificador del grupo para las noticias de esos períodos ya ha sido detectado.

La complejidad del algoritmo es cuadrática.

2.4.5 El sistema IBM

Este sistema [Dhar, 99] utiliza una representación de los documentos basada en el modelo vectorial, usando los pesos *TF-IDF*. Realizan varios pasos de pre-procesamiento a los documentos que incluyen: etiquetado morfológico (*part-of-speech tagging*), extracción de lemas (*stemming*), extracción de rasgos de los unigramas y bigramas de sustantivos adyacentes, etc. Para comparar a los documentos con los

centroides de los grupos utilizan la medida de semejanza empleada en el sistema de recuperación *Okapi* y no utiliza período de aplazamiento.

Los grupos *cl* se representan a través de sus centroides $\bar{c}$ y éstos se definen como la media de la frecuencia de los términos en el grupo, es decir:

$$TF(t,\bar{c}) = \frac{1}{|cl|}\sum_{d\in cl} TF(t,d)$$

El algoritmo de detección es un algoritmo de agrupamiento incremental que utiliza como medida de semejanza entre un documento *d* y el centroide $\bar{c}$ de un grupo *cl* una forma simétrica de la fórmula de *Okapi*, la cual se define de la forma siguiente [Dhar, 99]:

$$Ok(d,\bar{c}) = \sum_{t\in d\cap\bar{c}} TF(t,d)\cdot TF(t,\bar{c})\cdot IDF(t,cl)$$

donde las frecuencias del término *t* en los documentos han sido normalizadas por la longitud de los documentos y distorsionadas para evitar el sobre-pesado de los términos repetidos. Para permitir que los pesos de los términos varíen en la medida en que los grupos evolucionan el *IDF* se define como:

$$IDF(t,cl) = IDF_0(t) + \Delta IDF(t,cl)$$

donde $IDF_0(t)$ es el *IDF* estándar (independiente de los grupos) y $\Delta IDF(t,cl)$ es una medida de la semejanza entre dos conjuntos de documentos: D_t, el conjunto de los documentos que contienen al término *t* en toda la colección y el conjunto de documentos del grupo *cl*. Este término se define así:

$$\Delta IDF(t,cl) = \lambda\frac{2\cdot n_{t,cl}}{|D_t|\cdot|cl|}$$

donde $n_{t,cl}$ es el número de documentos en $D_t \cap cl$. Este factor puede ser interpretado como una media armónica entre la relevancia y la precisión (si D_t se interpreta como el conjunto de documentos relevantes y *cl* como el conjunto de documentos recuperados). Note que $\Delta IDF(t,cl) = 0$ si y sólo si $D_t \cap cl$ es vacío y $\Delta IDF(t,cl) = \lambda$ si y sólo si D_t=*cl*.

El algoritmo de agrupamiento procede como sigue: cada documento *d* se compara con todos los centroides de los grupos existentes. Sea *cl** el grupo que maximiza $Ok(d,\bar{cl}*)$. Si $Ok(d,\bar{cl}*) > \Theta_m$ se

incorpora *d* a *cl**. Si $\Theta_c \leq Ok(d, \bar{cl}*) \leq \Theta_m$ se etiqueta con el tópico correspondiente pero sin incorporarlo al grupo. Por último, si $Ok(d, \bar{cl}*) < \Theta_c$ y *d* contiene más de 20 términos diferentes se crea un nuevo grupo con el documento *d* (denominado semilla). La causa de esta última condición es que los documentos pequeños son menos estables como semillas.

A pesar de que el sistema permite trabajar con parámetros Θ_c y Θ_m diferentes, en los experimentos realizados por los autores no encontraron ninguna ventaja en esto, excepto en el caso de los grupos unitarios en que toman $\Theta_m > \Theta_c$ para hacer más difícil la unión de un documento a un grupo unitario.

2.4.6 El sistema Iowa

El sistema de la Universidad de Iowa [Eich, 99] representa a los documentos utilizando el modelo vectorial y los pesos *TF-IDF*. Al conjunto de términos se le extrae los lemas mediante el algoritmo de Porter [Port80] y se filtran mediante listas de parada. Los pesos de los términos son actualizados incrementalmente cada 10 ficheros de entrada. Los vectores de documentos se podan a los 100 términos más pesados mientras que los vectores de los grupos sólo contienen los 200 términos más pesados. Utilizan la medida de semejanza del coseno.

El algoritmo de detección usa un modelo de tubería (*pipeline*). Las noticias en cada fichero de entrada se agrupan usando un umbral de pertenencia (α). Cada grupo del conjunto así obtenido (teniendo en cuenta el período de aplazamiento especificado) es, entonces, comparado con los grupos previamente identificados por el sistema. Si esta semejanza supera el umbral de semejanza inter-grupos (α también), ambos grupos se unen. Si no, el grupo se compara con los grupos posteriores para comprobar si algún grupo futuro es suficientemente cercano con él. Los grupos unitarios que no cumplen ambas condiciones se consideran ruido. Los grupos no unitarios son considerados como un nuevo tópico.

En los experimentos realizados en la colección TDT-B se obtuvo el C_{DET} más bajo (0.0078) para α=0.15 y un período de aplazamiento de 100. Este sistema obtiene demasiados grupos (de 200 a 300 tópicos).

2.4.7 El sistema de Kurt

El sistema de Kurt [Kurt, 01] combina en un mismo sistema la detección y el seguimiento de tópicos. Su objetivo es detectar automáticamente nuevos tópicos a partir de múltiples fuentes e inmediatamente seguir la evolución de ellos. En este epígrafe se abordará lo relacionado con la detección de tópicos.

Inicialmente las noticias son procesadas para la extracción de los términos. Esta fase incluye la eliminación de las palabras vacías, eliminación de etiquetas (de tiempo, de información de recursos, etc.), la extracción de los lemas y correcciones simples. Para representar a los documentos se utiliza el clásico modelo vectorial y como esquema de pesado de los términos se usa el *TFC* (*term frequency component*), con el *IDF* incremental, esto es:

$$w(t,d) = \frac{TF(t,d)\cdot \log\left(\frac{N_p}{df_p(t)}\right)}{\sqrt{\sum_{j=1}^{n}\left[TF(t_j,d)\cdot \log\left(\frac{N_p}{df_p(t_j)}\right)\right]^2}}$$

donde p es el instante de tiempo en el que llega el documento d, N_p es el número de documentos acumulados en el instante p, y $df_p(t)$ es el número de documentos que contienen al término t en el instante p.

Para comparar a los documentos utilizan la medida del coseno penalizada por el tiempo. Como la suma de los cuadrados de los pesos *TFC* es 1, la medida del coseno es equivalente al producto escalar.

Se define, entonces, una ventana temporal basada en días y suavizada exponencialmente. Así, la función de semejanza temporal utilizada es:

$$sem(d_i,d_j) = \begin{cases} \cos(d_i,d_j) & si\ (fecha_i - fecha_j) < 1 \\ (fecha_i - fecha_j)^{-\alpha}\cdot \cos(d_i,d_j) & si\ 1 \le (fecha_i - fecha_j) \le m \\ 0 & si\ (fecha_i - fecha_j) > m \end{cases}$$

Para evitar las constantes alarmas de nuevos tópicos que ocurren en un sistema de detección, introducen un umbral de soporte (definido por el usuario), que no es más que el número de noticias necesarias de cada fuente antes de dar la alarma.

El algoritmo de detección está basado en el algoritmo de agrupamiento de los *K* vecinos más cercanos (*K-NN*), el cual consiste en lo siguiente:

1. Entrada del nuevo documento *d*.
2. Eliminar aquellas noticias cuya diferencia en días con el día del nuevo documento excedan el umbral de la ventana temporal *m*.
3. Calcular la semejanza del coseno entre el nuevo documento y los documentos de la ventana temporal. Seleccionar los *K* más semejantes (*DK*).
4. Aplicar la función semejanza temporal a los *K* documentos más semejantes.
5. Si la máxima semejanza no supera el umbral de novedad, etiquetar al documento como un nuevo tópico. Calcular el valor de soporte para el nuevo tópico.
6. Si la máxima semejanza supera el umbral de novedad entonces:
   a. Encontrar todos los tópicos que contienen al menos uno de sus vecinos más semejantes.
   b. Para cada tópico encontrado:
      - Calcular:

$$sem(d, DK) = \frac{1}{|P_{DK}|} \sum_{y \in P_{DK}} \cos(d, y) - \frac{1}{|Q_{DK}|} \sum_{z \in Q_{DK}} \cos(d, z)$$

      donde P_{DK} (Q_{DK}) es el conjunto de instancias positivas (negativas) entre los *K* vecinos más cercanos de *d*.
      - Si *sem*(*d*, *DK*) > *Umbral* añadir el documento *d* al tópico y recalcular los valores de soporte.
7. Si el valor de soporte de algún tópico excede el umbral de soporte, emitir la alarma de nuevo tópico.
8. Ajustar los contadores para el cálculo del *IDF* del próximo documento.
9. Ir al paso 1.

En los experimentos realizados con un corpus de 465 noticias (de ellas sólo 322 etiquetadas en 15 tópicos) provenientes de Reuters, utilizaron como umbral de la ventana temporal 15, 10, 7 y 4 días y como α = 0.25. Los valores obtenidos de la medida F1 fueron bajos (del orden de 0.2) y costes de detección altos (del orden de 0.16 sin normalizar).

La principal aportación de este algoritmo es que permite que un documento esté etiquetado por más de un tópico. Además tiene en cuenta de forma explícita las propiedades temporales de los documentos para la detección de los tópicos.

2.4.8 El sistema de Brants

El sistema de detección de tópicos de Thorsten Brants [Bran, 03] está basado en un modelo *TF-IDF* incremental con algunas extensiones que incluyen la generación de modelos específicos según la fuente, la normalización de las semejanzas basadas en los promedios, el pesado de los términos basado en las frecuencias inversas de los tópicos y la segmentación de los documentos.

Las noticias son preprocesadas, reconociendo las abreviaturas, reemplazando los numerales por sus dígitos, extrayendo los lemas, eliminando las palabras vacías y normalizando las abreviaturas.

Utilizan como modelo de representación de los documentos el vectorial y los pesos *TF-IDF*, los cuales se actualizan de forma incremental. Para reflejar las diferencias existentes en el vocabulario de las noticias provenientes de diversas fuentes construyen un modelo específico por fuente e incorporan, además, las frecuencias de los tópicos según las reglas de interpretación (ROI) que contiene la colección TDT. Una regla de interpretación puede verse como una categorización de alto nivel de los tópicos. Por ejemplo, una regla de interpretación de la colección TDT es *Elecciones*.

Su esquema de pesado de los términos *t* en el documento *d* proveniente de la fuente *s* en el instante de tiempo *p* es el siguiente:

$$w_p(t,d) = \frac{1}{Z_p(d)} TF(t,d) \cdot \log \frac{N_p}{df_{s,p}(t)} \cdot g\left(\log \frac{N_{e,t}}{ef(t)} \right)$$

Aquí, N_p es el número total de documentos en el instante de tiempo *p*, $Z_p(d)$ es una constante de normalización, $N_{e,t}$ es el número de tópicos en la regla de interpretación *r* que maximiza la ecuación

$ef(t) = \max_{r \in ROI} ef(r,t)$, siendo *ef*(*r*, *t*) el número de tópicos que contienen al término *t* y pertenecen a la regla *r*. Finalmente, *g* es una función de escala lineal:

$$g(x) = (x - A)\frac{D - C}{B - A} + C$$

con $A = \min_t \log(\frac{N_{e,t}}{ef(t)})$, $B = \max_t \log(\frac{N_{e,t}}{ef(t)})$, C = 0.8 y D = 1.

Esto provoca que los términos no específicos de un tópico se multipliquen a lo sumo por un factor de 0.8 mientras que los específicos reciban el peso *TF-IDF* original.

En el instante de tiempo *p* un nuevo conjunto de documentos C_p es adicionado al modelo y se actualizan, por tanto, las frecuencias $df_{s,p}(t)$ de la siguiente forma:

$$df_{s,p}(t) = df_{s,p-1}(t) + df_{C_p}(t)$$

donde $df_{C_p}(t)$ es el número de documentos en C_p provenientes de *s* que contienen al término *t*. Las frecuencias iniciales $df_{s,0}(t)$ se generan a partir de un conjunto de entrenamiento. Si no está disponible dicho conjunto para una fuente en particular, se calculan a partir de los conjuntos de entrenamiento de fuentes similares. Como los términos con baja frecuencia son poco informativos, sólo se usan los términos con $df_{s,p}(t) > 2$.

Una alta semejanza de un documento de un tópico amplio a otro documento no significa lo mismo que una alta semejanza de un documento de un tópico específico a otro documento. Para capturar esta diferencia se calcula el promedio de las semejanzas del nuevo documento *d* con todos los documentos existentes, denotado con $\overline{sem(d)}$.

Por otra parte, si *a*, *a'* y *b* son documentos que abordan el mismo tópico, pero *a* y *a´* provienen de una fuente *A* y *b* de otra *B*, entonces el valor esperado de la semejanza *E*[*sem*(*a*,*a'*)] > *E*[*sem*(*a*,*b*)]. Para reflejar estas diferencias se calcula el promedio de las semejanzas de las noticias de un mismo tópico, denotado $E_{s(d),s(d_i)}$, provenientes del par de fuentes de *d* (*s*(*d*)) y de d_i (*s*(d_i)).

Cada documento se divide en segmentos solapados por una ventana de tamaño fijo *l* (en palabras) y que se desplaza con paso fijo. Para comparar a dos documentos *d* y d_i, se calcula la semejanza de cada segmento de *d* con cada segmento de d_i.

De esta forma, la semejanza entre el nuevo documento *d* y uno existente d_i se calcula como sigue:

$$sem(d,d_i) = \max_{seg_k \in d, seg_q \in d_i} \left\{ sem_p(seg_k, seg_q) - \overline{sem(d)} - E_{s(d),s(d_i)} \right\}$$

donde:

$$sem_p(seg_k, seg_q) = \sum_t \sqrt{w_p(t, seg_k) \cdot w_p(t, seg_q)}$$

es la medida de Hellinger entre los segmentos de los documentos *d* y d_i.

Para detectar los nuevos tópicos se compara al nuevo documento *d* con todos los documentos previos y se calcula $score(d) = 1 - sem(d, d^*)$, donde *d** es el documento con la máxima semejanza. Si esta puntuación supera el umbral θ_s entonces *d* aborda un nuevo tópico y si no, *d* aborda un tópico existente.

Este sistema incorpora algunos aspectos novedosos como el pesado de los términos teniendo en cuenta la fuente y la regla de interpretación, así como medidas de semejanza entre segmentos de los documentos.

En los experimentos realizados con la colección TDT-C usando como colección de entrenamiento a la TDT-B se obtuvieron costes de detección normalizados de 0.5783. Con la colección TDT-C usando como entrenamiento las colecciones TDT-C y TDT-B se obtuvieron costes de 0.5691. Se realizaron experimentos en las colecciones TDT-C y TDT-D con un modelo del tiempo pero no se obtuvieron mejoras en la eficacia del sistema.

La principal aportación de este sistema es que reemplaza la tradicional medida de semejanza del coseno por la distancia de Hellinger, además de introducir elementos nuevos como los modelos *TF-IDF* específicos según la fuente de las noticias, la normalización de las semejanzas y la comparación parcial de los documentos a través de sus segmentos.

2.4.9 El sistema Dragón

El sistema Dragón [Lowe, 99], [Yamr, 00] usa aproximaciones estadísticas basadas en el modelo Beta-binomial [Gill, 90] para la detección de tópicos, donde se calcula la distribución de probabilidades de que las palabras (unigramas) aparezcan un número de veces en un documento de una longitud dada. Cada

vez que un nuevo documento es procesado, los parámetros para la distribución de cada palabra son re-estimados.

Su algoritmo de detección se basa en el algoritmo de agrupamiento *K-MEANS*. La decisión de crear un nuevo grupo es equivalente a declarar la aparición de un nuevo tópico. En este proceso de decisión se considera que un documento aborda un nuevo tópico cuando está más próximo al modelo discriminador que a un grupo de noticias existentes. Más específicamente, el sistema trabaja de la siguiente forma [Yamr, 00]:

- En cada momento existen *k* grupos de noticias, cada uno caracterizado por un conjunto de estadísticos. Para cada nueva noticia se calcula la distancia al grupo más cercano y si es menor que un umbral prefijado, se inserta en él y se actualizan los estadísticos. Si esa distancia supera el umbral anterior, se crea un nuevo grupo.
- Se itera a través de todas las noticias, asignándolas nuevamente al grupo más cercano. En este proceso algunos grupos pueden cambiar e incluso pueden aparecer nuevos. Este paso se repite un número de veces prefijado.

Como puede notarse este algoritmo requiere de una medida de distancia probabilística entre una noticia y un grupo, la cual se describe como sigue:

- Cada grupo es tratado como un "tópico", para el cual se construye un modelo de lenguaje basado en unigramas.
- A partir de una colección previa (*background material*) se construye un modelo de unigramas discriminador, denotado *U.*
- La distancia entre una noticia *d* y un grupo existente *cl* es el logaritmo del cociente de la probabilidad de que el modelo del grupo genere la noticia y la probabilidad de que la noticia sea generada por el discriminador, es decir,

$$dist(d, cl) = \sum_t P(t \mid d) \cdot \log \frac{P(t \mid U)}{P(t \mid cl)} + \varepsilon$$

Note que esta distancia está basada en la divergencia entre modelos (fórmula (5)). El término ε es un parámetro que mide la diferencia entre el índice de la noticia *d* y el índice medio entre la primera y la última noticia del grupo *cl*. Este parámetro fue introducido para provocar que los grupos tengan

una duración limitada en el tiempo [Alla, 98]. El sistema de detección necesita ajustar el término ε y el umbral usando corpus de prueba.

Una limitación de este sistema es su eficiencia. El algoritmo de detección requiere reiteradas estimaciones de los parámetros estadísticos cada vez que una noticia es añadida o eliminada de un grupo [Lowe, 99].

2.4.10 El sistema BBN

Este sistema de detección [Wall, 99], [Leek, 00] representa a los documentos mediante un modelo del lenguaje y usa una variante incremental del algoritmo *K-MEANS*, donde no se necesita de antemano fijar el número de grupos *k*.

Para comparar a los documentos utiliza dos tipos de métricas: de selección y de umbral. Como métrica de selección utiliza una medida de semejanza probabilística llamada *BBN topic spotting metric* y como métrica de umbral utiliza un híbrido entre la medida probabilística anterior y la tradicional métrica del coseno.

Dada una noticia *d*, una métrica de selección encuentra su grupo más similar. La métrica de selección utilizada por BBN se define como la probabilidad de generar el documento a partir del modelo de lenguaje asociado al grupo *cl, es* decir:

$$D(d, cl) = P(d \mid cl) = \prod_k P(t_k \mid cl)$$

El cálculo de la probabilidad de cada término $P(t_k \mid cl)$ se realiza con un método de suavizado de interpolación lineal (ver fórmula (2)).

Por otra parte, el objetivo de una métrica de umbral es determinar si una noticia debe incorporarse a un grupo. En BBN utilizan como métrica de umbral una combinación entre la medida del coseno y la medida probabilística anterior. Como la medida probabilística no está normalizada utilizan el método de normalización siguiente:

$$D(d, cl) = \frac{\log P(d \mid cl) - \mu}{\sigma}$$

donde μ es un estimado de la media del logaritmo de las probabilidades de la noticia para el grupo *cl* y σ un estimado de su desviación estándar.

La métrica de umbral utilizada decide que una noticia se incorpore a un grupo dado si la medida del coseno o la métrica BBN normalizada anterior es menor que un cierto umbral.

El algoritmo de agrupamiento que utiliza en el proceso de detección consiste en los pasos siguientes [Wall, 99]:

1. Aplicar un algoritmo de agrupamiento incremental a los documentos de la ventana actual, es decir, comparar cada noticia de la ventana actual con los grupos existentes, seleccionar el grupo más similar con ésta; si supera el umbral se une al grupo y si no, forma un nuevo grupo.
2. Modificar todos los grupos de acuerdo con las nuevas asignaciones.
3. Repetir los pasos 1 y 2 hasta que el agrupamiento no cambie.
4. Tomar las próximas noticias e ir al paso 1.

En este algoritmo el parámetro *k* del algoritmo *K-MEANS* es libre y permite reestructurar los grupos iniciales pobremente formados.

Este sistema usa una variante incremental del algoritmo *K-MEANS* que puede verse como un híbrido entre los algoritmos *Single-Pass* y *K-MEANS*. Este algoritmo no tiene la limitación de la selección inicial de las semillas y no exige la formación de *k* grupos como lo hace el tradicional *K-MEANS*. Además esta métrica requiere de una colección adicional para estimar las probabilidades de las palabras en el modelo general de frecuencias de los términos en inglés.

2.4.11 El sistema TNO

El sistema de detección de tópicos TNO [Spitt, 01] también está basado en un modelo del lenguaje (unigramas). Para el agrupamiento de las noticias combinan el algoritmo *Single-Pass* para establecer los grupos iniciales con un método de relocalización para estabilizar los grupos dentro de un período de aplazamiento especificado.

La semejanza entre una noticia *d* y un grupo existente se calcula como el promedio de las semejanzas entre la noticia y cada noticia d_i del grupo. La semejanza entre dos noticias se define como la suma de $P(d_i \mid d)$ y $P(d \mid d_i)$.

El algoritmo de agrupamiento realiza los siguientes pasos:

1. Para cada noticia de la ventana de aplazamiento, se calcula su semejanza con cada grupo existente. Si la semejanza al grupo más cercano es mayor que un cierto umbral, se añade la noticia a este grupo, si no se crea uno nuevo y ella es su semilla.
2. Cuando se alcanza la última noticia de la ventana, se procesan de nuevo todas ellas comparándose con cada grupo existente. Aquí puede ocurrir lo siguiente:
    - Una noticia puede cambiar a otro grupo si su semejanza con él supera tanto a la semejanza con su grupo actual como al umbral.
    - Si ni la semejanza de la noticia con su grupo actual ni su semejanza con el resto de los grupos supera el umbral, se crea un nuevo grupo con esta noticia como semilla.
    - En el caso de que la semejanza al grupo al que pertenece sea mayor que el umbral y también mayor que todas las semejanzas con los demás grupos, la noticia se queda en el mismo grupo.

Un grupo que ha permanecido inalterable por un período ininterrumpido de 15 días se convierte en un tópico inactivo. Con ello se limita la complejidad computacional, pues las nuevas noticias no tienen que compararse con dichos grupos.

El sistema se evaluó sobre el corpus correspondiente a las noticias del mes de abril de la colección TDT-B. En esta subcolección se obtuvo un coste de detección normalizado entre 0.15 y 0.27. A pesar de que el método de relocalización trata de atenuar la dependencia del orden del algoritmo *Single-Pass*, éste está restringido a la ventana de aplazamiento.

2.5 Algoritmos de Clasificación.

La clasificación automática de documentos puede concebirse como un proceso de "aprendizaje matemático-estadístico", durante el cual un algoritmo implementado computacionalmente capta las características que distinguen cada categoría o clase de documentos de las demás, es decir, aquellas que deben poseer los documentos para pertenecer a esa categoría. Estas características no tienen porqué indicar de forma absoluta e inequívoca la pertenencia a una clase o categoría, sino que más bien lo hacen en función de una escala o graduación. De esta forma, por ejemplo, documentos que posean una cierta característica tendrán un factor de posibilidades de pertenecer a determinada clase, de modo que la acumulación de dichas características arrojará un resultado que consiste en un coeficiente asociado a

cada una de las clases ya conocidas. Este coeficiente lo que expresa en realidad es el grado de confianza o certeza de que el documento en cuestión pertenezca a la clase asociada al coeficiente resultante.

Clasificación automática de texto.

La clasificación de texto (también conocida como categorización de texto o ubicación del tema) es la tarea de clasificar automáticamente un conjunto de documentos en categorías (o temas) dentro de un conjunto predefinido. Actualmente la exactitud de muchos de los sistemas de clasificación de texto compite con la de profesionales humanos especializados. Tal avance se debe principalmente a la combinación de tecnologías como son la recuperación de información y el aprendizaje automático, lo cual desde principios de los 90's ha ganado popularidad y eventualmente se ha convertido en el enfoque dominante para construir sistemas de clasificación de textos. La idea básica de este enfoque es que un proceso inductivo automáticamente construya un clasificador por observar las características de un conjunto de documentos previamente clasificados.
En la figura 2.4 se muestra una visión general de los elementos y procesos necesarios dentro de la clasificación automática de textos.

2.5.1 Técnicas de clustering

Existen varios métodos o algoritmos que utiliza Clustering para formar los grupos o clases en la data. Dentro de estos métodos se encuentran el K-Means, los métodos jerárquicos, algoritmos divisivos, clustering probabilístico, algoritmos basados en densidad, el método E-M (Expectation - Maximization), entre otros.
Estos métodos se pueden clasificar según los siguientes criterios:

- Clustering por partición.

  El método de partición separa n objetos en k grupos llamados particiones, donde cada partición representa un clúster. En este método se asume que:
  - $k \leq n$, o sea existen igual o menor cantidad de grupos que de objetos.
  - Cada partición debe contener al menos un objeto.

- Cada objeto debe ser miembro de una o más particiones (está permitido que un objeto pertenezca a más de una partición con diferentes grados de pertenencia o probabilidades).

En estos métodos se suele crear una partición inicial, que posteriormente es mejorada usando técnicas iterativas. En cada iteración se mueven los objetos de una partición a otra. Un ejemplo de técnica de este tipo es el algoritmo de las K-medias [Theodoridis y Koutroumbas, 1999], o el método E-M (Expectation - Maximization) [Lauritzen, 1995].

- Clustering jerárquico.

  Los métodos jerárquicos de clustering realizan una descomposición jerárquica del conjunto de datos, usando técnicas aglomerativas o divisivas.

- Otros métodos.

  Otros métodos son por ejemplo los basados en cuantización, que discretizan el espacio donde se quiere hacer el clustering en un número finito de categorías formando una estructura de cuadrícula, y realizando el clustering sobre esa estructura.

Existen otro métodos como los basados en modelos, que realizan un modelado de cada clúster y encuentran el mejor ajuste del modelo o los basados en redes neuronales, básicamente Mapas Autoorganizativos [Kohonen, 1997]. Además, existen algoritmos de clustering que se basan en combinar diferentes técnicas para mejorar el resultado final.

2.5.2 Métodos utilizados para realizar clustering

K-medias

El algoritmo de K-medias es un método de clustering por partición. Este método es uno de los más utilizados en aplicaciones científicas e industriales. Como su nombre lo indica representa cada uno de los clústeres por la media de sus puntos, es decir, por su centroide. Este método únicamente se puede aplicar

a atributos numéricos, sin embargo la representación mediante centroides tiene la ventaja de que tiene un significado gráfico y estadístico inmediato.

La función objetivo que utiliza este algoritmo es determinada a partir de la suma de las diferencias entre un punto y su centroide, expresado a través de la distancia entre ellos. La función objetivo que viene expresada por la suma de los cuadrados de los errores entre los puntos y sus centroides respectivos, es igual a la varianza total dentro del propio clúster.

Existen dos versiones del método de las k-medias.

Se basa en dos pasos iterativos: en el primer paso se reasignan todos los puntos a sus centroides más cercanos, y en el segundo paso se re calculan los centroides de los nuevos grupos. El proceso continúa hasta alcanzar un criterio de parada (por ejemplo que no se realicen nuevas reasignaciones).

La segunda versión reasigna los puntos basándose en un análisis más detallado de los efectos causados sobre la función objetivo al mover un punto del clúster donde se encuentra a otro nuevo. Si la reasignación es positiva se realiza, en caso contrario se queda como está.

Expectation – Maximization (EM)

El algoritmo de Expectación Máxima pertenece a los algoritmos de clustering por partición.
El funcionamiento de este método se basa en tratar de obtener la Función de Densidad de Probabilidad (FDP) desconocida a la que pertenecen el conjunto completo de datos. Esta función se puede aproximar mediante una combinación lineal de **NC** componentes, definidas a falta de una serie de parámetros

$\{\Theta\} = \cup\{\Theta_j \forall j = 1..NC\}$, que son los que hay que averiguar,

$$P(x) = \sum_{j=1}^{NC} \pi_j p(x; \Theta_j) \text{ con } \sum_{j=1}^{NC} \pi_j = 1 \quad (1)$$

donde π_j son las probabilidades a priori de cada clúster cuya suma debe ser 1, que también forman parte de la solución buscada, P(x) denota la FDP arbitraria $p(x; \Theta_j)$ la función de densidad del componente j.

Cada clúster se corresponde con las respectivas muestras de datos que pertenecen a cada una de las densidades que se mezclan. Se pueden estimar funciones de formas arbitrarias. El ajuste de los parámetros del modelo requiere alguna medida de su bondad, es decir, cómo encajan los datos sobre la distribución que los representa. Este valor de bondad se conoce como el likelihood de los datos. Se trataría entonces de estimar los parámetros buscados £, maximizando este likelihood (este criterio se conoce como ML-Maximun Likelihood).

Normalmente, lo que se calcula es el logaritmo de este likelihood, conocido como log-likelihood ya que es más fácil de calcular de forma analítica. La solución obtenida es la misma, gracias a la propiedad de monotonicidad del logaritmo. La forma de esta función log-likelihood es:

$$L(\Theta, \pi) = log \prod_{n=1}^{NI} P(x_n)$$

donde NI es el número de instancias, que suponemos independientes entre si.

2.6 Valoración final.

La detección y seguimiento de tópicos (TDT) es una línea de investigación en la que se agrupan diferentes aproximaciones, las cuales tratan de resolver los problemas propios que ella plantea. Está claro que según sea el método de representación de los documentos, la medida de semejanza empleada y el algoritmo de agrupamiento utilizado así será la eficiencia lograda en la detección de tópicos.

Los diferentes sistemas de detección desarrollados: CMU, UMASS, Papka, UPENN, IBM, Iowa, los sistemas de Kurt y de Brants, Dragón, BBN y TNO solucionan en gran parte los problemas y tareas

presentes en esta línea, pero dejan un espacio y a la vez dan una esperanza de mejora partiendo de sus resultados.

La tabla 2.2 resume las principales características de cada uno de los sistemas estudiados en la detección en línea.

Sistema	**Modelo de representación**	**Medida de semejanza**	**Algoritmo de agrupamiento**	**Propiedades temporales**
CMU	Vectorial con pesos *ltc*	Coseno ponderada por la posición relativa de los documentos en la ventana temporal	*Single-Pass*	Ventana temporal y parámetros en la medida de semejanza
UMASS	Vectorial con pesos *TF-IDF*, *IDF* estima-dos a partir de corpus retrospectivo	Coseno	*1-NN*	Orden cronológico
Papka	Vectorial con variante del esquema *Okapi tf*	Suma pesada	*Single-Pass*	Modelo de umbral
UPENN	Vectorial con pesos *TF-IDF*	Coseno	*Single-Link*	No
IBM	Vectorial con varian-te de *TF-IDF*	Medida de *Okapi*	*Single-Pass*	Orden cronológico
Iowa	Vectorial con	Coseno	Modelo de	Orden

	pesos *TF-IDF*		tubería	cronológico
Kurt	Vectorial con pesos *tfc*	Coseno penalizada por el tiempo	*K-NN*	Ventana temporal y parámetros en la medida de semejanza
Brants	Vectorial con variante del *TF-IDF* incremental	Variante de la medida de Hellinger	*1-NN*	No
Dragón	Modelo del lenguaje	Distancia probabilística	*K-Means*	Parámetros en la medida de distancia
BBN	Modelo del lenguaje	BBN y métrica de selección	*K-Means Incremental*	No
TNO	Modelo del lenguaje	Semejanza probabilística	Variante del *Single-Pass*	No

Tabla 2.2.- Características generales de los sistemas de detección en línea.

Las aproximaciones a la detección de tópicos estudiadas, en general, presentan las siguientes limitaciones:

1. Dependencia de los tópicos detectados al orden de presentación de las noticias. Esta dependencia trae como consecuencia que los tópicos detectados pueden ser diferentes. A pesar de que se establece un ordenamiento cronológico de las noticias durante su procesamiento, la presencia de diversas fuentes obliga a establecer un orden determinado. Los tópicos detectados dependerán de este orden. Esta limitación está presente en los sistemas CMU, UMASS, Papka, IBM, los sistemas de Kurt y Brants, Dragón y TNO.
2. Realizan una asignación irrevocable de los documentos a los grupos, es decir, esta asignación se realiza tan pronto llega un nuevo documento y, luego, no cambia más, por lo que errores cometidos

cuando se disponía de poca información no se recuperan y pueden mermar notablemente la eficacia del sistema. Está presente en CMU, UMASS, Papka, IBM y los sistemas de Kurt y Brants.

3. No tienen en cuenta las propiedades temporales de los documentos en el cálculo de la semejanza entre ellos. Algunos sólo las emplean de forma implícita en el ordenamiento cronológico de las noticias. Ejemplos de sistemas con esta limitación son: UMASS, UPENN, IBM, Iowa, el sistema de Brants, BBN y TNO.
4. Las propiedades temporales de los documentos que manipulan sólo están restringidas a su posición en la ventana temporal, la cual es de tamaño fijo. Está presente en CMU.
5. Presentan el llamado efecto de encadenamiento, es decir, dos noticias diferentes en cuanto a su contenido pueden unirse en un mismo grupo, lo que provoca que los grupos formados tengan muy poca cohesión interna. Esta limitación está presente en UPENN.
6. Todos los sistemas estudiados obtienen un conjunto de tópicos y no tienen en cuenta diferentes niveles de granularidad en ellos. Los tópicos en la vida real no están bien definidos, unos pueden ser más amplios que otros, por lo que sería interesante poder captar estos niveles de detalles.

Es por ello, que es necesario continuar desarrollando sistemas que superen las limitaciones señaladas.

2.6.1 Análisis valorativo de los diferentes algoritmos de agrupamiento y clasificación de textos.

Las colecciones de prueba (Corpus de textos).

Una de las principales contribuciones a las investigaciones de la Detección y Seguimiento de Tópicos es la creación de un corpus que comprende una colección de noticias de diversas fuentes. El corpus TDT-A contiene 863 documentos cronológicamente ordenados, estructurados en formato SGML y obtenidos de una muestra aleatoria de artículos de CNN y de Reuters en el período comprendido del 1ero de julio de 1994 al 30 de junio de 1995, etiquetados en 25 tópicos. Cada noticia del corpus está etiquetada con uno de tres valores posibles: Sí (la noticia aborda el tópico), No (la noticia no aborda el tópico) y Breve (la noticia menciona brevemente al tópico, es decir, menos del 10% de la noticia lo aborda).

TDT-B, coleccionó durante 6 semanas (de enero a febrero de 1998) las noticias provenientes de 6 fuentes: dos agencias de noticias (*New York Times News Service* y *Associated Press Worldstream News Service*), dos programas radiales (*PRI The World* y *VOA English News Programs*) y dos programas de televisión (*CNN Headline News y ABC World News Tonight*). TDT-B contiene aproximadamente 400 noticias y 70 tópicos.

TDT-C, es una colección más grande que contiene noticias de los meses de octubre a diciembre de 1998 provenientes de fuentes en español con 60 tópicos etiquetados manualmente. Cada tópico etiqueta 35 documentos como promedio.

El corpus TDT-D comprende aproximadamente 380 noticias en español publicadas en los meses de octubre de 2000 a enero de 2001. Está etiquetado en 50 tópicos.

2.6.2 Resultados de los algoritmos de agrupamiento.

Evaluación de los sistemas de detección en las colecciones de TDT

En este epígrafe se mostrarán los resultados obtenidos por los diferentes métodos y algoritmos en el corpus TDT-A., TDT-B.

En la tabla 2.3 aparecen los resultados de CMU, UMASS, Dragón y el sistema de Papka en el corpus de evaluación TDT-A. Como puede observarse, los valores de la medida F1 obtenidos son bajos.

Método(Algoritmo)	P_m **(%)**	P_{f_a} **(%)**	***F1***

CMU(*Single-Pass*)	59	1.43	0.40
UMASS(*1-NN*)	51	1.21	0.35
Dragón(*K-Means*)	58	3.47	0.28
Papka(*Single-Pass*)	**54**	**1.13**	**0.46**

Tabla 2.3.- Resultados obtenidos en el corpus de evaluación TDT-A.

En la evaluación se presentaron los sistemas de detección: BBN, CMU, Dragón, IBM, Iowa, UMASS y UPENN. Los resultados obtenidos en el corpus TDT-B se muestran en la tabla 2.4. Como puede observarse, IBM logró el coste de detección macro-promediado más bajo: 0.0042 correspondiente a un 20% de omisiones y un 0.07% de falsas alarmas.

Método(Algoritmo)	***Story-weighted***				***Topic-weighted***			
	P_m	P_{f_a}	C_{DET}	C_{DET} ***Norm.***[1]	P_m	P_{f_a}	C_{DET}	C_{DET} ***Norm.***
BBN(*K-Means Incremental*)	0.0941	0.0021	0.004	0.2	0.1295	0.0021	0.0047	0.235
UMASS(*1-NN*)	0.0913	0.0022	0.004	0.2	0.2091	0.0023	0.0064	0.32
Dragón(*K-Means*)	0.1638	0.0013	0.0045	0.225	0.1787	0.0013	0.0048	0.24
IBM(*Single-Pass)*	**0.1965**	**0.0007**	**0.0046**	**0.23**	**0.1766**	**0.0007**	**0.0042**	**0.21**
UPENN(*Single-Link)*	0.2997	0.0011	0.0070	0.35	0.2617	0.0011	0.0063	0.315
Papka(*Single-Pass*)	0.65	0.0172	0.0297	1.485	-	-	-	-
CMU(*Single-Pass*)	0.3526	0.0004	0.0075	0.375	0.2644	0.0004	0.0057	0.285
Iowa*(Modelo de tubería)*	0.6028	0.0009	0.0129	0.645	0.4311	0.0009	0.0095	0.475

Tabla 2.4.- Resultados obtenidos en el corpus de evaluación TDT-B.

[1] En la evaluación no se usaron los costes de detección normalizados.

Análisis de los resultados obtenidos en los corpus de evaluaciones TDT – A y TDT-B.

Se puede concluir que de los algoritmos evaluados el más eficiente es Single-Pass porque en los resultados obtenidos en el corpus de evaluación TDT-A obtuvo una medida de F1 de 0.46 es decir que por cada 100 se obtuvieron 46 tópicos positivos. En los resultados obtenidos en el corpus de evaluación TDT-B se obtuvo un coste de detección de macro-promediado 0.0042 lo que implica que se obtuvieron los valores más bajos de probabilidad de omisión y falsas alarmas.

2.6.3 Resultados de los algoritmos de clasificación.

En este epígrafe se mostrarán los resultados obtenidos por los diferentes algoritmos en el corpus TDT-C., TDT-D.

En la tabla 2.5 aparecen los resultados de *K-Means y Expectativa Máxima* en el corpus de evaluación TDT-C. Como puede observarse, los valores de la medida F1 obtenidos son altos.

Algoritmos de Clasificación	P_m **(%)**	P_{f_a} **(%)**	***F1***
K-Means	57	1.53	0.74
Expectativa Máxima	**52**	**1.60**	**0.88**

Tabla 2.5.- Resultados obtenidos en el corpus de evaluación TDT-C.

En la tabla 2.6 aparecen los resultados de *K-Means y Expectativa Máxima* en el corpus de evaluación TDT-D. Como puede observarse, *Expectativa Máxima* logró el coste de detección macro-promediado más bajo: 0.0074 correspondiente a un 11% de omisiones y un 0.01% de falsas alarmas.

Algoritmo	***Story-weighted***				***Topic-weighted***			
	P_m	P_{f_a}	C_{DET}	C_{DET} ***Norm.***[1]	P_m	P_{f_a}	C_{DET}	C_{DET} ***Norm.***
K-Means	0.0642	0.0724	0.1725	0.1	0.3255	0.0124	0.0159	0.246
Expectativa Máxima	**0.0811**	**0.0021**	**0.003**	**0.1**	**0.1081**	**0.0045**	**0.0074**	**0.21**

Tabla 2.6.- Resultados obtenidos en el corpus de evaluación TDT-D.

Análisis de los resultados obtenidos en los corpus de evaluaciones TDT- C y TDT- D.

Se puede concluir que de los algoritmos evaluados el más eficiente es Expectativa Máxima porque en los resultados obtenidos en el corpus de evaluación TDT-C obtuvo una medida de F1 de 0.88 es decir que por cada 100 se obtuvieron 88 tópicos positivos. En los resultados obtenidos en el corpus de evaluación TDT-D se obtuvo un coste de detección de macro-promediado 0.0074 lo que implica que se obtuvieron los valores más bajos de probabilidad de omisión y falsas alarmas.

[1] En la evaluación no se usaron los costes de detección normalizados.

CONCLUSIONES

- Se realizó un análisis de diferentes algoritmos de clasificación y agrupamiento empleados en la minería de textos, a partir de un corpus de texto, preparado a tal efecto.
- Para el análisis se empleo el sistema WEKA, implementado específicamente para minería de texto.
- El sistema WEKA, es libre y esta validado para este tipo de trabajos, por experimentos realizados anteriormente.
- Los resultados de este análisis permitirán demostrar la valides de estos algoritmos para problemas de minería de texto.

RECOMENDACIONES

- Continuar investigando en esta temática, con vistas a determinar otros elementos de estos algoritmos.
- Implementar mejoras en estos algoritmos para hacerlos más eficientes.
- Emplear la herramienta WEKA, en otros trabajos de este tipo.

REFERENCIAS BIBLIOGRÁFICAS

[AllA, 98], Allan, J.; Carbonell, J.; Doddington, G.; Yamron, J. and Yang, Y. "Topic Detection and Tracking Pilot Study: Final Report". Proceedings of DARPA Broadcast News Transcription and Understanding Workshop, pp. 194-218, 1998.

[Alla, 00] Allan, J.; Lavrenko, V. and Jin, H. "First Story Detection in TDT is hard". Proceedings of the Ninth International Conference on Information and Knowledge Management, pp. 374-381, 2000.

[Alla, 00a] Allan, J.; Lavrenko, V. ; Frey, D. and Khandelwal, V. "UMASS at TDT 2000". Proceedings TDT 2000 Workshop, 2000.

[Alla, 00b] Allan, J.; Lavrenko, V. and Jin, H. "Comparing Effectiveness in TDT and IR". CIIR Technical Report, 2000.

[Bellman, 1978], *An introduction to Artificial Intelligence: can computer think?.* San Francisco, California: Boyd & Fraser Publishing Company.

[Berry, 2000] Berry, Linoff: Mastering Data Mining, Wiley, 2000.

[Bran, 03] Brants, T.; Chen, F. and Farahat, A. "A System for New Event Detection". Annual ACM Conference on Research and Development in Information Retrieval. Proceedings of the 26th annual international ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR'03, Toronto, Canada, 2003.

[Call, 96] Callan, J. P. "Document Filtering with Inference Networks". Proceedings of ACM SIGIR, pp. 262-269, 1996.

[Carb, 99] Carbonell, J.; Yang, Y.; Lafferty, J.; Brown, R.D.; Pierce, T. and Liu, X. "CMU Report on TDT-2: Segmentation, detection and tracking". Proceedings of DARPA Broadcast News Workshop, pp. 117-120, 1999.

[Cutt, 92] Cutting, D. R.; Karger, D. R.; Pedersen, J. O. and Tukey, J. W. "Scatter/Gather: a Cluster-based approach to browsing large document collections". Proceedings of the 15th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 318-329, 1992.

[Cutt, 93] Cutting, D. R.; Karger, D. R. and Pedersen, J. O. "Constant Interaction-Time Scatter/Gather Browsing of Very Large Document Collections". Proceedings of the 16th International ACM SIGIR Conference on Research and Development in Information Retrieval, 1993.

[Crivat, 2005] Crivat,B.:SQL Server Data Mining Programmability.URL: http://msdn.microsoft.com/sql/bi/dmining/default.aspx?pull=/library/en-us/dnsql90/html/sqldmprgrm.asp. Fecha de Acceso: Dic 12, 2006.

[Eich, 99] Eichmann, D.; Ruiz, M.; Srinivasan, P.; Street, N.; Culy, C. and Menczer, F. "A Cluster-Based Approach to Tracking, Detection and Segmentation of Broadcast News". Proceedings of the DARPA Broadcast News Workshop, San Francisco, Morgan Kaufmann Publishers, pp. 69-76, 1999.

[Charniak, E. y McDermott, D., 1985]. Introduction to Artificial Intelligence. Addison-Wesley, Reading, Massachusetts.

[Gelbukh y Bolshakov, 1999], *Avances en Análisis Automático de Textos*. Proc. Foro: Computación, de la Teoría a la Práctica. IPN, México City, May 26 – 28, 1999.

[Gree, 00] Greengrass, Ed. "Information Retrieval: a Survey". November, 2000.

[Hearst, 1999], Untangling Text Data Mining, Proc. of ACL'99: The 37th Annual Meeting of the *Association for Computational Linguistics*, University of Maryland, June 20-26, 1999.

[Hill, 68] Hill, D. R. "A vector clustering technique". In Samuelson (ed.), Mechanized Information Storage, Retrieval and Dissemination, North-Holland, Amsterdam, 1968.

[Kaufman, L., y Rousseeuw, 1990] Finding groups in data: An introduction to cluster analysis. New York: John Wiley & Sons, Inc.

[Kodratoff, 1999], Knowledge Discovery in Texts: *A Definition and Applications*, Proc. of the 11th International Symposium on Foundations of Intelligent Systems (ISMIS-99), 1999.

[Kohonen, 1997]. Self-organizing maps (Second ed.).

[Kurzweil, 1990] Kurzweil, Ray. Artificial Intelligence: The age of intelligent machines, 1990.

[Kurt, 01] Kurt, H. "On-line New Event Detection and Tracking in a Multi-Resource Environment". Thesis for degree in Master in Science. Bilkent University, September, 2001.

[Lauritzen, 1995]. The EM algorithm for graphical association models with missing data. Computational Statistics and Data Analysis. 1995.

[Lars, 99] Larsen, B. and Aone, C. “Fast and Effective Text Mining Using Linear-time Document Clustering”. In KDD’99, San Diego, California, pp. 16-22, 1999.

[Leek, 00] Leek, T.; Jin, H.; Sista,S. and Schwartz, R. “The BBN Crosslingual Topic Detection and Tracking System”. Working Notes of the Third Topic Detection and Tracking Workshop, Feb. 2000.

[Llid, 02] Llidó Escrivá, D. M. “Extracción y Recuperación de Información Temporal”. Tesis doctoral, Universitat Jaume I, 2002.

[Lowe, 99] Lowe, S.A. “The Beta-Binomial Mixture Model and its application to TDT Tracking and Detection”. Proceedings of the 1999 DARPA Broadcast News Workshop, pp. 127-131, 1999.

[Luger y Stubblefield, 1993] Luger, G. F., y W. A. Stubblefield, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Benjamin/Cummings, 1993.

[Iyer, 05] Iyer, Raman and Crivat, Bogdan SQL Server Data Mining: Plug-In Algorithms. . Fecha de Acceso: Dic 13, 2006 URL: http://msdn.microsoft.com/sql/bi/dmining/default.aspx?pull=/library/en-us/dnsql90/html/ssdmpia.asp.

[MacLennan, 2004] MacLennan, J.: Unearth the New Data Mining Features of Analysis Services 2005.; development lead for the Data Mining engine in the SQL Server 2005. MSDN Magazine, September 2004. URL: http://msdn.microsoft.com/msdnmag/issues/04/09/AnalysisServices2005/.

[Murt ,83] Murtagh, F. “A survey of recent advances in hierarchical clustering algorithms”. Computer Journal, 26:354-359, 1983.

[Netz, 2005] Netz, A.; SQL Server 2000: Data Mining Helps Customers Make Better Business Decisions. Interviewed Netz, Amir; Microsoft SQL Server Development Manager. URL: http://www.microsoft.com/presspass/features/2000/04-24sql.mspx.

[Papk, 98] Papka, R. and Allan, J. “On line new event detection using Single Pass Clustering”. UMASS Computer Science Technical Report, pp. 98-21, 1998.

[Papk, 99] Papka, R. “On-line New Event Detection, Clustering and Tracking”. Ph.D. thesis, University of Massachusetts, Department of Computer Science, September 1999.

[Pons, 02] Pons-Porrata, A.; Berlanga-Llavori, R. and Ruiz-Shulcloper, J. “Temporal-Semantic Clustering of Newspaper Articles for Event Detection”. Pattern Recognition in Information Systems, Eds. José M. Iñesta y Luisa Micó, ICEIS Press, April 2002, pp. 104-113, 2002.

[Pons, 03] Pons-Porrata, A.; Berlanga-Llavori, R. and Ruiz-Shulcloper, J. “Building a hierarchy of events and topics for newspaper digital libraries”. Lectures Notes on Computer Sciences 2633, Springer-Verlag, Berlin Heidelberg, F. Sebastiani (Ed.), 2003.

[Quinlan, 1986] J. Ross Quinlan. Induction of decision trees. Machine Learning, 1986.

[Quinlan, 1993] J. Ross Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA, 1993.

[Ragh, 86] Raghavan V. and Wong, S.K.M. “A critical analysis of Vector Space Model for Information Retrieval”. Journal of the American Society on Information Science, vol. 37, No. 5, pp. 279-287, 1986.

[Rich, Knight, 1991] Rich, Elaine and Kevin Knight. Artificial Intelligence. New York: McGraw-Hill, 1991.

[Rich, 1994] Rich, E., Inteligencia Artificial, McGraw-Hill, 1994.

[Rijs, 79] Van Rijsbergen, C. J. “Information Retrieval”. Butterworths, London, second edition, 1979.

[Robe,95] Robertson, S. E.; Walker, W.; Jones, S.; Hancock-Beaulieu, M. and Gartford, M. “Okapi at TREC-3”. Proceedings of TREC-3, pp. 109-126, 1995.

[Rosete, 2004] Minería de Datos: El camino de la academia a la realidad cotidiana. Alejandro Rosete Suárez. CEIS, CUJAE. Ciudad de la Habana, Cuba.

[Salt, 89] Salton, G. “Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer”, Addison-Wesley, 1989.

[Schu, 99] Schultz J. M. and Liberman M. “Topic Detection and Tracking using idf-weighted cosine coefficient” Proceedings of the DARPA Broadcast News Workshop, pp. 189-192, 1999.

[Schalkoff, 1990] Schalkoff, R., Artificial Intelligence, McGraw-Hill, 1990.

[Shapiro, 1990] Shapiro, S. C.: Encyclopedia of Artificial Intelligence, Wiley Interscience-John Wiley & Sons, 1990.

[Spitt, 01] Spitters, M. and Kraaij, W. “TNO at TDT2001: Language Model-Based Topic Detection”. Topic

Detection and Tracking (TDT) Workshop, Gaithersburg, USA, November 2001.

[Tan, 1999], Text Mining: *The state of the art and challenges*, Proc. of the Workshop Knowledge Discovery from advanced Databases PAKDDD-99, Abril 1999.

[TangM, 2005]. Tang, Z., MacLennan J.: Data Mining with SQL Server, ISBN-10: 0-471-46261-6.

[TDT2, 98] National Institute of Standards and Technology. "The Topic Detection and Tracking Phase 2 (TDT2)", evaluation plan, version 3.7, 1998.

[TDT, 03] National Institute of Standards and Technology. "The 2003 Topic Detection and Tracking (TDT2003). Task definition and Evaluation Plan", version 1.0, 2003.

[Turing, 36], "On Computable Numbers, with an Application to the Entscheidungsproblem", Proceedings of the London Mathematical Society, vol. 42 (1936); "Correction", ibid. vol. 43 (1937).

[Turing, 50], "Máquinas computadores e inteligencia", en ANDERSON 74.

[Wal, l99] Walls, F.; Jin, H.; Sista, S. and Schwartz, R. "Topic Detection in Broadcast news". Proceedings of the DARPA Broadcast News Workshop, pp. 193-198, 1999.

[Winston, 1992], Artificial Intelligence. Addison-Wesley, 3rd edition.

[Witten and Frank, 2000] Ian H. Witten and Eibe Frank. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, 2000.

[Yamr, 00] Yamron, J. "Dragon´s Tracking and Detection Systems for TDT2000 Evaluation". Proceedings of Topic Detection and Tracking Workshop, pp. 75-80, 2000.

[Yang, 97] Yang, Y. "An evaluation of statistical approaches to text categorization", Technical report, Carnegie Mellon University, 1997.

[Yang, 98] Yang, Y; Pierce, T. and Carbonell, J. "A Study on Retrospective and On-Line Event Detection". Proceedings of the 21th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'98, pp. 28-36, 1998.

[Yang, 99] Yang, Y.; Carbonell, J.; Brown R.; Pierce, T.; Archibald B.T. and Liu X. “Learning approaches for Detecting and Tracking New Events”. IEEE Intelligent Systems 14(4):32-43, July/August 1999. Special Issue on Applications of Intelligent Information Retrieval.

[Yang, 00] Yang, Y.; Ault, T.; Pierce, T. and Lattimer, C. W. “Improving text categorization methods for event tracking”. Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'00, Athens, pp. 65-72, 2000.

BIBLIOGRAFÍA

Garay, E. Miguel. Inteligencia Artificial 2005 .Libro en preparación. CEIS-CUJAE, 2005.

Moreno Espino, Mailyn. *Metodologías Orientadas a Agentes: un estudio comparativo.* Tesis doctoral. Instituto Superior Politécnico “José Antonio Echevarría”, 2006.

Montes, A. Gelbukh. *Un método de agrupamiento de grafos conceptuales para minería de texto [pdf].* Centro de Investigación en Computación (CIC), IPN, México.

Montes Gómez, Manuel. *Minería de texto: Un nuevo reto computacional [pdf*]. Centro de Investigación en Computación, Instituto Politécnico Nacional, México.

1