UNIVERSIDAD JAUME I

Escuela Superior de Tecnología y Ciencias Experimentales

Departamento de Lenguajes y Sistemas Informáticos

TESIS DOCTORAL

DESARROLLO DE ALGORITMOS PARA LA ESTRUCTURACIÓN DINÁMICA DE INFORMACIÓN Y SU APLICACIÓN A LA DETECCIÓN DE SUCESOS

Presentada por:

Aurora Pons Porrata

Dirigida por:

Rafael Berlanga Llavori y José Ruiz Shulcloper

Castellón, Junio 2004

AGRADECIMIENTOS

Quiero dar mi más sincero agradecimiento a todas aquellas personas e instituciones que han posibilitado la realización de esta tesis:

A mis directores de tesis:

Rafael Berlanga y José Ruiz Shulcloper

Ambos con inagotable paciencia y dedicación me han proporcionado ideas y consejos para llevar adelante este trabajo. Ellos, más que como tutores, se han comportado como verdaderos amigos. En particular, quiero agradecer a José Ruiz Shulcloper por haberme introducido en el área del Reconocimiento de Patrones y haberme ayudado a dar mis primeros pasos en el fascinante mundo de la investigación. A Rafael Berlanga le agradezco, además, la confianza que depositó en mí, al aceptarme como estudiante prácticamente sin conocerme.

A *Dolores Llidó* y *Juan Manuel Pérez*, con quienes compartí despacho y me dieron todo su apoyo y cooperación.

Agradezco, además, a *María José*, *Badía*, *Quirós*, *Pablo*, *Teresa*, *Filiberto*, *Amable* y *Pedro* por haberme hecho sentir en este país como en mi propia tierra.

A mis amigas y amigos, a todos los profesores de mi departamento y a los miembros del grupo de Bases de Conocimiento Temporal por su aliento y preocupación. Especialmente a aquellos que me dieron su apoyo incondicional en los momentos difíciles.

En el plano personal, quiero hacer un agradecimiento con todo mi amor a *Gil*, mi esposo, por ser mi sostén, mi aliento y el aire que respiro. A mis dos hijos, *Reynaldito* y *Aurorita*, por ser la fuente que me inspira para ser cada día mejor. Y a mis padres y mi familia por todo lo que me han dado y ayudado en el aspecto personal y profesional.

Quiero darle las gracias, además, a aquellas instituciones que financiaron o contribuyeron con mi trabajo: al *Departamento de Computación de la Universidad de Oriente*, al *Departamento de Lenguajes y Sistemas Informáticos* y al *Rincón de la Solidaridad, estos últimos de la Universidad Jaume I*.

A todos, muchas gracias.

Castellón, a 30 de Marzo de 2004.

RESUMEN

La motivación fundamental de esta tesis es el problema de la detección de tópicos, cuyo objetivo es identificar en un flujo continuo de noticias, aquéllas que pertenecen a un nuevo tópico o a uno no conocido previamente. Los intentos de enfrentar este problema dieron lugar al desarrollo de un conjunto de algoritmos y técnicas para la estructuración dinámica de la información y su descripción.

En este trabajo se presentan y evalúan diferentes aproximaciones al problema del agrupamiento de colecciones dinámicas de objetos. En este sentido, se proponen soluciones a los tres problemas de clasificación que pudieran presentarse en la práctica: la obtención de particiones o grupos disjuntos, la obtención de cubrimientos o grupos solapados y la construcción de jerarquías de grupos. Así, se proponen tres nuevos algoritmos de agrupamiento: *Compacto Incremental*, *Fuertemente Compacto Incremental* y *Jerárquico Incremental*. Estos algoritmos son eficientes, permiten la manipulación de objetos mezclados incluso con ausencia de información, no imponen restricciones a la función de semejanza ni a la forma de los grupos, son independientes del orden de presentación de los objetos y no realizan una asignación irrevocable de los objetos a los grupos. A pesar de que los utilizamos para agrupar colecciones de documentos, su uso no está restringido a esta área, pues pueden utilizarse en cualquier problema del Reconocimiento de Patrones donde se necesite agrupar objetos de cualquier naturaleza.

La necesidad de proporcionar mecanismos eficientes para describir los tópicos detectados por un sistema de detección dio lugar al desarrollo de un método de construcción automática de resúmenes a partir de una partición de conjuntos de documentos. Nuestro método emplea el cálculo de los testores típicos como su operación primaria y, a partir de ellos, construye el resumen de cada grupo. A diferencia de los otros métodos propuestos en la literatura, la selección de las oraciones se basa en los términos frecuentes y discriminantes de cada grupo, los cuales pueden ser calculados con el algoritmo *LEX* aquí presentado. Este algoritmo demostró, a su vez, ser más eficiente que los algoritmos para el cálculo de los testores típicos existentes en la literatura.

Los algoritmos mencionados anteriormente fueron utilizados en el problema de la detección de tópicos. En el trabajo presentamos un sistema de detección de tópicos en línea, denominado *JERARTOP*, que va más allá de un sistema tradicional de detección, ya que extrae o infiere de forma completamente automática e incremental el conocimiento implícito en el flujo de documentos. Este conocimiento se expresa en forma de una jerarquía de tópicos/subtópicos, es decir, el sistema permite identificar no sólo los tópicos presentes en un flujo de noticias sino también los posibles sucesos más pequeños que ellos comprenden.

A diferencia de los sistemas de detección propuestos en la literatura, esta aproximación evita la definición de qué es un tópico, ya que es el usuario el que navegando por la jerarquía creada decide en qué nivel explora el conocimiento de su interés.

En nuestro sistema de detección se introduce una nueva forma de representación de los documentos que incorpora no sólo las propiedades semánticas de las noticias sino también sus propiedades temporales y espaciales. Teniendo en cuenta esta representación, se propone una medida de semejanza que expresa que dos noticias abordan el mismo tópico si tienen una alta semejanza semántica, proximidad temporal y coincidencia espacial.

El sistema de detección propuesto categoriza, además, en un conjunto de temáticas pre-definidas a cada nodo de la jerarquía creada y proporciona un resumen de los contenidos asociados en cada uno de ellos. Para la obtención de las temáticas se desarrolló un nuevo algoritmo de desambiguación basado en *WordNet* que utiliza todas las relaciones semánticas y es capaz de desambiguar sustantivos, adjetivos y verbos.

La experimentación realizada con varias colecciones de prueba permite constatar, a partir de la comparación con los sistemas de detección existentes, las posibilidades que brinda *JERARTOP* como nuevo sistema de detección de tópicos.

ABSTRACT

The motivation behind this work is the Topic Detection problem. Starting from a continuous stream of newspaper articles, this problem consists in determining for each incoming document, whether it reports on a new topic, or it belongs to some previously identified topic. The intents to tackle this problem gave birth to the development of several algorithms and techniques for the dynamic structuralizing of the information and its description.

In this work different approaches to the clustering problem of dynamic collections of objects are presented and evaluated. In this sense, solutions to the three classification problems that could be presented in the practice are proposed: the obtaining of partitions or disjoint clusters, the obtaining of covers or overlapped clusters and the construction of the cluster hierarchies. Thus, three new clustering algorithms are presented: *Incremental Compact Algorithm*, *Incremental Strongly Compact Algorithm* and *Incremental Hierarchical Algorithm*. These algorithms are efficient, can manage mixed and incomplete object descriptions and they neither impose restrictions to the similarity function nor the cluster shapes. Also, the obtained clusters do not depend on the arrival order of the objects and these algorithms do not make irrevocable clustering assignments. Although we employ them to cluster document collections, its use is not restricted to this area, since they can be employed in any problem of the Pattern Recognition where to cluster objects of any nature is needed.

The need to provide efficient mechanisms to describe the topics detected by a Detection System led to the emergence of a summarization method starting from a partition in document clusters. Our method employs the calculus of typical testors as its primary operation and from them, it constructs the summaries of each cluster. Unlike the other methods in the literature, the selection of sentences is based on the discriminating frequent terms of each cluster which can be efficiently computed with the proposed algorithm *LEX*. This algorithm outperforms the other algorithms to calculate the typical testors.

The aforementioned algorithms were used in the Topic Detection problem. In this work we present an on-line detection system, namely *JERARTOP*, which goes beyond a traditional detection system, because it extracts the implicit knowledge in the stream of documents in an incremental and completely automatic way. This knowledge is expressed by a hierarchy of topic/subtopics, that is, our system identifies not only the topics but also the structure of events they comprise.

Unlike the detection systems proposed in the literature, this approach avoids the definition of what a topic is, since is the user who navigating by the created hierarchy, decides in what level he explores the knowledge of interest.

In our detection system, a new document representation method that not only incorporates the semantic properties of articles but also its temporal and spatial properties is introduced. Taking into account this representation, a similarity measure that states that two documents reporting a same topic should have a high semantic similarity, time proximity and place coincidence is proposed.

Moreover, the proposed detection system categorizes each node of the created hierarchy on the pre-specified subject matters and it provides a summary of the associated contents in each one of them. A new disambiguation algorithm based on *WordNet* was developed for the obtaining of the subject matters. It uses all the semantic relations and it disambiguates nouns, adjectives and verbs.

The experiments carried out using several test collections have demonstrated, starting from the comparison with the existent detection systems, the possibilities that *JERARTOP* offers as new system of topic detection.

ACRÓNIMOS

- TDT: Topic Detection and Tracking.
- DARPA: Defense Advanced Research Projects Agency.
- TIDES: Translingual Information Detection, Extraction and Summarization.
- LDC: Linguistic Data Consortium.
- FSD: First Story Detection.
- DET: Detection Error Tradeoff.
- TF: Term Frequency.
- IDF: Inverse Document Frequency.
- *K*-NN: *K* Nearest Neighbors.
- TFC: Term Frequency Component.
- ROI: Rules of Interpretation.
- DUC: Document Understanding Conference.
- MA: Matriz de Aprendizaje.
- MD: Matriz de Diferencias.
- MB: Matriz Básica.
- TT: Testor Típico.
- WSD: Word Sense Disambiguation.
- PLN: Procesamiento del Lenguaje Natural.
- IPTC: International Press and Telecommunications Council.
- TREC: Text Retrieval Conferences.

LISTADO DE ALGORITMOS, FIGURAS Y TABLAS

ALGORITMOS

FIGURAS

TABLAS

ÍNDICE

1 INTRODUCCIÓN

1.1. INTRODUCCIÓN

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.

La Detección y Seguimiento de Tópicos (TDT, sigla en inglés) 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.

Uno de los aspectos más importantes de este problema es definir qué se entiende por suceso. Un *suceso* se define en el estudio piloto TDT1 como la única cosa que ocurre en un instante de tiempo. Por ejemplo, la erupción del Monte Pinatubo el 15 de junio de 1991 es un suceso [Alla98, TDT298]. Una definición más reciente incorporó a esta definición la componente espacial, planteando que un suceso es algún hecho que ocurre en un instante de tiempo y lugar específicos, asociado con algunas acciones específicas [Dodd99, Yang00]. En la actualidad se ha ampliado 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 [TDT03]. Como veremos más adelante, nuestra aproximación evita la definición de qué es un tópico, ya que es el usuario el que navegando por la jerarquía creada decide en qué nivel explora el conocimiento de su interés.

En este trabajo nosotros nos concentraremos en el problema de la Detección de Tópicos, el cual consiste en identificar en un flujo continuo de noticias periodísticas aquellas que pertenecen a un nuevo tópico o a uno identificado previamente. El objetivo de un sistema de detección es, por tanto, agrupar las noticias que abordan el mismo tópico.

Existen varias aplicaciones prácticas que pueden desarrollarse a partir de una buena solución al problema de la detección de tópicos. Esta tarea en la actualidad es ejecutada manualmente por comerciantes, analistas financieros, analistas de los medios de comunicación y editores de periódicos y agencias en línea, los cuales tienen la misión de coleccionar, interpretar y presentar las noticias provenientes de múltiples fuentes. Un sistema que organice automáticamente la información en tópicos sería muy útil en aplicaciones financieras y noticiosas, donde los procesos de decisión se basan en los nuevos sucesos y la evolución de los sucesos existentes.

En la actualidad se han desarrollado diversos sistemas para la detección de tópicos, entre los que podemos mencionar: CMU [Yang98], UMASS [Alla00a], el sistema de

Papka [Papk99], BBN [Wall99], UPENN [Schu99], DRAGON [Lowe99], IBM [Dhar99], entre otros.

Los sistemas actuales de detección de tópicos tienen en común que procesan los artículos en un orden cronológico según la fecha de publicación de las noticias y que usan un algoritmo de agrupamiento no supervisado e incremental para agrupar las noticias en tópicos. Por ejemplo, el sistema presentado en [Yang98] usa el algoritmo *Single-Pass*, una ventana temporal que agrupa a los documentos que llegan y una función de semejanza que toma en cuenta la posición de las noticias en la ventana temporal. Por otra parte, [Papk99] usa un algoritmo *Single-Pass* y define un conjunto de clasificadores de documentos cuyos umbrales toman en consideración la adyacencia temporal de las noticias. El sistema UMass [Alla00a] usa un algoritmo *1NN* sobre la secuencia de noticias para la detección de tópicos.

Los sistemas de detección de tópicos actuales presentan varias limitaciones. En primer lugar, utilizan algoritmos de agrupamiento incrementales que dependen del orden de presentación de los documentos. Como consecuencia de esto, el conjunto de tópicos obtenidos puede ser diferente al variar el orden de presentación de los documentos. Además, la mayoría de estos sistemas se basan en variantes del algoritmo *Single-Pass*, donde la asignación de los documentos a los grupos es irrevocable. En segundo lugar, no se tiene en cuenta en el proceso de detección las propiedades temporales de las noticias o si se toman en consideración, sólo se restringen a la posición de los documentos en una ventana temporal. Finalmente, todos los sistemas existentes 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.

En esta tesis presentaremos un sistema de detección de tópicos que no presenta las limitaciones señaladas anteriormente. Para ello, proponemos nuevos algoritmos de agrupamiento incrementales que son independientes del orden de presentación de los objetos y que no realizan una asignación irrevocable de los objetos a los grupos.

Estos algoritmos están basados en técnicas del Reconocimiento Lógico Combinatorio de Patrones [Vale91, Shul99], el cual es una aproximación a los problemas del Reconocimiento de Patrones relativamente poco conocida en la literatura en inglés. Este enfoque no es sólo un conjunto de técnicas y métodos para la solución de estos problemas, sino que es una respuesta metodológica necesaria al hecho de que en muchos problemas de clasificación, que aparecen frecuentemente en la Medicina, Geociencias, Criminalística y otras, las descripciones de los objetos incluyen variables cualitativas y cuantitativas mezcladas incluso con ausencia de información. A partir de las condiciones reales del problema la principal tarea de esta metodología es obtener el modelo apropiado para su solución. En ningún caso, se parte de un modelo matemático general y en él se sumerge al problema real, sino que en esta metodología se llega al modelo matemático después de un proceso de modelación adecuado.

Desde la perspectiva de un periodista, una noticia acerca de un suceso típicamente especifica las siguientes preguntas: ¿qué ocurrió?, ¿cuándo el suceso ocurrió?, ¿quiénes están involucrados? y ¿dónde ocurrió?. Las aproximaciones actuales no distinguen estas propiedades, considerando todo el documento como una bolsa de términos. En cambio, nosotros pensamos que la efectividad de un sistema de detección podría mejorarse si los documentos se representaran de tal forma que captaran estas propiedades. En el trabajo, mostraremos experimentos y resultados que confirman esta hipótesis.

Por otra parte, con el crecimiento de la información disponible en fuentes en línea constituye una necesidad poder proporcionar mecanismos eficientes para encontrar, presentar y resumir información textual. Para un analista de información, que un sistema de detección le presente el conjunto de tópicos encontrados en una colección de noticias resulta, sin dudas, de mucho interés pero esto sólo no basta. ¿Cómo este analista podría captar de una rápida hojeada el contenido de estos tópicos, para profundizar sólo en los de su interés? Sería interesante, por tanto, poder brindar de cada tópico detectado por el sistema un resumen y/o una clasificación temática del mismo. Como veremos en esta tesis, nuestro sistema de detección incorpora técnicas con este objetivo.

En resumen, con el propósito principal de detectar los tópicos en un flujo continuo de noticias, vamos a intentar resolver las siguientes cuestiones:

- ¿Cómo hacer un mejor uso de las propiedades temporales y espaciales de las noticias en el proceso de detección de tópicos? ¿Repercutirán estas propiedades en la calidad de la detección?
- ¿Cómo compararemos a las noticias para decidir si abordan o no el mismo tópico?
- ¿Cómo obtener una jerarquía de tópicos con diferentes niveles de granularidad de modo que el analista de información pueda navegar por ella y seleccionar el tópico o subtópico de su interés?
- ¿Cómo proporcionar resúmenes de cada uno de los tópicos generados en la jerarquía anterior?
- ¿Cómo clasificar en temáticas generales a cada uno de los tópicos detectados para facilitar el análisis por parte de un usuario?

1.2. OBJETIVOS

En líneas generales, con esta tesis se pretende desarrollar algoritmos incrementales de agrupamiento basados en técnicas del Reconocimiento Lógico Combinatorio de Patrones que, junto al desarrollo de métodos de descripción conceptual de colecciones de textos, permitan el procesamiento dinámico de la información. Estas técnicas serán utilizadas, en particular, para la detección de tópicos en un flujo de noticias periodísticas.

Más concretamente, los objetivos de la presente tesis se podrían sintetizar en:

- Definir métodos de representación conceptual de las noticias, que logren captar, además de sus propiedades semánticas, sus propiedades temporales y espaciales.
- Definir medidas de semejanza entre documentos acordes con los métodos de representación propuestos.
- Desarrollar algoritmos incrementales para la estructuración de un conjunto de objetos basado en las técnicas del Reconocimiento Lógico Combinatorio de Patrones que reúnan los siguientes requerimientos:
    - Permitan la construcción de particiones, cubrimientos y jerarquías de grupos.
    - Los grupos construidos sean independientes del orden de presentación de los objetos, es decir, la estructuración es única una vez fijados el

criterio de agrupamiento y los criterios de similitud entre los valores de las variables y las descripciones de los objetos.

- No realicen una asignación irrevocable de los objetos a los grupos.
- No impongan restricciones a la medida de semejanza a utilizar ni a la forma de los grupos obtenidos.

- Desarrollar métodos de descripción conceptual de colecciones de documentos.
- Mostrar la viabilidad práctica de las técnicas desarrolladas, mediante su aplicación a la detección de sucesos y tópicos en un flujo dinámico de noticias.

1.3. METODOLOGÍA

La metodología que hemos propuesto para acometer la consecución de los objetivos de la tesis consta de seis fases. Éstas se resumen a continuación:

1. Estudio previo. Durante esta fase se analizarán las principales aproximaciones a la problemática de la detección de tópicos. Esta fase consta de las siguientes subfases:
   1.1. Estudio de los sistemas de detección de tópicos. En esta fase definiremos un marco conceptual común, bajo el cual, describiremos los sistemas de detección de tópicos más relevantes descritos en la literatura. Mediante este marco identificaremos las diferencias, ventajas y limitaciones de estos sistemas.
   1.2. Estudio de los métodos de representación de documentos y medidas de semejanza conceptuales entre documentos. En esta fase estudiaremos los distintos modelos de representación de documentos que se han utilizado en los sistemas de detección de tópicos, con el objetivo de introducir nuevas componentes que describan a los documentos, como por ejemplo, componentes que tengan en cuenta sus propiedades temporales y espaciales. De la misma forma, se estudiarán las distancias conceptuales que se han definido en las aproximaciones existentes en la literatura, con el objetivo de definir nuevas medidas de semejanza acordes con la representación de los documentos que se proponga.
   1.3. Estudio de los algoritmos de agrupamiento no supervisados e incrementales. En esta fase se hará un estudio de los algoritmos de agrupamiento no supervisados e incrementales propuestos en la literatura y se identificarán sus ventajas y limitaciones.
   1.4. Estudio de las técnicas de descripción de conjuntos de documentos. En esta fase se hará un estudio de los algoritmos propuestos en la literatura para la descripción de conjuntos de documentos.
2. Definición de métodos de representación de las noticias y medidas de semejanza adecuadas. En esta fase se propondrá un nuevo método de representación de las noticias que incorpore además de las propiedades semánticas, sus propiedades temporales y espaciales. Conjuntamente con esto,

se propondrá una medida de semejanza apropiada para este método de representación.

3. Desarrollo de algoritmos de agrupamiento incrementales. En esta fase se propondrán nuevos algoritmos de agrupamiento incrementales basados en las técnicas del Reconocimiento Lógico Combinatorio de Patrones que permitan la obtención de cubrimientos, particiones y jerarquías y no presenten las limitaciones detectadas en la fase 1.3.
4. Desarrollo de técnicas de descripción conceptual de conjuntos de documentos. En esta fase se desarrollarán técnicas de descripción de conjuntos de documentos que sean capaces no sólo de captar lo común en cada conjunto sino también lo que permite distinguir a un conjunto de otro.
5. Diseño de un sistema de detección de tópicos. En esta fase se propondrá un sistema de detección de tópicos que supere las limitaciones señaladas en la fase 1.1. Este sistema utilizará los métodos de representación de las noticias y la medida de semejanza propuestos en la fase 2, así como incorporará los algoritmos de agrupamiento y técnicas de descripción desarrolladas en las fases 3 y 4 respectivamente.
6. Evaluación. En esta fase analizaremos el impacto de los métodos y técnicas propuestas, evaluaremos la calidad del proceso de detección de tópicos y contrastaremos los resultados con los obtenidos por otros sistemas de detección existentes. Para la evaluación se seleccionarán un conjunto de colecciones de prueba y se utilizarán las medidas de evaluación de la calidad reportadas en la literatura.

1.4. ORGANIZACIÓN DE LA MEMORIA

La memoria se ha organizado en tres partes fundamentales. La **primera parte** está formada solamente por el capítulo 2 que constituye el estado del arte acerca de los sistemas de detección de tópicos. Los capítulos 3 y 4 forman la **segunda parte** de la memoria. Ellos contienen los algoritmos y herramientas propuestos para la organización dinámica de la información y que serán utilizados en nuestro sistema de detección de tópicos, a pesar de que su uso no está restringido a esta área. Finalmente, la **tercera parte** de la memoria está formada por los capítulos 5 y 6 donde se explica nuestro sistema de detección y se muestran los experimentos realizados y los resultados obtenidos. Las **conclusiones finales** de la memoria se exponen en el capítulo 7. A continuación explicaremos el contenido de cada capítulo con más detalle.

En el capítulo 2 se hace una introducción a la línea de investigación de la Detección y Seguimiento de Tópicos y se describen los principales sistemas de detección de tópicos reportados en la literatura bajo un marco conceptual común. De cada sistema se estudia su modelo de representación de las noticias, la medida de semejanza utilizada, el tratamiento que se hace de las propiedades temporales de las noticias y el algoritmo de agrupamiento empleado. Finalmente, el capítulo concluye con una comparación entre los sistemas anteriores y el análisis de sus limitaciones.

En el capítulo 3 se presentan tres nuevos algoritmos de agrupamiento no supervisados e incrementales. Cada uno de ellos incorpora un nuevo rasgo de utilidad en algunas aplicaciones. El primer algoritmo que presentaremos construye particiones de la colección de objetos a agrupar, es decir, los grupos son disjuntos. En cambio, el segundo algoritmo construye cubrimientos, es decir, los grupos obtenidos son

solapados. Finalmente, el tercer algoritmo es capaz de construir jerarquías de grupos con diferentes niveles de granularidad. Estos algoritmos se basan en un conjunto de propiedades matemáticas que son demostradas en este capítulo.

El capítulo 4 aborda la problemática de la descripción de conjuntos de documentos. En él se presenta un método efectivo para resolver el problema de resumir un conjunto de documentos, que está basado en la Teoría de Testores [Lazo01] y es independiente del dominio. A partir de un conjunto de grupos de documentos, nuestro método selecciona los términos frecuentes en cada grupo pero que además no lo son en el resto de los grupos. Una vez que se seleccionan estos términos, el método extrae las oraciones que logran cubrir a este conjunto y las ordena para producir resúmenes de calidad. Para lograr calcular de forma rápida los términos discriminantes de cada grupo se propone un nuevo algoritmo que calcula los testores típicos y que logró ser más eficiente que los propuestos en la literatura.

En el capítulo 5 se propone un sistema de detección de tópicos en línea, denominado *JERARTOP* que va más allá de un sistema tradicional de detección, ya que extrae o infiere de forma completamente automática e incremental el conocimiento implícito en el flujo de documentos. Este conocimiento se expresa en forma de una jerarquía de tópicos/subtópicos, es decir, el sistema permite identificar no sólo los tópicos presentes en un flujo de noticias sino también los posibles sucesos más pequeños que ellos comprenden. A diferencia de los sistemas de detección propuestos en la literatura, esta aproximación evita la definición de qué es un tópico, ya que es el usuario el que navegando por la jerarquía creada decide en qué nivel explora el conocimiento de su interés. El sistema de detección propuesto categoriza, además, en un conjunto de temáticas pre-definidas a cada nodo de la jerarquía creada y proporciona un resumen de los contenidos asociados en cada uno de ellos. Para la construcción de la jerarquía de tópicos y la descripción de cada uno de ellos, el sistema propuesto utiliza los algoritmos propuestos en los capítulos 3 y 4.

En el capítulo 6 se describe el entorno experimental con el objetivo de evaluar la efectividad del sistema de detección presentado en el capítulo anterior. En primer lugar, se discutirán los recursos utilizados en los experimentos: las colecciones de prueba, las herramientas utilizadas y las medidas de evaluación de la calidad que emplearemos. Finalmente, mostraremos los experimentos realizados y las conclusiones a las que hemos llegado. En estos experimentos analizaremos el impacto en la detección que tiene cada una de las componentes de la representación de las noticias, así como de las diferentes variantes de funciones de semejanza que pueden utilizarse. Además, optimizaremos los parámetros que utilizan nuestros algoritmos en función de las medidas de evaluación de la calidad.

En el capítulo 7 se presentan las principales aportaciones realizadas y las líneas de investigación futuras que podrían establecerse tomando como base este trabajo.

2 SISTEMAS DE DETECCIÓN DE TÓPICOS

2.1 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, sigla 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 [Alla98]. 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 [Alla98]. 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.

2.2 CONCEPTOS PRELIMINARES

2.2.1 Definiciones de suceso y tópico

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, sigla en inglés de *Linguistic Data Consortium*) para facilitar esta tarea ha identificado ciertos tópicos generales y proporcionado un conjunto de reglas para ello [TDT03].

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 [Alla98, TDT298]. 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 [TDT03].

El conjunto de acciones conectadas que tienen un enfoque común se define como **actividad** (campañas electorales, investigaciones, labores de socorro en desastres) [Papk99, TDT03].

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 [TDT03], en otras palabras, son las narraciones en las que se determina la ocurrencia de sucesos, de hechos.

Se considera que varias noticias 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

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

enfoque coherente alrededor del tópico, aún cuando no haya algún suceso subyacente claro [TDT298].

2.2.2 Principales tareas

Las tareas básicas definidas en el estudio TDT son las siguientes [TDT03]:

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

La figura 2.1 ilustra las nociones básicas de cada tarea [Wayn00].

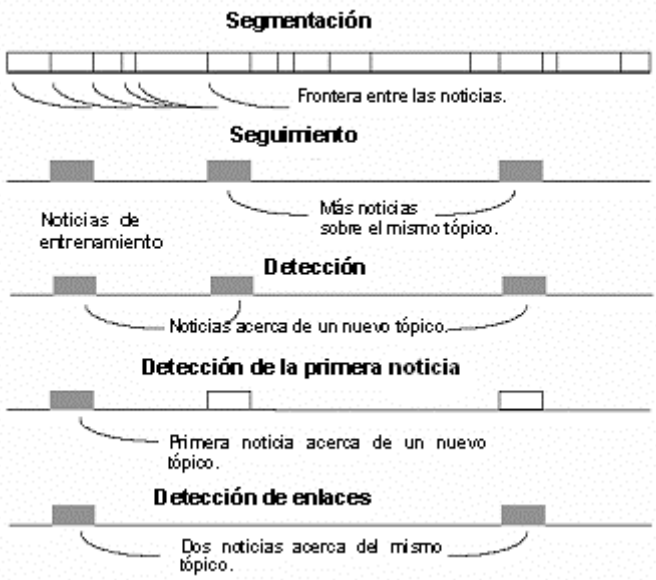


Fig. 2.1.- Tareas TDT.

La tarea de ***segmentación*** consiste en descomponer 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 identificar. La detección es, entonces, el problema de reconocer en un flujo de noticias aquéllas que pertenecen a un nuevo tópico o a uno no conocido previamente. En otras palabras, es una tarea de aprendizaje no supervisado, es decir, sin muestra de entrenamiento etiquetada [Alla98].

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) [Yang98, Papk99]. 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 y al finalizar el día 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 [Alla98].

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 identificar 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 [Alla00]. Esta tarea obviamente 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 con mayor eficacia. 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 [Shul95], 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 [Yang00] 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*** identifica 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.

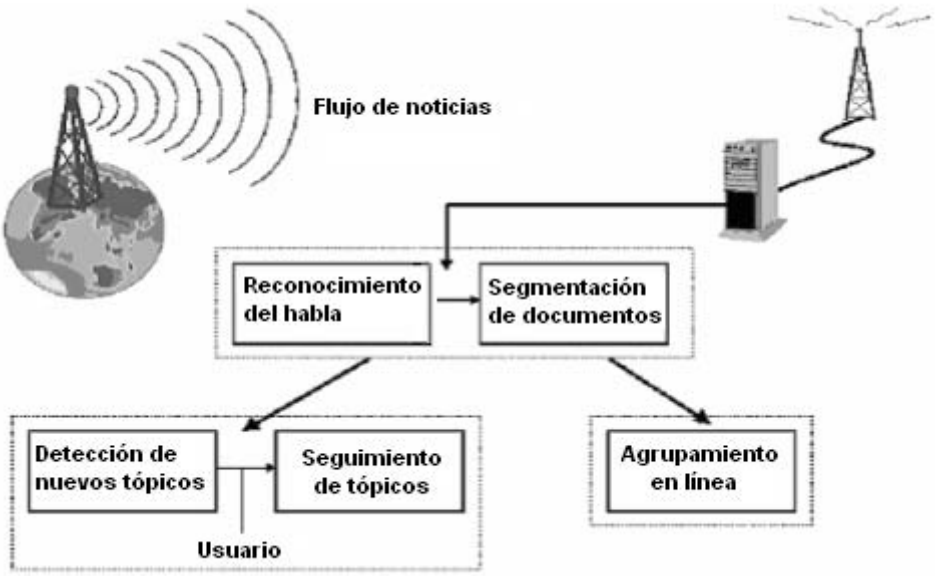


Fig. 2.2.- Arquitectura de un sistema de detección y seguimiento de tópicos.

Una vez analizadas cada una de las tareas de TDT, en la figura 2.2 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.

Las contribuciones de esta tesis se encuadran principalmente en la tarea de detección de tópicos.

2.2.3 Las colecciones de prueba

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 TDT1 fue desarrollado por los miembros del Proyecto de Investigación Piloto TDT y constituye el primer corpus de evaluación para las investigaciones en TDT. Éste contiene 15863 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 1^{ero} de julio de 1994 al 30 de junio de 1995, etiquetados en 25 tópicos. El corpus puede ser obtenido del Consorcio de Datos Lingüísticos[2]. 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).

La segunda fase de TDT, TDT2, coleccionó durante 26 semanas (de enero a junio 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*). TDT2 contiene aproximadamente 40000 noticias en inglés y 100 tópicos. El corpus TDT2 está dividido en tres partes con fines de investigación. El primer tercio del corpus (los datos coleccionados en Enero/Febrero de 1998) forma el conjunto de entrenamiento, el segundo tercio (Marzo/Abril de 1998), el conjunto de prueba y el último tercio (Mayo/Junio de 1998) forma el conjunto de evaluación [TDT298].

TDT3, desarrollado por LDC, es una colección más grande que contiene noticias de los meses de octubre a diciembre de 1998 provenientes de fuentes en inglés y chino mandarín con 120 tópicos etiquetados manualmente. Cada tópico etiqueta 350 documentos como promedio. Más detalles de estos corpus pueden encontrarse en [Graf00].

El corpus TDT4 comprende aproximadamente 28000 noticias en inglés publicadas en los meses de octubre de 2000 a enero de 2001. Está actualmente etiquetado en 40 tópicos.

2.2.4 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* [Rijs79] y el *Coste de Detección* [TDT298].

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

En TDT existen, además, dos tipos de errores: los errores por omisión (*misses*) y las falsas alarmas (*false alarms*) [TDT298]. 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

[2] http://www.ldc.upenn.edu/TDT/Pilot/TDT.Study.Corpus.v1.3.ps

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 siguiente [Alla00b]:

	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 [Yang99]:

- **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. Además, se cumple que $P_m = 1 - Relevancia$.

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}) [Pons02]:

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

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

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 [Pons02]:

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

Hasta aquí hemos 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 [Yang97].

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 [TDT298]:

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

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 [Pons02]:

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

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

Los parámetros $P_{tópico}$ (probabilidad a priori de que un documento sea relevante a un tópico), $coste_{f_a}$ y $coste_m$ se determinan empíricamente a partir de un corpus de

entrenamiento. En la evaluación de TDT2 se usó $P_{tópico}$ = 0.02 y las constantes $coste_{f_a}$=$coste_m$ = 1 [TDT298], mientras que en las evaluaciones posteriores de TDT se usó $coste_{f_a}$ = 0.1 y $coste_m$ = 1 [TDT03].

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 [TDT298]. 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*) [Mart97], 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* [Papk99]. Ejemplos de curvas DET se muestran en la figura 2.3.

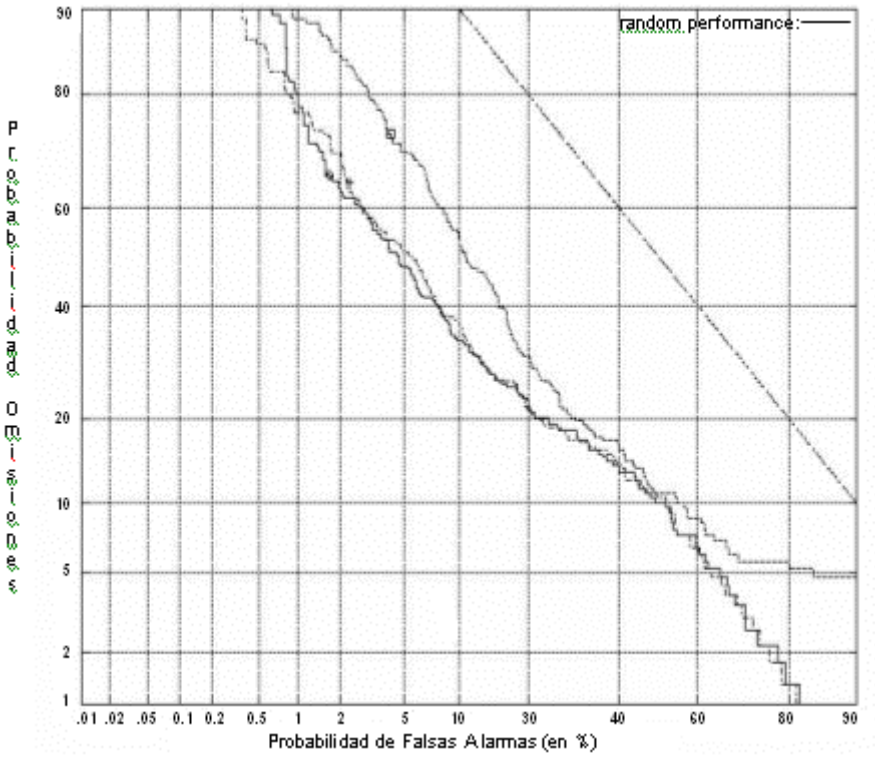


Fig. 2.3.- Ejemplos de Curvas DET.

2.3 Estructura general de un Sistema de Detección

En esta tesis, consideraremos 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 que permita agrupar las noticias en tópicos.

En los siguientes epígrafes describiremos 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 [Ragh86, Salt89] 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 [Gree00]. 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. Para tomar en consideración documentos de diferentes longitudes, es usual, que los vectores sean normalizados. La mayoría de los vectores de documentos son ralos.

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 $P(t_i \mid d)$ 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 de la colección, 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 del documento.

  Un análisis detallado de estos métodos de normalización puede verse en [Gree00].

- *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 la colección 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 [Call96].

- El pesado *ltc* es una variante del esquema *TF-IDF* que fue implementada en el sistema SMART 11.0 [Salt89] 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* [Robe95] y se define como:

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

  donde:

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

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

  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 mencionamos 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 2.4 muestra las técnicas de indexación más utilizadas.

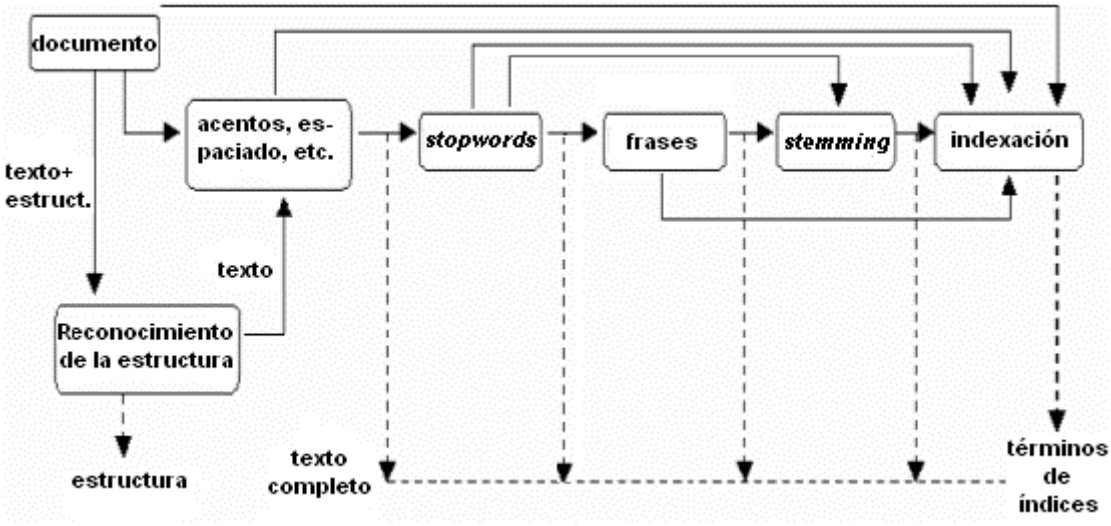


Fig. 2.4.- Técnicas de indexación.

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% [Rijs79].

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* [Alla00a]. 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) del epígrafe 2.3.1 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 comparemos 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:

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

Podríamos 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

Como mencionamos en el epígrafe 2.2.1 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 [Llid02].

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

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 libres, 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. Tampoco se conoce la cantidad de tópicos que podrían aparecer.
- 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* [Hill68], el *K-Means incremental* [Lars99], el de los *K vecinos más cercanos* (*K-NN*) y variantes del *Scatter/Gather* [Cutt92, Cutt93].

2.4 PRINCIPALES APROXIMACIONES EN LA DETECCIÓN DE TÓPICOS

En este epígrafe analizaremos las principales aproximaciones existentes en la detección de tópicos. De cada una de ellas estudiaremos 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 estudiaremos las características principales de cada uno de ellos.

2.4.1 El sistema CMU

El sistema CMU, desarrollado por la Universidad Carnegie Mellon [Yang98, Yang99, Carb99] 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* (ver epígrafe 2.3.2). 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) [Yang99].

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* [Cutt92, Cutt93] con el método *Fractionation* para la selección de los centroides y la estrategia de agrupamiento *Group-average* [Murt83, Will88]. Para la detección en línea, se usa el algoritmo *Single-Pass* [Hill68].

Detección retrospectiva [Yang98, Yang99]

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 por los autores en los experimentos realizados sobre la colección TDT1 [Yang98].

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 [Yang98, Yang99, Carb99]:

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 (ver epígrafe 2.3.2 *TF-IDF*), 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 [Yang99]:

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:

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

   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 TDT1 en la detección en línea los autores utilizaron un tamaño de ventana de 2500 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 [Alla00a] 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 [Papk98, Papk99, Papk99a], 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* (ver epígrafe 2.3.2). 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) (ver epígrafe 2.3.2). 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 (ver epígrafe 2.3.4).

   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 θ =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 [Papk99]:

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 por los autores en el corpus de evaluación TDT1 obtuvieron los mejores resultados para $\theta = 0.225$, $\beta = 0.000008$ y $n = 400$ [Papk98]. 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 [Schu99] 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* [Murt83, Will88] 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 [Dhar99] 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 [Dhar99a]:

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

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 $\Theta_c \le Ok(d,\bar{cl}*) \le \Theta_m$ 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 [Eich99] 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 por los autores en la colección TDT2 obtuvieron el C_{DET} más bajo (0.0078) para α=0.15 y un período de aplazamiento de 100. Este sistema obtiene, a nuestro juicio, demasiados grupos (de 2000 a 3000 tópicos).

2.4.7 El sistema de Kurt

El sistema de Kurt [Kurt01] 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. Nosotros nos referiremos en este epígrafe a 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 exceda 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 46539 noticias (de ellas sólo 1322 etiquetadas en 15 tópicos) provenientes de fuentes turcas y de Reuters, utilizaron como umbral de la ventana temporal 15, 10, 7 y 4 días y como $\alpha = 0.25$. 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 [Bran03] 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 por los autores con la colección TDT3 usando como colección de entrenamiento a la TDT2 obtuvieron costes de detección normalizados de 0.5783. Con la colección TDT4 usando como entrenamiento las colecciones TDT3 y TDT2 obtuvieron costes de 0.5691. Los autores realizaron experimentos en las colecciones TDT3 y TDT4 con un modelo del tiempo pero no obtuvieron mejoras en la eficacia de su 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. En la competición TDT2002 este sistema obtuvo el segundo lugar de los cuatro participantes.

2.4.9 El sistema Dragón

El sistema Dragón [Lowe99, Yamr00] usa aproximaciones estadísticas basadas en el modelo Beta-binomial [Gill90] 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 [Yamr00]:

- 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) del epígrafe 2.3.4). 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 [Alla98]. 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 [Lowe99].

2.4.10 El sistema BBN

Este sistema de detección [Wall99, Leek00] 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* (ver epígrafe 2.3.1), 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) del epígrafe 2.3.1)

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 [Wall99]:

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 [Spitt01] 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 $P(d \mid d_i)$. En el epígrafe 2.3.1 se describió el cálculo de estas probabilidades.

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 TDT2. En esta subcolección obtuvieron 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 Evaluación de los sistemas de detección en las colecciones de TDT

En este epígrafe mostraremos los resultados obtenidos por los diferentes sistemas de detección en línea reportados en las evaluaciones de TDT1, TDT2 y TDT3.

En la tabla 2.1 aparecen los resultados de CMU, UMASS, Dragón y el sistema de Papka en el corpus de evaluación TDT1 [Alla98, Papk99]. Como puede observarse, los valores de la medida F1 obtenidos son bajos.

Método	P_m (%)	P_{f_a} (%)	*F1*
CMU	59	1.43	0.40
UMASS	51	1.31	0.47
Dragón	58	3.47	0.28
Papka	54	1.13	0.46

Tabla 2.1.- Resultados obtenidos en el corpus de evaluación TDT1.

En la evaluación oficial de TDT2 se presentaron los sistemas de detección: BBN, CMU, Dragón, IBM, Iowa, UMASS y UPENN. Los resultados obtenidos se muestran en la tabla 2.2 [Schu99, Papk99]. 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	*Story-weighted*				*Topic-weighted*			
	P_m	P_{f_a}	C_{DET}	C_{DET} *Norm.*[3]	P_m	P_{f_a}	C_{DET}	C_{DET} *Norm.*
BBN	0.0941	0.0021	0.004	0.2	0.1295	0.0021	0.0047	0.235
UMASS	0.0913	0.0022	0.004	0.2	0.2091	0.0023	0.0064	0.32
Dragón	0.1638	0.0013	0.0045	0.225	0.1787	0.0013	0.0048	0.24
IBM	**0.1965**	**0.0007**	**0.0046**	**0.23**	**0.1766**	**0.0007**	**0.0042**	**0.21**
UPENN	0.2997	0.0011	0.0070	0.35	0.2617	0.0011	0.0063	0.315
Papka	0.65	0.0172	0.0297	1.485	-	-	-	-
CMU	0.3526	0.0004	0.0075	0.375	0.2644	0.0004	0.0057	0.285
Iowa	0.6028	0.0009	0.0129	0.645	0.4311	0.0009	0.0095	0.475

Tabla 2.2.- Resultados obtenidos en el corpus de evaluación TDT2.

Los resultados obtenidos en cuanto al coste de detección normalizado en la evaluación TDT3 se muestran en la figura 2.5 [Dodd00]. En esta competencia participaron 6 sistemas: BBN, IBM, Dragón, UMASS, CMU y la Universidad de Taiwán. Como puede observarse, nuevamente el menor coste de detección normalizado lo obtuvo el sistema IBM.

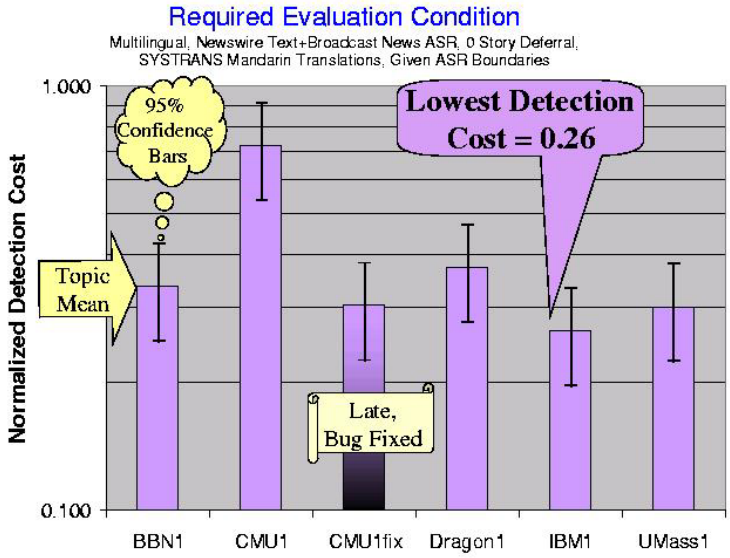


Fig. 2.5.- Resultados de la evaluación TDT3.

[3] En la evaluación TDT2 no se usaron los costes de detección normalizados.

2.6 CONCLUSIONES

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 2.3 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* estimados 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 variante de *TF-IDF*	Medida de *Okapi*	*Single-Pass*	Orden cronológico
Iowa	Vectorial con pesos *TF-IDF*	Coseno	Modelo de tubería	Orden 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 2.3.- 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.

3 ALGORITMOS DE AGRUPAMIENTO

3.1 INTRODUCCIÓN

Dado un conjunto de objetos definidos en términos de un conjunto de rasgos, los *algoritmos de agrupamiento* intentan construir particiones o cubrimientos de este conjunto, donde la semejanza intra-grupo sea máxima y la semejanza inter-grupos sea mínima.

En muchas aplicaciones la colección de objetos a agrupar es dinámica, es decir, nuevos objetos se incorporan constantemente a la colección. Un ejemplo de estas aplicaciones es la detección y seguimiento de tópicos en un flujo continuo de noticias. Los algoritmos de agrupamiento clásicos necesitan conocer todos los objetos para formar los grupos, por lo que cada vez que la colección de objetos se modifica, es necesario agrupar nuevamente la colección completa. Por eso, se necesitan algoritmos de agrupamiento capaces de actualizar los grupos cada vez que un nuevo objeto llega sin necesidad de empezar todo el proceso desde el principio. A este tipo de algoritmos se les llama *incrementales*.

En la literatura podemos encontrar algoritmos de agrupamiento incrementales, entre los que podemos mencionar: *Single-Pass* [Hill68], *K-Means Incremental* [Lars99], *el algoritmo de las estrellas* [Asla98], *GLC* [Sanc01], entre otros. En su mayoría estos algoritmos presentan como limitación la dependencia de los grupos obtenidos al orden de presentación de los objetos y se restringe a los grupos a tener formas esféricas o elipsoidales. Además realizan una asignación irrevocable de los objetos a los grupos o presentan el indeseado efecto de encadenamiento.

En este capítulo presentaremos tres nuevos algoritmos de agrupamiento incrementales que no presentan las limitaciones señaladas. Además cada uno de ellos incorpora un nuevo rasgo de utilidad en algunas aplicaciones. El primer algoritmo que presentaremos construye particiones de la colección de objetos a agrupar, es decir, los grupos son disjuntos. En cambio, el segundo algoritmo construye cubrimientos, es decir, los grupos obtenidos son solapados. Finalmente, nuestro tercer algoritmo es capaz de construir jerarquías de grupos con diferentes niveles de granularidad.

Como veremos más adelante en esta tesis, el sistema de detección de tópicos que propondremos utilizará los algoritmos propuestos en este capítulo.

Es bueno mencionar, que aunque estos algoritmos los emplearemos en el problema de la detección de tópicos, su uso no está restringido a esta área, sino que pueden utilizarse en cualquier problema del Reconocimiento de Patrones donde se necesite agrupar colecciones dinámicas de objetos.

Este capítulo está estructurado como sigue. En el próximo epígrafe daremos una clasificación general de los métodos de agrupamiento y clasificaremos a los tres algoritmos que propondremos. En la sección 3.2 estudiaremos en detalle el primer algoritmo propuesto: el compacto incremental y demostraremos matemáticamente las

propiedades en las que se basa. Del mismo modo, en la sección 3.3 presentaremos el algoritmo fuertemente compacto incremental. Finalmente, en la sección 3.4 nos referiremos al algoritmo jerárquico incremental y abordaremos sus principales características.

3.1.1 Métodos de agrupamiento

El proceso de agrupamiento de objetos envuelve algunos aspectos:

- La representación de los objetos que se van a agrupar.
- La selección de los atributos o rasgos que se tendrán en cuenta para la clasificación.
- La selección de la medida de semejanza acorde con el problema.
- Si se incorporará conocimiento del dominio.
- La selección del método de agrupamiento apropiado y sus parámetros.
- El tipo de grupos a crear, si son conjuntos duros o difusos, si estarán o no formando una jerarquía, si el número de grupos a formar está predeterminado o no.
- La forma en que se validarán los grupos obtenidos.
- Si la colección de objetos es dinámica, los requerimientos para la actualización constante de los grupos.

A lo largo de los últimos años, se han desarrollado diversos métodos de agrupamiento, que difieren en sus presupuestos teóricos o empíricos y, por tanto, producen estructuraciones en grupos diferentes. En general, podemos clasificar los métodos de agrupamiento en [Pons01]:

1. Atendiendo a si usan o no presupuestos de la teoría de las Probabilidades:
   - Probabilísticos.
   - No Probabilísticos.
2. Atendiendo a si se dispone de una muestra de los grupos a formar:
   - Supervisados.

     En un problema de clasificación supervisada existe un universo de objetos a clasificar dividido en clases. Se conocen, por tanto, las clases y se tiene una muestra de cada una de las clases que será usada para el entrenamiento del clasificador. Además se cuenta con una muestra de comprobación que permite validar la calidad de los grupos obtenidos.
   - No Supervisados.

     En un problema de clasificación no supervisada existe un universo de objetos a clasificar, no se conocen las clases en que se divide ese universo y el problema de la validación de los grupos obtenidos es aún un problema por resolver. Los problemas de clasificación no supervisada se dividen, a su vez, en clasificación *No Supervisada Restringida* y *Libre*, en dependencia de si se conoce a priori o no la cantidad de grupos deseados [Shul95].

3. Atendiendo a la forma de construcción de los grupos:
   - Jerárquicos.

     Los algoritmos jerárquicos proceden iterativamente considerando las semejanzas entre todos los pares de grupos y producen una secuencia anidada de particiones. El resultado puede ser mostrado gráficamente mediante un árbol llamado dendrograma, el cual proporciona una taxonomía o índice jerárquico. Típicamente son algoritmos avaros sin retrocesos.

     Existen dos aproximaciones básicas para generar grupos jerárquicos:

     - *Aglomerativo*: Comienza considerando a los objetos como grupos individuales y calculando la matriz de semejanza entre los grupos. Luego, en cada iteración se une el par de grupos más similar y se actualiza la matriz de semejanza. El proceso se repite hasta que quede un solo grupo.
     - *Divisivo*: Comienza con un solo grupo que contiene a todos los objetos y en cada iteración se divide un grupo en dos hasta que queden tantos grupos como objetos individuales existían. En cada paso es necesario decidir qué grupo se va a dividir y cómo se va a dividir.

   - Particionales.

     Los algoritmos particionales descomponen la colección en grupos de objetos, entre los cuales no existen relaciones jerárquicas. Para encontrar la estructuración óptima se define un criterio de "bondad" de la clasificación y se generan y evalúan todas las posibles estructuraciones para seleccionar la que mejor satisface dicho criterio. Debido a que la generación exhaustiva de todas las estructuraciones es en la práctica imposible, se han desarrollado técnicas heurísticas que logran realizar el agrupamiento con un bajo costo computacional y que utilizan diversos parámetros de entrada para controlar el agrupamiento, como por ejemplo, el número de grupos requerido, el mínimo o el máximo tamaño de los grupos, un umbral de semejanza intra-grupo, el grado de solapamiento de los grupos, etc. Estos algoritmos definen una función que evalúa la calidad de la estructuración, la cual es iterativamente optimizada. A diferencia de los jerárquicos, este tipo de algoritmo, analiza las decisiones tomadas a priori, moviendo los objetos entre grupos hasta que se alcanza una estructuración óptima.

4. Atendiendo a la forma en que procesan los objetos:
   - Incrementales.

     Los objetos son procesados secuencialmente y cada objeto es colocado en un grupo existente o forma uno nuevo.

   - No Incrementales.

     Asumen que todos los objetos están disponibles antes de comenzar el proceso de agrupamiento.

En este capítulo presentaremos tres nuevos algoritmos de agrupamiento no supervisados libres, no probabilísticos e incrementales. Los dos primeros son particionales y el último es jerárquico.

3.2 ALGORITMO COMPACTO INCREMENTAL

El agrupamiento de datos es uno de los problemas centrales en la Minería de Datos y de Textos. Entre las técnicas de agrupamiento existentes, el criterio de agrupamiento en conjuntos compactos tiene la propiedad de que la semejanza entre un par de objetos de un mismo grupo es máxima. Los grupos obtenidos por este criterio son relativamente pequeños y densos.

Existen muchas aplicaciones donde este criterio resulta de utilidad: la determinación de zonas con perspectividad semejante de ciertos minerales, el análisis de datos en problemas de clasificación supervisada determinando la estructura interna de las clases de entrenamiento y otros. En estos problemas se procesan conjuntos de datos estáticos, es decir, que no varían.

Existen otros problemas en los que el conjunto de datos se incrementa en el tiempo. Por ejemplo, el análisis del comportamiento de las facturas de una empresa, el estudio de las características sociales de los turistas que llegan a un hotel, la identificación y seguimiento de sucesos en un flujo de noticias, etc. Para este tipo de aplicaciones se requiere de un algoritmo incremental que obtenga los conjuntos compactos.

En este epígrafe se introduce un nuevo algoritmo incremental eficiente para estructurar un conjunto de datos en conjuntos compactos [Pons02d]. Este algoritmo se apoya en algunas propiedades de los conjuntos compactos que son demostradas a continuación.

3.2.1 Conceptos básicos

Sea ζ una colección de objetos y S una función de semejanza entre objetos simétrica, es decir, para todo $O_i,O_j \in \zeta$ se cumple que $S(O_i, O_j) = S(O_j, O_i)$. Aquí sólo consideraremos este tipo de función de semejanza. Sea, además, β_0 un umbral de semejanza definido por el usuario.

Definición 3.1. *Se dice que dos objetos O_i y O_j son β_0-semejantes si $S(O_i, O_j) \geq \beta_0$. Si para todo O_j de la colección de objetos ζ se cumple que $S(O_i, O_j) < \beta_0$ entonces O_i se denomina β_0-aislado.*

Definición 3.2. *Diremos que $NU \subseteq \zeta$, $NU \neq \varnothing$ es una componente conexa si se cumple que* [Mart00]:

1. $\forall O_i,O_j \in NU,\ O_i \neq O_j,\ \exists O_{i_1},\ldots,O_{i_q} \in NU\ [O_i=O_{i_1} \wedge O_j=O_{i_q} \wedge \forall p=1,\ldots,q\text{-}1\ S(O_p,O_{p+1}) \geq \beta_0]$.
2. $\forall O_i \in \zeta\ [O_j \in NU \wedge S(O_i,O_j) \geq \beta_0] \Rightarrow O_i \in NU$.
3. *Todo elemento β_0-aislado es una componente conexa (degenerada).*

Definición 3.3. *Diremos que $NU \subseteq \zeta$, $NU \neq \varnothing$, es un conjunto compacto si se cumple que* [Mart00]:

1. $\forall O_j \in \zeta\ [O_i \in NU \wedge \max_{\substack{O_t \in \zeta \\ O_t \neq O_i}} \{S(O_i,O_t)\} = S(O_i,O_j) \geq \beta_0\ ] \Rightarrow O_j \in NU$.

2. $[\max_{\substack{O_i \in \zeta \\ O_i \neq O_p}} \{S(O_p, O_i)\} = S(O_p, O_t) \geq \beta_0 \wedge O_t \in NU] \Rightarrow O_p \in NU.$

3. $|NU|$ *es mínimo.*

4. *Todo elemento* β_0*-aislado constituye un conjunto compacto (degenerado).*

La condición 1 dice que todo objeto de *NU* tiene en *NU* a sus objetos más β_0-semejantes. La condición 2 dice que no existe fuera de *NU* un objeto cuyo objeto más β_0-semejante esté en *NU*. La condición 3 dice que *NU* es el conjunto más pequeño que cumple las condiciones 1 y 2.

El criterio de agrupamiento basado en los conjuntos compactos forma, al igual que el de las componentes conexas, una partición. Esta partición es, además, única para cada conjunto de datos dado.

Ejemplo 3.1. Sean $\zeta=\{O_1,O_2,O_3,O_4,O_5,O_6,O_7,O_8,O_9,O_{10},O_{11},O_{12}\}$ y la matriz de semejanza siguiente, obtenida a partir de una función de semejanza *S*:

$$MS = \begin{array}{c} \\ O_1 \\ O_2 \\ O_3 \\ O_4 \\ O_5 \\ O_6 \\ O_7 \\ O_8 \\ O_9 \\ O_{10} \\ O_{11} \\ O_{12} \end{array} \begin{pmatrix} 1.0 & 0.85 & 0.75 & 0.78 & 0.63 & 0.70 & 0.51 & 0.38 & 0.43 & 0.27 & 0.22 & 0.13 \\ & 1.0 & 0.90 & 0.90 & 0.76 & 0.62 & 0.58 & 0.45 & 0.56 & 0.38 & 0.36 & 0.27 \\ & & 1.0 & 0.88 & 0.83 & 0.50 & 0.60 & 0.47 & 0.65 & 0.47 & 0.47 & 0.33 \\ & & & 1.0 & 0.85 & 0.58 & 0.68 & 0.55 & 0.63 & 0.45 & 0.42 & 0.30 \\ & & & & 1.0 & 0.48 & 0.73 & 0.65 & 0.80 & 0.65 & 0.58 & 0.50 \\ & & & & & 1.0 & 0.51 & 0.36 & 0.27 & 0.11 & 0.05 & 0.00 \\ & & & & & & 1.0 & 0.83 & 0.66 & 0.58 & 0.48 & 0.51 \\ & & & & & & & 1.0 & 0.66 & 0.63 & 0.53 & 0.61 \\ & & & & & & & & 1.0 & 0.83 & 0.80 & 0.70 \\ & & & & & & & & & 1.0 & 0.88 & 0.85 \\ & & & & & & & & & & 1.0 & 0.80 \\ & & & & & & & & & & & 1.0 \end{pmatrix}$$

Si consideramos $\beta_0 = 0.8$, entonces las componentes conexas son las siguientes:

$NU_1=\{O_1,O_2,O_3,O_4,O_5,O_9,O_{10},O_{11},O_{12}\}$, $NU_2=\{O_7,O_8\}$ y $NU_3=\{O_6\}$

y los conjuntos compactos son los siguientes:

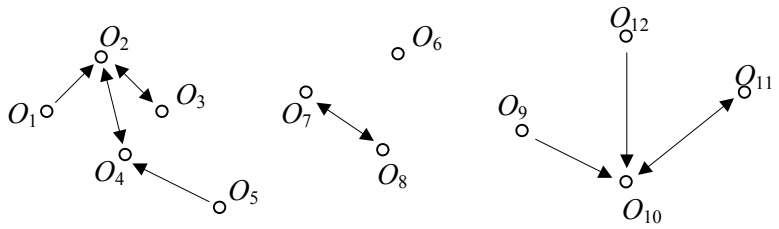
$NU_1=\{O_1,O_2,O_3,O_4,O_5\}$, $NU_2=\{O_9,O_{10},O_{11},O_{12}\}$, $NU_3=\{O_7,O_8\}$ y $NU_4=\{O_6\}$

Entre las componentes conexas y los conjuntos compactos se cumple la siguiente propiedad [Mart00]:

Toda componente conexa es la unión finita de conjuntos compactos.

Definición 3.4. *Sea S una función de semejanza entre objetos. Llamaremos grafo basado en la máxima* β_0*-semejanza según S, y lo denotaremos máx-S, al grafo orientado* $G = (\zeta, E)$ *cuyos vértices son los objetos de la colección* ζ *y existe un arco del vértice* O_i *al* O_j *si se cumple que* O_j *es el objeto más* β_0*-semejante a* O_i.

Ejemplo 3.2. El grafo *máx-S* correspondiente a la matriz de semejanza del ejemplo 3.1 con $\beta_0 = 0.8$ es el siguiente:



Proposición 3.1. *El conjunto de todos los conjuntos compactos de ζ coincide con el conjunto de todas las componentes conexas del grafo máx-S asociado a ζ sin tener en cuenta la orientación.*

Demostración. Es inmediato, a partir de la definición 3.4, que toda componente conexa del grafo *máx-S* asociado a ζ sin tener en cuenta la orientación, es un conjunto compacto de ζ.

Sea $C = \{O_{i_1}, O_{i_2}, ..., O_{i_k}\}$, $k>1$, un conjunto compacto y $G = (\zeta, E)$ el grafo *máx-S* asociado a ζ sin tener en cuenta la orientación.

Supongamos que C no es una componente conexa de G. Por lo tanto, existen $O_{i_j}, O_{i_r} \in C$, tales que entre ellos no existe un camino en G.

Entonces, construyamos el conjunto $NU = \{O_{i_j}\}$ y agreguemos a él todos los objetos O_{i_t} de C tales que:

$$\max_{\substack{O_{i_s} \in C \\ O_{i_s} \neq O_{i_j}}} \{S(O_{i_j}, O_{i_s})\} = S(O_{i_j}, O_{i_t}) \geq \beta_0 \text{ ó } \max_{\substack{O_{i_s} \in C \\ O_{i_s} \neq O_{i_t}}} \{S(O_{i_s}, O_{i_t})\} = S(O_{i_j}, O_{i_t}) \geq \beta_0$$

Se repite el proceso para cada objeto de C agregado a NU hasta que no se pueda agregar a nadie más.

Por construcción de NU el grafo *máx-S* asociado a NU sin tener en cuenta la orientación es conexo, $O_{i_r} \notin NU$ por la suposición y, además, NU es el conjunto más pequeño que satisface las condiciones 1 y 2 de la definición de conjunto compacto (3.3). Como $NU \subset C$, por la condición 3 de la definición de conjunto compacto, C no puede ser un conjunto compacto.

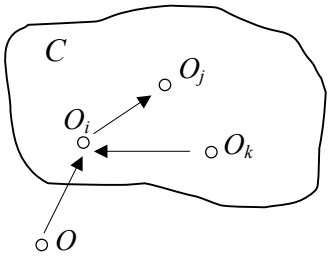
Si C fuera un conjunto unitario, es decir, O_{i_j} es un objeto β_0-aislado, entonces una componente conexa de G sería $\{O_{i_j}\}$, pues no existe ningún arco de O_{i_j} a cualquier otro vértice de dicho grafo. ∎

3.2.2 Nuevas propiedades de los conjuntos compactos

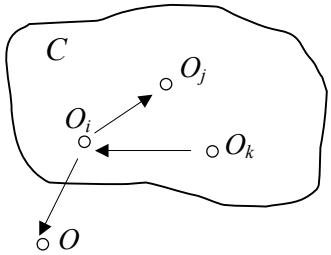
Definición 3.5. *Sean C un conjunto compacto de una colección de objetos ζ, S una función de semejanza y* β_0 *un parámetro definido por el usuario. Decimos que un objeto O está conectado con C si existe al menos un O'∈C que cumpla una de las dos condiciones siguientes:*

1. *O' es el objeto más* β_0*-semejante a O según S.*
2. *O es el objeto más* β_0*-semejante a O' según S.*

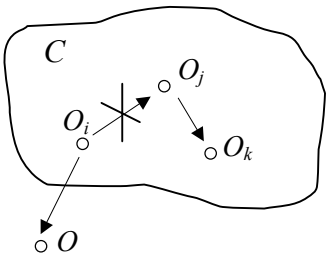
Un objeto *O* puede conectarse con un conjunto compacto *C* de varias formas. La figura 3.1 muestra todos los casos que pueden presentarse. En los casos I y II el objeto *O* no rompe ningún arco del grafo *máx-S* asociado a *C*. En los casos III y IV sí se rompen arcos.



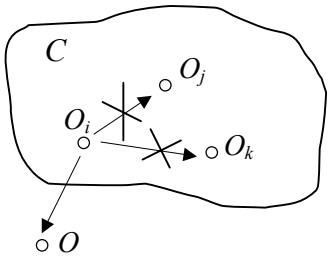
Caso I: O_i es el objeto más β_0-semejante a *O*.



Caso II: *O* y O_j son los objetos más β_0-semejantes a O_i.



Caso III: *O* es el objeto más β_0-semejante a O_i y O_j deja de serlo.



Caso IV: *O* es el objeto más β_0-semejante a O_i y O_j y O_k dejan de serlo.

Fig. 3.1.- Formas de conexión entre un objeto y un conjunto compacto.

Sean ζ una colección de objetos, *S* una función de semejanza, $\wp$ el conjunto de todos los conjuntos compactos de ζ, $C \in \wp$ un conjunto compacto, $G_c = (C,U)$ el grafo *máx-S* asociado a *C* y NG_c el grafo G_c sin tener en cuenta la orientación. Sea, además, *O* un objeto tal que $O \notin \zeta$.

Lema 3.1. *Si O no está conectado con C, entonces C será un conjunto compacto de* $\zeta \cup \{O\}$.

Demostración. Sea *G* el grafo *máx-S* asociado a ζ sin tener en cuenta la orientación. Por la proposición 3.1, NG_c es una componente conexa de *G*. Si *O* no está conectado con *C*, por definición de componente conexa de un grafo, *O* no puede pertenecer a dicha componente conexa. Por tanto, NG_c será también una componente conexa del grafo *máx-S* asociado a $\zeta \cup \{O\}$ sin tener en cuenta la orientación y, en consecuencia, *C* será un conjunto compacto de $\zeta \cup \{O\}$. ∎

Corolario. *Si O es* β_0*-aislado, entonces el conjunto de todos los conjuntos compactos de* $\zeta \cup \{O\}$ *es* $\wp \cup \{\{O\}\}$.

Lema 3.2. *Si O está conectado con C y O no rompió ningún arco de G_c, entonces O y todos los objetos de C pertenecerán a un mismo conjunto compacto de $\zeta \cup \{O\}$.*

Demostración. Por la proposición 3.1, NG_c es una componente conexa del grafo *máx-S* asociado a ζ sin tener en cuenta la orientación. Si *O* no rompió ningún arco de G_c, entonces NG_c sigue siendo conexa en el grafo *máx-S* asociado a $\zeta \cup \{O\}$ sin tener en cuenta la orientación. Si *O* está, además, conectado con *C*, entonces *O* pertenece a la misma componente conexa de los objetos de *C* y, por tanto, *O* y todos los objetos de *C* pertenecerán a un mismo conjunto compacto de $\zeta \cup \{O\}$. ∎

Definición 3.6. *Un punto de articulación de un grafo no orientado es un vértice v tal que cuando removemos v y todos los arcos incidentes en él, la componente conexa de v se divide en dos o más partes* [AhoH83].

Sean *G'* el grafo *máx-S* asociado a $\zeta \cup \{O\}$ y *H*, el subconjunto de objetos de *C* que pertenecen a la componente conexa de *O* en *G'* sin tener en cuenta la orientación.

Lema 3.3. *Si O está conectado con C y O rompe uno y sólo un arco de G_c (caso III), entonces si el conjunto $C \setminus H \neq \emptyset$, $C \setminus H$ es un conjunto compacto de $\zeta \cup \{O\}$.*

Demostración. Sea $O_i \in C$ el objeto al cual *O* es el más β_0-semejante y tal que se rompe su único arco *a* que incide al exterior. Si O_i no es un punto de articulación en NG_c, entonces al eliminar el arco *a* el grafo sigue siendo conexo y, por tanto, *H*=*C*.

Si O_i es un punto de articulación en NG_c, entonces el grafo $G'' = (C, U \setminus \{a\})$ sin tener en cuenta la orientación no es conexo. Si $C \setminus H \neq \emptyset$, entre cualesquiera dos objetos de $C \setminus H$ existe un camino en NG_c porque, de lo contrario, *C* no sería un conjunto compacto. Como O_i es un punto de articulación en NG_c, no existe en *H* ningún otro objeto conectado con $C \setminus H$.

Por tanto, $C \setminus H$ es una componente conexa en *G'* sin tener en cuenta la orientación, es decir, $C \setminus H$ es un conjunto compacto en $\zeta \cup \{O\}$. ∎

Teorema 3.1. *El conjunto de todos los conjuntos compactos de $\zeta \cup \{O\}$ es:*

$$\wp' = \left[\wp \setminus \left(N \cup R \cup M\right)\right] \cup \left[\bigcup_{i=1}^{|R|} \{R_i \setminus H_i\}\right] \cup \left[\{O\} \cup \bigcup_{i=1}^{|N|} N_i \cup \bigcup_{i=1}^{|R|} H_i \cup \bigcup_{i=1}^{|M|} J_i\right] \cup K$$

donde:

N es el conjunto de todos los conjuntos compactos de $\wp$ que están conectados con *O* y donde no se rompieron arcos y $N_i \in N$.

R es el conjunto de todos los conjuntos compactos de $\wp$ que están conectados con *O* y donde se rompió uno y sólo un arco y $R_i \in R$.

M es el conjunto de todos los conjuntos compactos de $\wp$ que están conectados con *O* y donde se rompió más de un arco y $M_i \in M$.

H_i es el conjunto de todos los objetos de R_i que pertenecen a la misma componente conexa de *O* en *G*' sin tener en cuenta la orientación.

J_i es el conjunto de todos los objetos de M_i que pertenecen a la misma componente conexa de *O* en *G*' sin tener en cuenta la orientación.

K es el conjunto de todos los conjuntos compactos que pueden formarse en cada $M_i \backslash J_i$, $i=1,...,|M|$.

Demostración. Como dijimos anteriormente, un nuevo objeto $O \notin \zeta$ puede conectarse sólo de cuatro formas diferentes con los conjuntos $C \in \wp$. Es importante observar que estos 4 casos no son excluyentes, incluso pudieran ocurrir todos a la vez varias veces cada uno. Por esta razón no es posible, ni necesario, analizar todas las posibles combinaciones. La ocurrencia de cada caso conllevará a una determinada modificación en el conjunto $\wp$.

La demostración de este teorema se apoya en los lemas 3.1, 3.2 y 3.3 y en la proposición 3.1.

A $\wp$' pertenecerán por el lema 3.1, todos los conjuntos compactos de $[\wp \setminus (N \cup R \cup M)]$, pues con ellos el nuevo objeto *O* no está conectado.

Además, por el lema 3.3, dado un conjunto compacto R_i, el conjunto $R_i \backslash H_i$ mantiene su propiedad de ser compacto si $R_i \backslash H_i \neq \varnothing$ y, por tanto, los conjuntos de $\left[\bigcup_{i=1}^{|R|} \{R_i \setminus H_i\}\right]$ también pertenecen a $\wp$'.

La componente conexa de *G*' a la que pertenece *O* ($\left[\{O\} \cup \bigcup_{i=1}^{|N|} N_i \cup \bigcup_{i=1}^{|R|} H_i \cup \bigcup_{i=1}^{|M|} J_i\right]$) es, por la proposición 3.1, un conjunto compacto. Note que a ella pertenecen todos los conjuntos compactos de *N* (según lema 3.2).

No es difícil ver que:

$$\zeta \setminus \left\{[\wp \setminus (N \cup R \cup M)] \cup \left[\bigcup_{i=1}^{|R|} \{R_i \setminus H_i\}\right] \cup \left[\{O\} \cup \bigcup_{i=1}^{|N|} N_i \cup \bigcup_{i=1}^{|R|} H_i \cup \bigcup_{i=1}^{|M|} J_i\right]\right\}$$

puede ser diferente del vacío. Esto es debido a la posibilidad de que *O* provoque la ruptura de más de un arco en al menos un conjunto compacto M_i. Es inmediato que no podemos afirmar que $M_i \backslash J_i$ es un conjunto compacto. Luego, a partir del conjunto $\left[\bigcup_{i=1}^{|M|} \{M_i \setminus J_i\}\right]$ se hace necesario construir el conjunto *K* de todos sus posibles conjuntos compactos. Observe que el cardinal de este conjunto es mucho más pequeño que el de $\zeta \cup \{O\}$. Con la inclusión de los conjuntos compactos existentes en *K* se construye completamente $\wp$'. ∎

3.2.3 Algoritmo compacto incremental

Este algoritmo se basa en los resultados teóricos demostrados anteriormente para construir de forma incremental los conjuntos compactos. Cuando un nuevo objeto llega, se calcula su semejanza con todos los objetos de los grupos existentes. Entonces, se crea un nuevo grupo formado por el nuevo objeto y todos los objetos conectados con él en el grafo de máxima β_0-semejanza (*máx-S*).

Cada vez que un objeto se añade al nuevo grupo, se elimina del grupo al que pertenecía. Puede suceder que al eliminar un objeto (que pertenece al conjunto compacto del nuevo objeto) del grupo al que pertenecía, provoque que ese grupo se quede inconexo y, por tanto, deje de ser un conjunto compacto. Finalmente, se

reconstruyen los grupos que potencialmente pueden haber perdido la propiedad de ser conjuntos compactos.

A continuación presentamos los pasos del algoritmo compacto incremental [Pons02d].

Entrada: Medida de semejanza *S*.
Umbral de semejanza β_0.

Salida: Conjuntos compactos.

Algoritmo

Paso 1. Cada vez que se presenta un nuevo objeto *O* se calcula la semejanza con todos los objetos de los conjuntos compactos existentes.

Paso 2. Se seleccionan los objetos que son más β_0-semejantes a *O* y aquellos a los que *O* es el más β_0-semejante. Si no existen tales objetos, en virtud del corolario del lema 3.1, el conjunto {*O*} es un nuevo conjunto compacto y terminar.

Paso 3. Los objetos seleccionados en el paso 2 pertenecen a conjuntos compactos de *N*, *R* ó *M* (ver teorema 3.1), luego:

a) Se crea un nuevo conjunto compacto formado por *O* y todos los objetos que pertenecen a la misma componente conexa de *O* en el grafo *máx-S* de $\zeta \cup \{O\}$ sin tener en cuenta la orientación. Note que esta componente conexa está formada por todos los objetos de los conjuntos de *N* y una parte de los objetos de cada uno de los conjuntos de *R* y *M*.

b) Estos objetos son eliminados de los conjuntos compactos a los que pertenecían.

c) Los conjuntos compactos que quedan vacíos se eliminan.

d) Las partes que quedaron de los conjuntos de *R* siguen siendo conjuntos compactos, en virtud del lema 3.3.

Paso 4. Puede suceder que al eliminar un objeto (que pertenece a la componente conexa de *O*) del conjunto compacto M_i de *M* al que pertenecía provoque que se quede inconexo y, por tanto, deje de ser un conjunto compacto. En cada parte de los conjuntos M_i que quedaron hacer:

a) Seleccionar un objeto.

b) Construir su componente conexa (por la proposición 3.1 sus objetos forman un conjunto compacto).

c) Eliminar los objetos de esa componente conexa de la parte de M_i.

d) Mientras queden objetos ir al paso 4a).

Algoritmo 3.1.- Algoritmo Compacto Incremental.

3.2.4 Análisis de la complejidad computacional

En este algoritmo cada objeto *O* tiene asociado el grupo al que pertenece y tres informaciones: $A(O)$ que contiene el o los objetos más β_0-semejantes a *O*, es decir, $\{O' \in \zeta \ / \ S(O,O') = \max_{\substack{O'' \in \zeta \\ O'' \neq O}} \{S(O,O'')\} \wedge S(O,O') \geq \beta_0\}$, $SimilMax(O)$, el valor de esa máxima similitud y $De(O)$ que contiene los objetos a los que *O* es el más β_0-semejante, es decir, $\{O' \in \zeta \ / \ O \in A(O')\}$.

La complejidad computacional de este algoritmo es $O(n^2)$, pues en el paso 1 cada objeto se compara con todos los existentes. El paso 3 conforma la componente conexa a la que el nuevo objeto pertenece. Este paso tiene complejidad $O(e)$, pues se trabaja con listas de adyacencia [Horo75]. Aquí e es la cantidad de aristas del grafo. Aunque desde el punto de vista teórico e es del orden n^2, en la práctica los grafos de máxima semejanza no son grafos completos y lo que ocurre con mucha frecuencia es que la cantidad de objetos más semejantes a uno dado es 1. Por eso, en nuestro caso, hemos estimado experimentalmente que $e=cn$, donde c es la cantidad máxima de objetos más semejantes a uno dado. En el paso 4 se reconstruyen las componentes conexas de los grupos que quedaron inconexos, lo cual, es $O(n+e)$.

Con relación a la complejidad espacial, nuevamente si tenemos en cuenta lo anterior, la complejidad sería $O(n)$, pues para cada objeto sólo tendríamos que almacenar el valor de su máxima semejanza y la lista de los objetos más β_0-semejantes, cuyo cardinal nuevamente podría aproximarse por una constante.

3.2.5 Discusión

En este epígrafe se presenta un nuevo algoritmo incremental de agrupamiento que crea una partición en conjuntos compactos de la colección de objetos.

La variante no incremental de este criterio de agrupamiento necesitaba almacenar la matriz de semejanza (de orden n^2) de los objetos de la colección lo que, sin dudas, constituye una gran limitación cuando se desea trabajar con colecciones dinámicas o grandes de objetos. La variante incremental desarrollada no necesita almacenar esta matriz.

Otra ventaja del algoritmo propuesto es que permite encontrar grupos con formas arbitrarias en oposición a los algoritmos incrementales como *K-Means Incremental* [Lars99] y *Single-Pass* [Hill68] que utilizan medidas centrales para generar los grupos, y como consecuencia, se restringe a los grupos a tener formas esféricas o elipsoidales.

Además el agrupamiento obtenido por el algoritmo propuesto es único, es decir, no depende del orden de presentación de los objetos, a diferencia del *K-Means Incremental* y el *Single-Pass*, que pueden producir diferentes agrupamientos al cambiar el orden de entrada de los objetos. Esta unicidad hay que entenderla de la siguiente manera: dados m objetos, la estructuración en conjuntos compactos es única, independientemente del orden en que los m objetos sean considerados. Es claro que en una población dinámica, la llegada de un nuevo objeto puede producir nuevas estructuraciones, pero de poblaciones diferentes. Sin embargo, en el caso de la estructuración en compactos, una vez que ha sido considerada una población determinada, los objetos que la conforman pudieron haber llegado en cualquier orden produciendo la misma estructuración.

El algoritmo propuesto no necesita fijar a priori el número de grupos a obtener, es decir, es aplicable a problemas de clasificación no supervisada libre.

El criterio de agrupamiento basado en los conjuntos compactos forma grupos disjuntos y más cohesionados y pequeños que los formados por las componentes conexas basadas solamente en la β_0-semejanza. Este último criterio es el utilizado en el algoritmo incremental *GLC* [Sanc01].

La complejidad computacional del algoritmo incremental presentado sigue siendo la misma que la de su variante no incremental, pero muy inferior a la aplicación reiterada del algoritmo no incremental.

El algoritmo propuesto puede ser utilizado en todas las tareas que requieran del agrupamiento de objetos en compactos y del procesamiento dinámico de la información, tales como la organización de la información, navegación, filtrado, distribución y la detección y seguimiento de sucesos en un flujo de documentos, el análisis del comportamiento de las facturas de una empresa, el estudio de las características sociales de los turistas que llegan a un hotel, entre otras aplicaciones. Los resultados obtenidos por este algoritmo en los experimentos realizados utilizando varios conjuntos de datos pueden verse en [Pons02d].

Es bueno resaltar que este algoritmo puede utilizarse, además, en problemas donde se deseen agrupar objetos de cualquier naturaleza, descritos por rasgos cuantitativos y cualitativos mezclados, incluso con ausencia de información.

3.3 ALGORITMO FUERTEMENTE COMPACTO INCREMENTAL

Otra de las técnicas de agrupamiento existentes es el criterio de agrupamiento en conjuntos fuertemente compactos, el cual tiene la propiedad de que la semejanza entre un par de objetos de un mismo grupo es máxima. Los grupos obtenidos por este criterio son solapados y relativamente pequeños y densos.

Existen muchas aplicaciones donde se necesita formar grupos de objetos de los que sabemos que por su naturaleza son solapados, por lo que este criterio resulta de utilidad. Existen, además, otros problemas en los que el conjunto de datos se incrementa en el tiempo. Por ejemplo, la organización de documentos en diferentes temáticas. Un documento que trate acerca de la mortalidad por el virus VIH en un país dado, aparecerá junto a otros que aborden el tema de la medicina, pero también formará parte de otros grupos de documentos, digamos por caso, que hablen de problemas sociales tales como la esperanza de vida al nacer o los que traten sobre los derechos humanos, entre otros. Estas estructuraciones pueden ser de utilidad para analistas de información de diferentes perfiles. Para este tipo de aplicaciones se requiere de un algoritmo incremental que obtenga los conjuntos fuertemente compactos.

En este epígrafe se introduce un nuevo algoritmo incremental eficiente para estructurar un conjunto de datos en conjuntos fuertemente compactos [Pons02c]. Este algoritmo se apoya en algunas propiedades de los conjuntos fuertemente compactos que son demostradas a continuación.

3.3.1 Conceptos básicos

Definición 3.7. *Sea* ζ *una colección de objetos y* β_0*, un parámetro definido por el usuario. Diremos que* $NU \subseteq \zeta$, $NU \neq \varnothing$, *es un conjunto fuertemente compacto si* [Mart00]:

1. $\forall O_j \in \zeta\ [O_i \in NU \wedge \max\limits_{\substack{O_t \in \zeta \\ O_t \neq O_i}} \{S(O_i, O_t)\} = S(O_i, O_j) \geq \beta_0\ ] \Rightarrow O_j \in NU.$
2. $\exists O_i \in NU\ \forall O_j \in NU\ \exists O_{i_1},...,O_{i_q} \in NU \quad [O_i = O_{i1} \wedge O_j = O_{i_q} \wedge \forall p<q\ [\max\limits_{\substack{O_t \in \zeta \\ O_t \neq O_{i_p}}} \{S(O_{i_p}, O_t)\} = S(O_{i_p}, O_{i_{p+1}}) \geq \beta_0\ ]].$
3. *No existe NU' que cumpla las condiciones 1 y 2 y* $NU \subset NU'$.

Todo elemento β_0-aislado constituye un conjunto fuertemente compacto (degenerado).

La condición 1 dice que todo objeto de *NU* tiene en *NU* al objeto más β_0-semejante. La condición 2 significa que en *NU* existe un objeto tal que para cualquier otro que pertenezca a *NU* existe una sucesión de objetos de *NU* tales que uno es el más β_0-semejante al siguiente. La condición 3 significa que *NU* es el conjunto más grande que cumple las condiciones 1 y 2. Este criterio forma grupos no disjuntos (solapados).

Ejemplo 3.3. Sean ζ y la matriz de semejanza del ejemplo 3.1. Si consideramos β_0=0.8, entonces los conjuntos fuertemente compactos son:

$NU_1=\{O_1,O_2,O_3,O_4\}$, $NU_2=\{O_2,O_3,O_4,O_5\}$, $NU_3=\{O_9,O_{10},O_{11}\}$, $NU_4=\{O_{10},O_{11},O_{12}\}$, $NU_5=\{O_7,O_8\}$ y $NU_6=\{O_6\}$

Entre las componentes conexas, los conjuntos compactos y los conjuntos fuertemente compactos definidos anteriormente se satisfacen las siguientes relaciones [Mart00]:

1. *Todo conjunto compacto es la unión finita de conjuntos fuertemente compactos.*
2. *Toda componente conexa es la unión finita de conjuntos fuertemente compactos.*

Definición 3.8. *Una componente fuertemente conexa de un grafo orientado es un conjunto maximal de vértices en el cual para todo vértice en el conjunto existe un camino a cualquier otro vértice del conjunto* [AhoH83].

Todo objeto β_0-aislado es una componente fuertemente conexa (degenerada).

3.3.2 Nuevas propiedades de los conjuntos fuertemente compactos

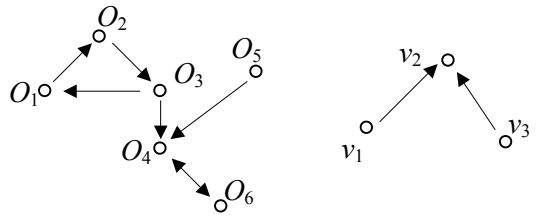
Definición 3.9. *Sea G = (X, E) un grafo orientado y $B \subset X$. B es una base del grafo G si se cumplen las dos condiciones siguientes* [Kake87]*:*

1. *Todo vértice del conjunto X es descendiente de al menos un vértice de B.*
2. *No existe en el conjunto X un subconjunto de menor número de elementos con dicha propiedad.*

Definición 3.10: *Sean G = (X ,E) un grafo orientado y P una partición del conjunto de vértices X, donde V_i denota a cada elemento de P. Se llama grafo reducido de G, y lo denotaremos como G_r = (V, E_r), al grafo cuyos vértices son los subgrafos generados por los elementos V_i de P y existe un arco del vértice v_i al v_j si existe un vértice x en V_i y un vértice y en V_j tal que existe el arco de x a y en G* [Kake87].

Definición 3.11. *Sea G un grafo orientado. Llamaremos grafo reducido según la fuerte-conexidad de G, y lo denotaremos como G_r/f-C, al grafo reducido de G asociado a la partición inducida por las componentes fuertemente conexas de G. En el grafo G_r/f-C los vértices son las componentes fuertemente conexas de G y existe un arco de un vértice v a otro v' en el grafo reducido, v≠v', si existe al menos un arco en G de un vértice en la componente fuertemente conexa que representa v a algún vértice en la componente fuertemente conexa que representa v'.*

Ejemplo 3.4. En la figura siguiente se muestra un grafo G y su grafo reducido según la fuerte-conexidad.



Grafo G

Grafo reducido según la fuerte-conexidad de G

Las componentes fuertemente conexas del grafo G son:

$v_1=\{O_1,O_2,O_3\}$, $v_2=\{O_4,O_6\}$ y $v_3=\{O_5\}$.

Las bases del grafo G son:

$\{O_1,O_5\}$, $\{O_2,O_5\}$ y $\{O_3,O_5\}$.

La base del grafo reducido según la fuerte-conexidad de G es $\{v_1,v_3\}$.

Proposición 3.2. *Sean S una función de semejanza y $G = (\zeta, E)$ el grafo máx-S. El grafo $G_r/f\text{-}C = (V, U)$ tiene una única base formada sólo por los vértices de grado interior nulo.*

Demostración. Como se demostró en [Kake87] el grafo reducido asociado a la partición inducida por las componentes fuertemente conexas de G no tiene circuitos. En efecto, si el grafo G fuera fuertemente conexo, su grafo reducido $G_r/f\text{-}C$ se reduce a un punto y, por tanto, no posee circuitos. En caso contrario, supongamos que $G_r/f\text{-}C$ tiene un circuito $c=(v_1,v_2,...,v_p,...,v_q,...,v_1)$. Consideremos dos vértices diferentes x y y de ζ, tales que x y y pertenecen a las componentes fuertemente conexas asociadas a v_p y v_q, respectivamente. Como c es un circuito, entre x y y existe un camino en ambos sentidos en G. Entonces, v_p y v_q no pueden ser componentes fuertemente conexas diferentes. Por tanto, $G_r/f\text{-}C$ no posee circuitos.

Por otro lado, supongamos que $v_1 \in V$ tiene grado interior no nulo y pertenece a la base de $G_r/f\text{-}C$. Sabemos que todo grafo admite al menos una base. Por tanto, existe al menos un $v_2 \in V$ tal que $<v_2, v_1> \in U$. v_2 no puede estar en la base, pues ningún elemento de una base puede ser descendiente de otro elemento de la base por definición de base. Si v_2 tiene grado interior nulo, v_2 no es descendiente de ningún elemento de V y tendría que estar en la base. Luego, v_2 tiene que tener grado interior no nulo. Esto significa que existe al menos un $v_3 \in V$ tal que $<v_3, v_2> \in U$ con $v_3 \neq v_1$, pues si no, se formaría un circuito. Como v_3 tiene grado interior no nulo seguimos el procedimiento. Dado que V es finito, existe un último v_n tal que $<v_n, v_{n-1}> \in U$ y v_n tiene grado interior no nulo, por lo que es inevitable utilizar algún v_i, $i \in \{1,...,n\text{-}1\}$, formándose un circuito. Por lo tanto, la base de $G_r/f\text{-}C$ está formada sólo por vértices de grado interior nulo. Como los vértices de grado interior nulo pertenecen a toda base de un grafo, esta base es única. ∎

Sean $G = (\zeta, E)$ el grafo *máx-S* , $G_r/f\text{-}C=(V,U)$ su grafo reducido, $B=\{b_1,...,b_r\}$ la base de $G_r/f\text{-}C$, $b_i \in B$ y $D(b_i)$ el conjunto de vértices descendientes de b_i en $G_r/f\text{-}C$.

Sea, además, F: $V \rightarrow \zeta$ tal que $F(v_i)$ es una componente fuertemente conexa en G. F es sobreyectiva porque $G_r/f\text{-}C$ es el grafo reducido de G.

Lema 3.4. $K = \bigcup_{v_j \in D(b_i)} F(v_j)$ *es un conjunto fuertemente compacto en* ζ.

Demostración. $K \neq \phi$, ya que $D(b_i) \neq \phi$. Probemos que K cumple las siguientes condiciones:

1. $\forall x \in \zeta\ [x' \in K \wedge \max_{\substack{x_j \in \zeta \\ x_j \neq x}} \{S(x', x_j)\} = S(x', x) \geq \beta_0\ ] \Rightarrow x \in K.$

Como $x' \in K$ existe al menos un j tal que $x' \in F(v_j)$, $v_j \in D(b_i)$. Luego, un elemento cualquiera x, más β_0-semejante a x', pertenece a $F(v_j)$ o pertenece a $F(v_t)$, $t \neq j$, $v_t \in V$.

Si $x \in F(v_j)$, entonces $x \in K$. Por el contrario, si $x \in F(v_t)$, implica que en el grafo reducido $G_r/f\text{-}C$ tiene que existir un arco de v_j a v_t, por lo que v_t sería descendiente de v_j y, por tanto, de b_i. Luego, $v_t \in D(b_i)$ y $x \in K$.

2. $\exists x' \in K\ \forall x \in K\ \exists x_{i_1}, ..., x_{i_q} \in K\ [\, x' = x_{i1} \wedge x = x_{i_q} \wedge \forall p < q$
$[\max_{\substack{x_t \in \zeta \\ x_t \neq x_{i_p}}} \{S(x_{i_p}, x_t)\} = S(x_{i_p}, x_{i_{p+1}}) \geq \beta_0\ ]].$

Sea $x' \in K$ tal que $x' \in F(b_i)$. Entonces, para todo $x \in K$ se cumple que $x \in F(b_i)$ ó $x \in F(v_j)$, $v_j \neq b_i$, $v_j \in D(b_i)$. En el primer caso, existen $x_{i_1}, ..., x_{i_q} \in F(b_i)$ tal que $x' = x_{i1} \wedge x = x_{i_q} \wedge \forall p < q\ [\max_{\substack{x_t \in \zeta \\ x_t \neq x_{i_p}}} \{S(x_{i_p}, x_t)\} = S(x_{i_p}, x_{i_{p+1}}) \geq \beta_0\ ]]$ porque $F(b_i)$ es una componente fuertemente conexa en el grafo G y, por tanto, $x_{i_1}, ..., x_{i_q} \in K$.

En el segundo caso, como $v_j \in D(b_i)$, existen $v_{i_1}, ..., v_{i_r} \in D(b_i)$ tal que $b_i = v_{i1} \wedge v_j = v_{i_r} \wedge \forall p < r\ \exists < v_{i_p}, v_{i_{p+1}} > \in U$. Basta con escoger los elementos $x_{i_t}\ t = 1, ..., q$ de la manera siguiente: $x_{i_1} = x'$, $x_{i_q} = x$. Como $< v_{i_p}, v_{i_{p+1}} >$ y $< v_{i_{p+1}}, v_{i_{p+2}} > \in U$ $\exists w, y, z, t\ \ w \in F(v_{i_p})$, $y, z \in F(v_{i_{p+1}})$ y $t \in F(v_{i_{p+2}})$ tal que y es el más β_0-semejante a w, t es el más β_0-semejante a z y como $F(v_{i_{p+1}})$ es una componente fuertemente conexa en G existe un camino en G entre y y z. Luego, existe un camino entre w y t en G. Por tanto, existe un camino entre x' y x en G.

3. No existe $K' \supset K$ que cumple las condiciones 1 y 2.

Supongamos que $\exists K' \supset K$ que cumple las condiciones 1 y 2. En $K' \setminus K$ no puede existir x que sea el más β_0-semejante a un $x' \in K$, pues tendría que estar en K. Luego, cualquier elemento de $K' \setminus K$ es el más β_0-semejante de uno de $K' \setminus K$ y/o tiene su más β_0-semejante en K. No puede ocurrir que todos los elementos de $K' \setminus K$ tengan su más β_0-semejante en $K' \setminus K$, ya que no se cumpliría la condición 2 (no habría manera de llegar de un elemento de $K' \setminus K$ a K ni tampoco habría manera de llegar de un elemento de K a uno de $K' \setminus K$). Luego, tiene que existir $x \in K' \setminus K$ que tenga su más β_0-semejante en K y no sea el más β_0-semejante de un $y \in K$. En virtud de lo anterior y de la condición 2 que también cumple K', tiene que existir un $x \in K' \setminus K$ tal que se pueda llegar a cualquier elemento de K.

Luego, existe v_j tal que $x \in F(v_j)$ y $b_i \in D(v_j)$ y b_i no sería un elemento de la base del grafo reducido G_r/*f-C*. Por tanto, no puede existir *K'*. ∎

Teorema 3.2. *El conjunto de todos los conjuntos fuertemente compactos de ζ es:*

$$\bigcup_{i=1}^{r} \left\{ \bigcup_{v_j \in D(b_i)} F(v_j) \right\}$$

Demostración. Supongamos que existe *K'* conjunto fuertemente compacto en ζ tal que no existe $b_j \in B$ que cumpla que $K' = \bigcup_{v_t \in D(b_j)} F(v_t)$. Sean $\{P_1,...,P_n\}$ la partición en componentes fuertemente conexas de *G*. Consideremos los $P_{i_1},...,P_{i_m}$ tal que $P_{i_j} \cap K' \neq \phi \ \forall j = 1,...,m$. Demostremos que $P_{i_j} \subset K' \ \forall j = 1,...,m$.

Sea P_{i_j} una componente fuertemente conexa de *G* con $j \in \{1,...,m\}$. Supongamos que $\exists x \in P_{i_j}$ tal que $x \notin K'$. Como $x \in P_{i_j}$, el grado interior de *x* debe ser diferente de cero. Pero si esto se cumple significa que $\exists x' \in K'$ tal que *x* es el más β_0-semejante a *x* y, en ese caso, *K'* no sería un conjunto fuertemente compacto. Por lo tanto, $\bigcup_{j=1}^{m} P_{i_j} = K'$.

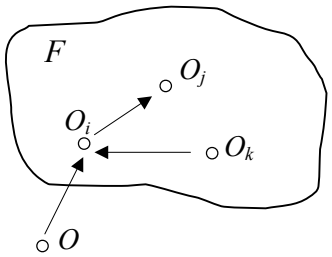
Sean $v_{i_j}, j = 1,...,m$ los vértices del grafo G_r/*f-C* correspondientes a $P_{i_j}, j = 1,...,m$. Por condición 2 de la definición de conjunto fuertemente compacto (3.7) $\exists x' \in K'$ tal que existe un camino entre *x'* y cualquier elemento de *K'*. Por tanto, entre el v_{i_t} tal que $x' \in F(v_{i_t})$ y cualquier otro v_{i_t}, $j \neq t$ existe un camino en G_r/*f-C*. Luego, $v_{i_j}, j = 1,...,m$, $j \neq t$ son descendientes de v_{i_t} en G_r/*f-C*. Pero, por la suposición, $v_{i_t} \notin B$, lo que significa que v_{i_t} tiene que ser descendiente de un $b_i \in B$ por definición de base. Luego, $K' \subset K = \bigcup_{v_j \in D(b_i)} F(v_j)$ y *K'* no sería fuertemente compacto, pues incumple la tercera condición de la definición 3.7. Luego, $v_{i_t} \in B$. ∎

Corolario. *El número de conjuntos fuertemente compactos de ζ es igual al cardinal de la base de G_r/f-C.*

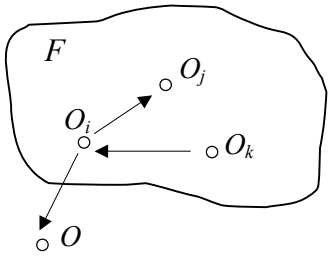
Definición 3.12. *Sean F un conjunto fuertemente compacto (compacto) de una colección de objetos ζ, S una función de semejanza y* β_0 *un parámetro definido por el usuario. Decimos que un objeto O está conectado con F si existe al menos un O' ∈ F que cumple una de las dos condiciones siguientes:*

1. *O' es el objeto más* β_0*-semejante a O según S.*
2. *O es el objeto más* β_0*-semejante a O' según S.*

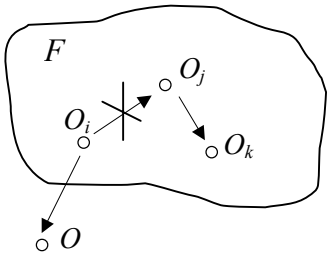
Un objeto *O* puede conectarse con un conjunto fuertemente compacto (compacto) *F* de varias formas. La figura 3.2 muestra todos los casos que pueden presentarse. En los casos I y II el objeto *O* no rompe ningún arco del grafo *máx-S* asociado a *F*. En los casos III y IV sí se rompen arcos.



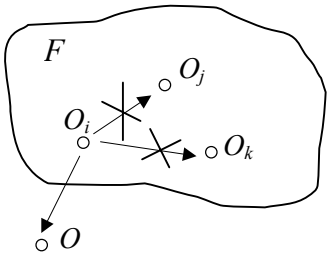
Caso I: O_i es el objeto más β_0-semejante a O.



Caso II: O y O_j son los objetos más β_0-semejantes a O_i.



Caso III: O es el objeto más β_0-semejante a O_i y O_j deja de serlo.



Caso IV: O es el objeto más β_0-semejante a O_i y O_j y O_k dejan de serlo.

Fig. 3.2.- Formas de conexión entre un objeto y un conjunto fuertemente compacto.

Sean ζ una colección de objetos, S una función de semejanza, $\wp$ el conjunto de todos los conjuntos fuertemente compactos de ζ, $F \in \wp$ un conjunto fuertemente compacto y $G_F=(F,U)$ el grafo *máx-S* asociado a F. Sea, además, O un objeto tal que $O \notin \zeta$.

Proposición 3.3. *Si O no está conectado con F, entonces F será un conjunto fuertemente compacto de $\zeta \cup \{O\}$.*

Corolario. *Si O es β_0-aislado, entonces el conjunto de todos los conjuntos fuertemente compactos de $\zeta \cup \{O\}$ es $\wp \cup \{\{O\}\}$.*

Proposición 3.4. *Si O es el objeto más β_0-semejante a un objeto de F y O no rompió ningún arco de G_F, entonces O y todos los objetos de F pertenecerán a un mismo conjunto fuertemente compacto de $\zeta \cup \{O\}$.*

Demostración. Como F es un conjunto fuertemente compacto $\exists x \in F \ \forall y \in F$ $\exists x_{i_1},...,x_{i_q} \in F$ [$x = x_{i1} \wedge$ $y = x_{i_q} \wedge$ $\forall p<q$ [$\max_{\substack{x_t \in \zeta \\ x_t \neq x_{i_p}}} \{S(x_{i_p}, x_t)\} = S(x_{i_p}, x_{i_{p+1}}) \geq \beta_0$]].

Luego, en particular, existe un camino en G_F de x a O' y, por tanto, de x a O. Por tanto, $F \cup \{O\}$ satisface la condición 2 de la definición de conjunto fuertemente compacto (3.7).

Como O es el más β_0-semejante a un objeto $O' \in F$ pueden ocurrir dos cosas:

a. Existe al menos $O'' \in F$ tal que O'' es el más β_0-semejante a O y, en ese caso, $F \cup \{O\}$ cumple la condición 1 y también la condición 3 de la definición de conjunto fuertemente compacto (3.7). Por tanto, $F \cup \{O\}$ sería un conjunto fuertemente compacto y O y todos los objetos de F estarían en el mismo conjunto fuertemente compacto.

b. Existe al menos $O''' \in \zeta \backslash F$ tal que O''' es el más β_0-semejante a O y, en ese caso, existe F' conjunto fuertemente compacto tal que $F' \supset F \cup \{O\}$. Luego, O y todos los objetos de F estarían en el mismo conjunto fuertemente compacto. ∎

Sean G' el grafo *máx-S* asociado a $\zeta \cup \{O\}$, C un conjunto compacto, tal que C es la unión de ciertos conjuntos fuertemente compactos, esto es, $C = \bigcup_{F_i \in \wp} F_i$, $G_c = (C,U)$ el grafo *máx-S* asociado a C, NG_c el grafo G_c sin tener en cuenta la orientación y H, el subconjunto de objetos de C que pertenecen a la componente conexa de O en G' sin tener en cuenta la orientación. Sabemos por el lema 3.3 que si O está conectado con C y O rompe uno y sólo un arco de G_c, entonces si el conjunto $C \backslash H \neq \emptyset$, $C \backslash H$ es un conjunto compacto de $\zeta \cup \{O\}$. Sea $O' \in C$ el objeto al cual O es el más β_0-semejante y tal que se rompe su único arco a que incide al exterior. Note que este arco a se romperá también en los conjuntos fuertemente compactos F_i donde pertenece el objeto O'.

3.3.3 Algoritmo fuertemente compacto incremental

Este algoritmo se basa en los resultados teóricos demostrados anteriormente para construir de forma incremental los conjuntos fuertemente compactos. Cuando un nuevo objeto llega, se calcula su semejanza con todos los objetos de los grupos existentes.

Para determinar los conjuntos fuertemente compactos se aprovecha la propiedad que éstos cumplen de ser un cubrimiento de los conjuntos compactos. Por eso, lo primero que se hace es hallar el conjunto compacto al que pertenece el nuevo objeto O y reconstruir los conjuntos compactos a partir de los conjuntos fuertemente compactos que perdieron conexiones producto de la llegada de O. A partir de cada uno de ellos se construyen los cubrimientos de los fuertemente compactos.

Para ello, se determina la base del grafo reducido asociado a la partición inducida por la relación de fuerte-conexidad de cada conjunto compacto y, a partir de ella, se construyen los conjuntos fuertemente compactos que conforman el cubrimiento de dicho compacto.

En el algoritmo 3.2 se muestran los pasos del algoritmo fuertemente compacto incremental [Pons02c].

Observación: Conservar la información de cuáles son los conjuntos fuertemente compactos que conforman un compacto resulta de mucha utilidad en el paso 4 dado que en virtud del lema 3.3 no sería necesario procesar a los objetos que pertenecen a conjuntos fuertemente compactos que conforman un compacto donde sólo se perdió un arco. Bastaría con colocar directamente a dicho compacto en la lista de compactos a procesar.

Note que en este algoritmo se aprovecha la propiedad que cumplen los conjuntos fuertemente compactos de ser un cubrimiento de conjuntos compactos. Por eso, en lugar de hallar los conjuntos fuertemente compactos en todo el conjunto de objetos que modificaron sus arcos producto de la llegada del nuevo objeto se construye el grafo reducido asociado a cada conjunto compacto modificado por separado y se aplica el teorema 3.2.

Entrada: Medida de semejanza S.

Umbral de semejanza β_0.

Salida: Conjuntos fuertemente compactos.

Algoritmo

Paso 1. Cada vez que se presenta un nuevo objeto O se calcula la semejanza con todos los objetos de los conjuntos fuertemente compactos existentes.

Paso 2. Se seleccionan los objetos que son más β_0-semejantes a O y aquellos a los que O es el más β_0-semejante. Si no existen tales objetos, en virtud del corolario de la proposición 3.3, el conjunto $\{O\}$ es un nuevo conjunto fuertemente compacto y terminar.

Paso 3. a) Se construye el conjunto compacto $C(O)$ al que pertenece O, con todos los objetos que pertenecen a la componente conexa de O en el grafo *máx-S* de $\zeta \cup \{O\}$ sin tener en cuenta la orientación (proposición 3.1). Se coloca el conjunto compacto construido en la lista de compactos a procesar.

b) Estos objetos son eliminados de todos los conjuntos fuertemente compactos a los que pertenecían. Sean estos conjuntos $FC_1,...,FC_n$, los cuales son eliminados de $\wp$.

Paso 4. Para cada objeto del conjunto $K = \left(\bigcup_{i=1}^{n} FC_i \right) \setminus C(O)$:

a) Se construye el conjunto compacto al que él pertenece y se coloca en la lista de compactos a procesar.

b) Se eliminan esos objetos del conjunto K.

Paso 5. Para cada compacto C en la lista de compactos a procesar se construye su cubrimiento en conjuntos fuertemente compactos:

a) Hallar las componentes fuertemente conexas del grafo *máx-S* asociado a C, G_c.

b) Construir el grafo reducido de G_c.

c) Determinar la base B del grafo reducido de G_c, que estará formada por todos los vértices del grafo reducido cuyo grado interior sea nulo.

d) Para cada elemento b_i de B:

i. Construir el conjunto fuertemente compacto F a partir de b_i, esto es, a F pertenecerán todos los vértices de G_c cuyos vértices correspondientes en el grafo reducido sean descendientes de b_i.

ii. Agregar F al conjunto de conjuntos fuertemente compactos $\wp$.

Algoritmo 3.2.- Algoritmo Fuertemente Compacto Incremental.

3.3.4 Análisis de la complejidad computacional

La complejidad computacional de este algoritmo es $O(n^2)$, pues en el paso 1 cada objeto se compara con todos los existentes. El paso 3 conforma el conjunto compacto (componente conexa) al que el nuevo objeto pertenece. Este paso tiene complejidad $O(e)$, pues se trabaja con listas de adyacencia [Horo75]. Aquí e es la cantidad de aristas del grafo. Como mencionamos en el epígrafe 3.2.4 hemos estimado experimentalmente que $e=cn$, donde c es la cantidad máxima de objetos más semejantes a uno dado. En el paso 4 se reconstruyen las componentes conexas de los grupos que perdieron arcos, lo cual, es $O(n+e)$. El paso 5 se encarga de construir las componentes fuertemente conexas

para, a partir de ellas, conformar los conjuntos fuertemente compactos. Este paso es $O(n+e)$, por lo que no sube la complejidad total del algoritmo.

La complejidad espacial (teniendo en cuenta lo anterior) es $O(n)$, pues para cada objeto sólo tendríamos que almacenar el valor de su máxima semejanza y la lista de los objetos más β_0-semejantes, cuyo cardinal nuevamente podría aproximarse por una constante.

3.3.5 Discusión

En este epígrafe se presenta un algoritmo de agrupamiento incremental que crea un cubrimiento en conjuntos fuertemente compactos de la colección de objetos. La variante no incremental de este criterio de agrupamiento necesitaba almacenar la matriz de semejanza (de orden n^2) de los objetos de la colección. La variante incremental desarrollada no necesita almacenar esta matriz.

Otra ventaja del algoritmo propuesto es que permite encontrar grupos con formas arbitrarias. Además el agrupamiento obtenido por el algoritmo propuesto es único, es decir, no depende del orden de presentación de los objetos. Esto lo diferencia de otros algoritmos con el mismo propósito (ej. *algoritmo de las estrellas* [Asla98]) que pueden producir diferentes agrupamientos al cambiar el orden de entrada de los objetos.

El algoritmo propuesto no necesita fijar a priori el número de grupos a obtener, es decir, es aplicable a problemas de clasificación no supervisada libre. La complejidad computacional del algoritmo incremental presentado sigue siendo la misma que la de su variante no incremental, pero muy inferior a la aplicación reiterada del algoritmo no incremental.

El algoritmo propuesto puede ser utilizado en todas las tareas que requieran del agrupamiento de objetos en fuertemente compactos y del procesamiento dinámico de la información, tales como, por ejemplo, navegación, filtrado, distribución; el estudio de la morbilidad de una población; la estructuración de la información para estudios socio-económicos; entre otras aplicaciones. Elementos como la *base* y los conjuntos fuertemente compactos pueden resultar de mucha utilidad para el analista de información. Los resultados obtenidos por este algoritmo en los experimentos realizados utilizando varios conjuntos de datos pueden verse en [Pons02c].

Es bueno resaltar que este algoritmo puede utilizarse, además, en problemas donde se deseen agrupar objetos de cualquier naturaleza, descritos por rasgos cuantitativos y cualitativos mezclados, incluso con ausencia de información.

3.4 Algoritmo jerárquico incremental

Como mencionamos en el epígrafe 3.1.1, los algoritmos de agrupamiento jerárquicos tradicionales se caracterizan por construir una jerarquía de grupos y proceden iterativamente considerando las semejanzas entre todos los pares de grupos para producir una secuencia anidada de particiones. Sin embargo, los niveles de las jerarquías generadas se corresponden con pasos del algoritmo de agrupamiento, es decir, en cada nivel se unen dos grupos del nivel anterior o se divide un grupo del nivel anterior, en dependencia de si el algoritmo jerárquico es aglomerativo o divisivo. Además estos algoritmos no son incrementales, pues necesitan de antemano conocer la colección de objetos que se desea agrupar.

Existen problemas prácticos donde se necesita crear jerarquías de objetos, pero donde cada nivel de la jerarquía represente diferentes niveles de abstracción de la colección de objetos original. Por ejemplo, en la organización de una colección de documentos de acuerdo a las temáticas que abordan, tales como los directorios existentes en la Web o para descubrir la estructura temporal de los tópicos y sucesos reportados por una colección de noticias, es decir, no sólo obtener los tópicos sino también los posibles sucesos en los que ellos se pueden descomponer. Nótese que en estos ejemplos, se necesita además que el algoritmo sea incremental por la naturaleza dinámica de las colecciones a agrupar.

En este epígrafe se introduce un nuevo algoritmo jerárquico incremental para estructurar un conjunto de datos en una jerarquía con diferentes niveles de abstracción [Pons03].

3.4.1 Requerimientos del algoritmo

El algoritmo jerárquico propuesto utiliza una rutina de agrupamiento que debe satisfacer los siguientes requerimientos:

- Debe ser incremental.
- No debe depender del orden de presentación de los objetos.

Algunos ejemplos de rutinas de agrupamiento que satisfacen estos requerimientos y que, por tanto, pueden ser utilizadas en nuestro algoritmo jerárquico son: el algoritmo *GLC* [Sanc01], el compacto incremental (ver epígrafe 3.2) y el fuertemente compacto incremental (ver epígrafe 3.3). El algoritmo *GLC* encuentra de forma incremental las componentes conexas del grafo de β_0-semejanza, mientras que el compacto incremental (fuertemente compacto incremental) está basado en la construcción incremental de los conjuntos compactos (fuertemente compactos) a partir del grafo de máxima β_0-semejanza.

Este algoritmo necesita además de un método para calcular los representantes de cada uno de los grupos (vea los distintos métodos que pueden utilizarse en el epígrafe 2.3.4).

3.4.2 Algoritmo jerárquico incremental

Debido a que el algoritmo propuesto es incremental se asume que existe una jerarquía de objetos, en la que hay que incorporar al nuevo objeto. El primer nivel de la jerarquía constituye un cubrimiento de la colección de objetos original, mientras que los restantes niveles están formados por los grupos que se forman a partir de los representantes de los grupos del nivel inferior de la jerarquía.

Cuando un nuevo objeto llega, deben analizarse todos los grupos existentes en todos los niveles de la jerarquía para decidir qué cambios se producen debido a la llegada del nuevo objeto. Primeramente, se aplica la rutina de agrupamiento al primer nivel de la jerarquía para incorporar el nuevo objeto al cubrimiento existente en este nivel. Esto puede provocar que grupos existentes se eliminen y/o que se creen nuevos grupos.

Cuando en un cierto nivel de la jerarquía se eliminan grupos, sus correspondientes representantes del siguiente nivel de la jerarquía deben ser eliminados de los grupos a los que pertenecen. De forma similar, cuando en un nivel de la jerarquía se crean nuevos grupos, deben calcularse los nuevos representantes para ser incorporados en el

cubrimiento correspondiente al siguiente nivel de la jerarquía. Todos los miembros de los grupos que sufrieron cambios (es decir, alguno de sus miembros fue uno de los representantes eliminados) y los nuevos representantes que se calcularon son añadidos a una cola de objetos a procesar para ser incorporados al cubrimiento existente en este nivel de la jerarquía. Con este propósito, se aplica nuevamente la rutina de agrupamiento para cada uno de los objetos incluidos en la cola. Este proceso se repite hasta que se alcanza el nivel tope de la jerarquía.

Es bueno señalar que en cada nivel de la jerarquía, la medida de semejanza y la rutina de agrupamiento no tienen que ser las mismas, por lo que pudieran emplearse distintas medidas de semejanza y rutinas de agrupamiento diferentes.

Analicemos un ejemplo. En la figura 3.3 (a) se muestra una posible jerarquía de grupos con 4 niveles. En el primer nivel de la jerarquía existen 10 grupos, mientras que en el segundo nivel existen 4 grupos. Como puede verse los representantes de los grupos 3, 4 y 5 del primer nivel de la jerarquía formaron un grupo en el segundo nivel de la jerarquía, mientras que los del 1 y 2 formaron otro grupo diferente.

Cuando el nuevo objeto *O* llega, se crea un nuevo grupo (el 11) con algunos objetos del grupo 2 y todos los miembros del grupo 3. Esto provoca la eliminación de los grupos 2 y 3 y la creación del grupo 2' con los objetos del antiguo grupo 2 que no se incorporaron al grupo de *O*. Es obvio que esto ha de afectar a los grupos formados en el segundo nivel de la jerarquía con los representantes de estos dos grupos y, a su vez, a los de los restantes niveles superiores vinculados con estos dos grupos del nivel primario.

La figura 3.3 (b) muestra los cambios que se producen en el segundo nivel de la jerarquía. Como puede observarse, los representantes $\bar{c}_2$ y $\bar{c}_3$ se eliminan y se crean dos nuevos representantes: $\bar{c}_2'$ y $\bar{c}_{11}$. Debido a que $\bar{c}_2$ se eliminó, el representante $\bar{c}_1$ debe ser incorporado a la cola de objetos a procesar, por pertenecer a un grupo que sufrió cambios. Por la misma razón, se incorporan los representantes $\bar{c}_4$ y $\bar{c}_5$. De esta forma, los representantes $\bar{c}_1$, $\bar{c}_4$, $\bar{c}_5$, $\bar{c}_2'$ y $\bar{c}_{11}$ se incorporan a la cola de objetos a procesar y se aplica la rutina de agrupamiento para incorporarlos a los grupos existentes en el segundo nivel de la jerarquía. Note que los grupos formados por los representantes de los grupos 6, 7, 8, 9 y 10 del primer nivel de la jerarquía no se alteran.

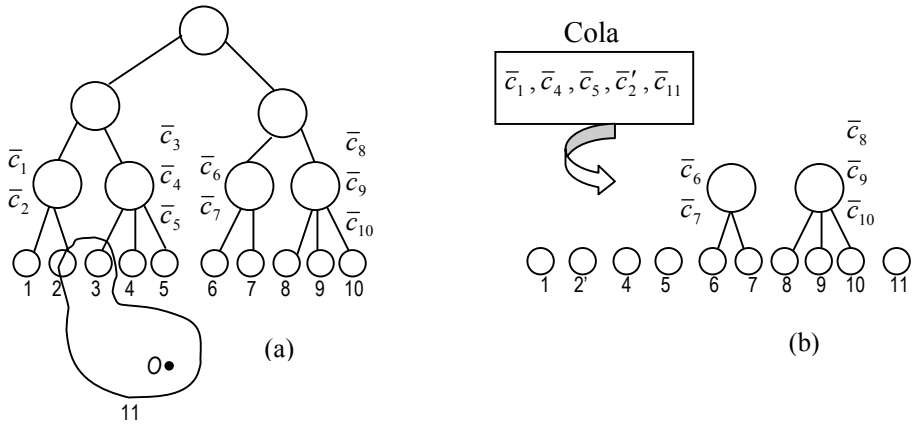


Fig. 3.3.- Una jerarquía de grupos.

Los pasos del algoritmo jerárquico incremental se muestran a continuación [Pons03].

Entrada: Medida de semejanza *S* y sus parámetros.
Rutina de agrupamiento y sus parámetros.

Salida: Jerarquía de grupos.

Algoritmo

Paso 1. Llegada del nuevo objeto *O*.
Nivel = 1

Paso 2. Aplicar la rutina de agrupamiento al primer nivel de la jerarquía de grupos existente.

Paso 3. Sea *GE* el conjunto de los grupos que fueron eliminados cuando se aplicó la rutina de agrupamiento.
Sea *NG* el conjunto de los nuevos grupos que se formaron.
Incrementar *Nivel* en 1.
Sea *Q* la cola con los objetos a procesar en *Nivel*, *Q* = ∅.

Paso 4. Eliminar en *Nivel* todos los representantes de los grupos que pertenecen a *GE*.
Añadir a la cola *Q* a todos los miembros de los grupos en *Nivel* que tienen al menos un representante de los que fueron eliminados.
Eliminar estos grupos de la lista de grupos existentes en este nivel.

Paso 5. Calcular los representantes de los grupos que pertenecen a *NG* y añadirlos a la cola *Q*.

Paso 6. Para cada objeto de la cola *Q* aplicar la rutina de agrupamiento.

Paso 7. Si *Nivel* no es el tope de la jerarquía ir al paso 3.

Algoritmo 3.3.- Algoritmo Jerárquico Incremental.

3.4.3 Discusión

En este epígrafe hemos presentado un nuevo algoritmo de agrupamiento jerárquico incremental que obtiene una jerarquía de grupos, donde cada nivel representa un nivel de abstracción diferente de la colección de objetos original. En cada nivel de la jerarquía existe un conjunto de grupos de objetos abstractos, los cuales son precisamente los representantes de los grupos del nivel inferior.

La jerarquía obtenida por el algoritmo puede ser útil para los especialistas poder analizar colecciones de objetos desconocidas. Su principal ventaja es que el conjunto de grupos generados en cada nivel de la jerarquía es independiente del orden de presentación de los objetos.

3.5 Conclusiones

En este capítulo se presentan tres nuevos algoritmos incrementales de agrupamiento que se enmarcan dentro de la clasificación no supervisada.

El primero de ellos crea particiones de la colección de objetos, mientras que el segundo crea cubrimientos. Las variantes no incrementales de los algoritmos compacto y fuertemente compacto necesitaban almacenar la matriz de semejanza (de orden n x n) de los objetos de la colección lo que, sin dudas, constituye una gran limitación cuando se desea trabajar con colecciones grandes de objetos. Las variantes incrementales

desarrolladas no necesitan almacenar esta matriz. La complejidad computacional de las variantes incrementales de estos algoritmos sigue siendo la misma que la de las variantes no incrementales, pero muy inferior a la aplicación reiterada de los respectivos algoritmos no incrementales.

El tercer algoritmo propuesto crea jerarquías de grupos con diferentes niveles de abstracción. Para la obtención de los grupos en cada nivel de la jerarquía pudieran emplearse los algoritmos compacto y fuertemente compacto incrementales.

Una ventaja de los algoritmos propuestos es que permiten encontrar grupos con formas arbitrarias en oposición a los algoritmos como *K-Means Incremental* y *Single-Pass* que utilizan medidas centrales para generar los grupos, y como consecuencia, se restringe a los grupos a tener formas esféricas o elipsoidales.

Además el agrupamiento obtenido por los algoritmos propuestos es único, es decir, no depende del orden de presentación de los objetos, a diferencia del *K-Means Incremental*, *Single-Pass* y el *algoritmo de las estrellas* que pueden producir diferentes agrupamientos al cambiar el orden de entrada de los objetos. Ninguno de los algoritmos propuestos necesita fijar a priori el número de grupos a obtener.

Los tres algoritmos propuestos pueden ser utilizados en todas las tareas que requieran del agrupamiento de objetos y del procesamiento dinámico de la información, tales como organización de la información, navegación, filtrado, distribución y la detección y seguimiento de sucesos en un flujo de documentos, entre otras. Además, pueden utilizarse en problemas donde se deseen agrupar objetos de cualquier naturaleza, descritos por rasgos métricos, espaciales, cualitativos, y categóricos mezclados, incluso con ausencia de información.

4 TÉCNICAS PARA LA DESCRIPCIÓN DE CONJUNTOS DE DOCUMENTOS

4.1 INTRODUCCIÓN

Con el crecimiento de la información disponible en fuentes en línea es una necesidad poder proporcionar mecanismos eficientes para encontrar y presentar información textual. Los sistemas de Recuperación de Información convencionales muestran un conjunto de documentos ordenados según su relevancia a la consulta formulada por un usuario.

Más recientemente se han desarrollado métodos para resumir un documento. Sin embargo, estas técnicas no están todavía a la altura de los requerimientos de un sistema de recuperación actual [Gold00].

Considere la situación donde un usuario formula una consulta, por ejemplo, relacionada con una noticia y obtiene cientos de documentos relevantes. Muchos documentos recuperados tendrán prácticamente la misma información y los resúmenes de los documentos individuales ayudan, pero probablemente serán muy similares entre sí.

Es necesario, por tanto, un sistema de construcción de resúmenes que tome en consideración los otros resúmenes ya generados y capte lo común a un conjunto de documentos pero además lo que distingue a unos documentos de otros.

En la literatura, el problema de resumir un conjunto de documentos (llamado *multidocument summarization*) ha recibido mucha atención en los últimos años (véase por ejemplo, [Mani97, Barz02, Hira03]). DUC (*Document Understanding Conference*) [4] es una conferencia que se celebra cada año con el objetivo de estimular el desarrollo de los sistemas de construcción automática de resúmenes.

Básicamente, un sistema de construcción de resúmenes de documentos trata de determinar cuáles oraciones deben incluirse en el resumen y cómo organizarlas para hacerlo comprensible. La mayoría de las aproximaciones se basan en una función que le asigna un grado de importancia a cada una de las oraciones, tomando en consideración la posición de las oraciones en los documentos, su longitud y los términos que ellas incluyen [Mani01]. Para ello, se aplican técnicas estadísticas (análisis de frecuencia, análisis de varianza, etc.) y técnicas lingüísticas (*tokens*, nombres, resolución de anáfora, análisis del discurso, etc.). De esta forma, las oraciones presentes en un conjunto de documentos pueden ser ordenadas con el fin de seleccionar las más apropiadas para el resumen.

Una de las principales limitaciones de estos procedimientos es que son lentos, porque el peso o importancia de una oración depende si se han seleccionado o no otras oraciones [Marc01].

---

[4] http://www-nlpir.nist.gov/projects/duc/

En general, un sistema de construcción de resúmenes de conjuntos de documentos debe satisfacer los siguientes requerimientos [Gold00]:

- Agrupamiento: habilidad de agrupar informaciones semejantes incluidas en los documentos.
- Cubrimiento: habilidad de encontrar y extraer los temas principales de los documentos.
- Anti-redundancia: habilidad de minimizar la redundancia de la información extraída.
- Cohesión: habilidad de combinar la información de una manera útil para el lector.
- Coherencia: habilidad de generar resúmenes legibles y relevantes al usuario.

En este capítulo presentaremos un método efectivo para resolver el problema de resumir un conjunto de documentos, que está basado en la Teoría de Testores [Lazo01] y es independiente del dominio. A partir de un conjunto de grupos de documentos, nuestro método selecciona los términos frecuentes en cada grupo pero que además no lo son en el resto de los grupos. Una vez que se seleccionan estos términos, el método extrae las oraciones que logran cubrir a este conjunto y las ordena para producir resúmenes de calidad.

Para lograr calcular de forma rápida los términos discriminantes de cada grupo es preciso un algoritmo que calcule de modo eficiente los testores típicos. Por eso, primeramente, presentamos en este capítulo un nuevo algoritmo para el cálculo de los testores típicos que logró ser más eficiente que los propuestos en la literatura [Sant03].

4.2 ALGORITMO PARA EL CÁLCULO DE LOS TESTORES TÍPICOS

Uno de los problemas del Reconocimiento de Patrones es la selección de variables, que se utiliza para reducir de modo eficiente el número de variables (atributos o rasgos) con los cuales se deben describir los objetos y para encontrar los rasgos que inciden de manera determinante en un problema. En el Reconocimiento Lógico Combinatorio de Patrones [Shul95] una alternativa de solución al problema de la selección de variables es a partir de la utilización del conjunto de testores típicos.

El concepto de testor aparece por primera vez en un trabajo de los científicos rusos Cheguis y Yablonskii [Cheg55] en un intento por tratar de simplificar la tarea de encontrar desperfectos en circuitos eléctricos que podían ser modelados a través de funciones del álgebra de la lógica.

En 1966 un grupo de científicos rusos, encabezado por Yu. I. Zhuravlev, dan un nuevo enfoque al concepto de testor [Dmit66] que permite aplicaciones importantes vinculadas a los problemas de clasificación con aprendizaje y selección de variables.

En esencia, un testor es un conjunto de características (rasgos) que diferencia a elementos (objetos) de clases distintas. Un testor típico es un testor al que si le eliminamos cualquiera de sus rasgos pierde la propiedad de ser testor.

Posteriormente este concepto ha sido extendido y variado para ajustarlo a otras aplicaciones. Ejemplos de estas extensiones son: los testores de Aizenberg y Tsipkin [Aize71] que consideran los rasgos asociados a dos predicados diferentes, los testores difusos de Goldman [Gold80] que se generan utilizando criterios de comparación reales,

los testores de Andreev [Andr81] que trabajan con clases no disjuntas, los *k*-testores primos de Shulcloper y Lazo [Shul91] que constituyen una extensión de los testores a la lógica *k*-valente, entre otros. Una panorámica de la evolución del concepto de testor puede verse en [Lazo01].

Los testores típicos son determinantes en problemas como la evaluación de la importancia informacional de los rasgos y la selección de variables, pues permiten reducir la dimensión del espacio de representación de los objetos y como sistemas de conjuntos de apoyo en múltiples algoritmos de clasificación, como por ejemplo ALVOT y KORA-Ω [Shul95]. Algunos trabajos donde se usan los testores típicos para evaluar la importancia informacional de los rasgos son: [Alba98], [Lazo99], entre otros.

En este epígrafe se expone un nuevo algoritmo para el cálculo de todos los *testores típicos* de una matriz de aprendizaje. Cuando hablemos de testores típicos nos restringiremos aquí, a los testores típicos en el sentido clásico de Zhuravlev, es decir, aquellos surgidos de problemas del Reconocimiento de Patrones donde las clases son conjuntos "duros" y disjuntos, los criterios de comparación por rasgos son booleanos y la función de semejanza entre objetos es la igualdad simple.

4.2.1 Conceptos básicos

Sea $\zeta = \{O_1, O_2, ..., O_m\}$ un conjunto de m objetos e $I(O_1)$, $I(O_2)$, ..., $I(O_m)$ sus descripciones en términos de un conjunto de n rasgos o características $\Re = \{X_1, X_2, ..., X_n\}$, donde cada rasgo X_j tiene asociado un conjunto M_j que se denomina *conjunto de valores admisibles del rasgo* X_j. De este modo, las descripciones de los objetos se expresan como $I(O_i) = (X_1(O_i), X_2(O_i), ..., X_n(O_i))$, $X_j(O_i) \in M_j$, $j = 1,...,n$, $i = 1,...,m$. El tipo de un rasgo depende de la naturaleza de su conjunto de valores admisibles.

Definición 4.1. *Se llama criterio de comparación de valores de la variable* X_j *a una función* C_j*:* M_j x $M_j \rightarrow L_j$*, tal que si* C_j *es un criterio de disimilaridad se cumple que* $C_j(X_j(O), X_j(O)) = \min_{y \in L_j} \{y\}$ *y si* C_j *es un criterio de similaridad se cumple que* $C_j(X_j(O), X_j(O)) = \max_{y \in L_j} \{y\}$*, donde* L_j*, con* $j = 1,...,n$ *es un conjunto totalmente ordenado,* [Shul95].

Ejemplo 4.1. Veamos dos ejemplos de criterios de comparación por rasgos (de disimilaridad) booleanos, es decir, donde el conjunto $L_j = \{0,1\}$:

1. Criterio de comparación de igualdad estricta:

$$C_k(X_k(O_i), X_k(O_j)) = \begin{cases} 0 & si\ X_k(O_i) = X_k(O_j) \\ 1 & en\ otro\ caso \end{cases}$$

2. Criterio de comparación de error admisible:

$$C_k(X_k(O_i), X_k(O_j)) = \begin{cases} 0 & si\ \left|X_k(O_i) - X_k(O_j)\right| \le \varepsilon \\ 1 & en\ otro\ caso \end{cases}$$

El primero de los criterios es aplicable a rasgos definidos en cualquier dominio mientras que el segundo sólo es aplicable a rasgos numéricos. En este último, ε es un valor real positivo.

Definición 4.2. *Sean c_1, ..., c_r un cubrimiento finito de subconjuntos propios de ζ. Se llama r-uplo de pertenencia de un objeto O a las clases $c_1,...,c_r$, denotado por $\overline{\alpha}(O)$, al tuplo $(\overline{\alpha_1}(O),...,\overline{\alpha_r}(O))$, donde $\overline{\alpha_k}(O)=\begin{cases}1 & si\ O\in c_k\\ 0 & si\ O\notin c_k\end{cases}$. Se le llama Matriz de Aprendizaje (MA) al tuplo $(I(O_1),\overline{\alpha}(O_1),I(O_2),\overline{\alpha}(O_2),...,I(O_m),\overline{\alpha}(O_m))$ que contiene las descripciones de los objetos conjuntamente con sus r-uplos de pertenencia* [Shul95].

Ejemplo 4.2. Consideremos la siguiente información acerca de los rasgos en términos de los cuales se describen a los objetos:

	Tipo	Dominio de definición	Criterio de comparación	Datos adicionales
X_1	Continuo	(-5.5,5.5)	error admisible	ε = 0.5
X_2	Discreto	{10,11,…,20}	error admisible	ε = 2
X_3	Booleano	{True, False}	igualdad estricta	
X_4	*k*-valente	{azul,blanco,rojo,verde}	igualdad estricta	

Una MA con objetos pertenecientes a 3 clases es la siguiente:

	X_1	X_2	X_3	X_4	$\overline{\alpha}$		
O_1	1.3	10	True	azul	1	0	0
O_2	-1.3	13	True	rojo	1	0	0
O_3	2.4	16	False	azul	0	1	0
O_4	-2.4	19	False	blanco	0	1	0
O_5	2.6	10	True	verde	0	0	1
O_6	-2.6	20	False	rojo	0	0	1

Definición 4.3. *Dada una MA y dados los criterios de comparación booleanos de cada uno de los rasgos, $C_1,...,C_n$, llamaremos Matriz de Diferencias de MA, y la denotaremos por MD, a la matriz booleana formada por las m' filas: $S_{ij}=(\delta_1^{ij},...,\delta_n^{ij})$, $i,j=1,...,m$, $i\neq j$, donde $\delta_p^{ij}=C_p(X_p(O_i),X_p(O_j))$, $p=1,...,n$ y $m'=\sum_{i=1}^{r-1}\sum_{t=i+1}^{r}|c_i||c_t|$* [Shul95].

Es decir, la matriz de diferencias es una matriz booleana que se obtiene como resultado de comparar todos los pares de objetos de clases distintas.

Ejemplo 4.3. La matriz de diferencia correspondiente a la MA del ejemplo 4.2 es la siguiente:

MD:	***Filas***	***Objetos comparados***
	1110	O_1 con O_3
	1111	O_1 con O_4
	1001	O_1 con O_5
	1111	O_1 con O_6
	1111	O_2 con O_3
	1111	O_2 con O_4
	1101	O_2 con O_5

1110	O_2 con O_6
0111	O_3 con O_5
1101	O_3 con O_6
1111	O_4 con O_5
0001	O_4 con O_6

Definición 4.4. *Sean i_p, i_q filas de la Matriz de Diferencias. Diremos que i_p es subfila de i_q si* $\forall j\,[a_{i_q j} = 0 \Rightarrow a_{i_p j} = 0]$ *y además* $\exists j_0\,[a_{i_q j_0} = 1 \wedge a_{i_p j_0} = 0]$. *También diremos que i_q es superfila de i_p.* [Shul95a].

Definición 4.5. *Sea i_q una fila de la Matriz de Diferencias MD. La fila i_q es básica si no existe fila alguna i_p que sea subfila de i_q. Llamaremos Matriz Básica (MB) a la matriz formada exclusivamente por las filas básicas de MD* [Shul95].

Ejemplo 4.4. La matriz básica correspondiente a la MD del ejemplo 4.3 se muestra a continuación:

MB:
1110
0001

Definición 4.6. *El subconjunto $\tau = \{X_{i_1}, ..., X_{i_s}\}$ de rasgos de una MA es un testor si al eliminar de su MB todas las columnas, excepto las correspondientes a los elementos de τ, no existe fila alguna completa de ceros. τ constituye un testor típico (TT) si al quitarle cualquiera de sus rasgos deja de ser testor* [Shul95].

Dicho de otro modo, siendo A la matriz formada sólo por las columnas correspondientes a los elementos de τ en la MB, τ es un testor típico si al eliminar cualquier columna de A aparece al menos una fila de ceros.

Ejemplo 4.5. Dada la matriz básica del ejemplo 4.4, son testores típicos los subconjuntos de rasgos $\{X_1, X_4\}$, $\{X_2, X_4\}$ y $\{X_3, X_4\}$. Sólo estos tres conjuntos cumplen esa propiedad para la matriz básica dada.

Note que cualquier conjunto que contenga a un testor típico seguirá siendo testor, pero no típico.

Definición 4.7. *Llamaremos longitud del testor τ al cardinal de τ.*

4.2.2 Algoritmos existentes para el cálculo de los testores típicos

Para el cálculo de los testores típicos se han desarrollado varios algoritmos que por su estrategia de cómputo pueden clasificarse en:

- Algoritmos de escala exterior.
- Algoritmos de escala interior.

Los algoritmos de escala exterior son aquellos que realizan el cálculo de los testores típicos generando los elementos del conjunto potencia del conjunto de columnas de la MB en un determinado orden (usualmente utilizando un *n*-uplo booleano característico

[Shul95]), de tal forma que, usando determinados criterios, el algoritmo trata de evitar el análisis de todos los subconjuntos.

Por el contrario, los algoritmos de escala interior realizan el cálculo de los testores típicos basados en el estudio de la estructura interna de la matriz, encontrando condiciones que garantizan la característica de testor y la tipicidad en las columnas asociadas a las posiciones unitarias de la matriz.

Ejemplos de algoritmos de escala interior son *CC* [Agui84] y *CT* [Brav83] y de escala exterior, *BT* [Shul82], *TB* [Shul82], *REC* [Shul95a] y *CER* [Ayaq97].

Algoritmo *BT*

El algoritmo *BT* es de escala exterior [Shul82]. El orden que caracteriza a este algoritmo es el generado por los números naturales de forma ascendente en su notación binaria mediante un *n*-uplo booleano. Éste constituye un orden total sobre los elementos no nulos del conjunto potencia del conjunto de rasgos de la MA.

El *BT* comienza entonces por el *n*-uplo (0...01) y procede comprobando si cada vector (*n*-uplo) generado es un testor o un testor típico. Según el resultado establece "saltos" en el conjunto potencia. El algoritmo termina cuando llega al *n*-uplo (1...11). Para ello, se apoya en caracterizaciones de los conceptos de testor y testor típico y en proposiciones que definen los saltos cuando el *n*-uplo analizado es o no un testor [Shul95].

En [Sanc97] se le hicieron modificaciones al algoritmo para mejorar su comportamiento. Estas modificaciones consisten en realizar un ordenamiento previo de la MB de modo que quede lo más parecido posible a una matriz triangular inferior para provocar la mayor cantidad de saltos posibles en el algoritmo y, por tanto, aumentar su eficiencia. Con este reordenamiento se logran dos condiciones importantes: la primera fila garantiza el mayor salto posible al comienzo del algoritmo y la primera fila que impide que se satisfaga la condición de testor provoca el mayor salto entre todas las que cumplen esta condición.

A nuestro juicio, el *BT*, incluso con la modificación, presenta dos inconvenientes. El primero consiste en que genera muchos *n*-uplos que son supraconjuntos de testores típicos. Esto es inherente al orden implantado y, por tanto, los saltos que tiene definido sólo pueden evitar los supraconjuntos consecutivos al que acaba de detectar como testor y muchos de los supraconjuntos restantes serán examinados innecesariamente. El segundo de los inconvenientes radica en que para comprobar si un nuevo *n*-uplo es o no testor siempre es preciso analizar las filas de la MB desde el principio, como si nunca antes se hubiese hecho.

Algoritmo *CT*

El algoritmo *CT* [Brav83] es de escala interior y trabaja basado en los conceptos de *conjunto de filas independientes* y *conjunto completo*. Inicialmente se encuentran los testores y luego, se verifica si son típicos o no. Por definición, todo conjunto completo es un testor. El método del *CT* consiste en explorar las posiciones unitarias de la MB encontrando los conjuntos completos y comprobando, entonces, si son testores típicos.

Para mejorar el funcionamiento del algoritmo, se le aplicó un ordenamiento previo a la MB [Sanc97]. Este cambio, en esencia, consiste en colocar como primera fila de la MB aquella que tenga la menor cantidad de valores unitarios entre todas las filas de MB y si existe más de una fila que cumpla esta propiedad, escoger aquella que tenga la

mayor entropía global; o sea, aquella en la que la suma de los unos presentes en las columnas donde esta fila tiene valores unitarios sea mayor.

El *CT*, como los demás algoritmos de escala interior, presenta la deficiencia de que repiten los testores típicos encontrados. Este problema, a nuestro juicio, es inherente a su estrategia. Para que un conjunto τ sea un testor típico debe existir para cada rasgo de τ al menos una fila que tenga un 1 en esa columna y cero en todas las restantes de τ, pero esta fila no tiene por qué ser única; de hecho, es más probable que no lo sea. Esto se traduce para el *CT* en que un mismo testor típico se origina a partir de más de un par conjunto completo-conjunto de filas independientes.

Algoritmo *CC*

El algoritmo *CC* [Agui84] es de escala interior y está basado en los conceptos de *conjunto compatible* y *conjunto regular*. Dos elementos unitarios de MB $a_{i_1 j_1}$ y $a_{i_2 j_2}$ son compatibles si $a_{i_1 j_2} = a_{i_2 j_1} = 0$. El conjunto $D = \{ a_{i_1 j_1}, ..., a_{i_s j_s} \}$ de valores unitarios de MB es compatible si $s = 1$ ó sus elementos son compatibles dos a dos. Un conjunto regular es un conjunto compatible, en el cual las columnas de MB asociadas a él constituyen un testor.

El *CC* tiene como peculiaridad, a diferencia del *CT*, que su objetivo es garantizar la tipicidad. Esto está implícito en la definición de conjunto compatible, luego tiene que comprobar si se cumple la propiedad de testor.

El *CC* presenta la misma limitación señalada anteriormente al algoritmo *CT*, pues en este caso, un mismo testor típico se origina a partir de más de un conjunto compatible. Además el algoritmo tiene otra deficiencia: para obtener un testor típico es necesario explorar un gran número de combinaciones de las posiciones unitarias de la MB, es decir, de conjuntos compatibles.

Algoritmos *REC* y *CER*

Ambos son algoritmos de escala exterior. El *REC* [Shul95a] tiene la particularidad de que trabaja directamente con la matriz de aprendizaje, lo cual lo sitúa en desventaja con relación a los restantes algoritmos por el enorme manejo de información que tiene que realizar. El *CER* [Ayaq97] fue propuesto para resolver este problema y lo hizo introduciendo un nuevo orden en el conjunto potencia de los rasgos. Más adelante mostraremos los ordenamientos de los algoritmos de escala exterior tratados en este epígrafe, ya que lo consideramos un elemento primordial en el desempeño de cada algoritmo.

A pesar de las modificaciones que se le han realizado a los algoritmos existentes para el cálculo de los testores típicos, aún persiste el problema del incremento exponencial del tiempo necesario para su cálculo a medida que aumentan las dimensiones de las matrices de aprendizaje. En este campo se han desarrollado otras metodologías tales como la paralelización de algunos de estos algoritmos [Sanc97], una herramienta basada en algoritmos genéticos para calcular no todos los testores típicos, sino sólo los de costo mínimo [Sanc99], entre otras.

El algoritmo que a continuación mostramos, y al que llamamos *LEX*, constituye un intento de disminuir el tiempo necesario para la obtención de todos los testores típicos de una matriz de aprendizaje.

4.2.3 Propiedades del algoritmo *LEX*

El algoritmo *LEX* puede clasificarse como de escala exterior, debido a que la principal peculiaridad de este tipo de algoritmos es que buscan los testores típicos implantando un orden sobre el conjunto potencia de los rasgos y definen ciertos saltos con el objetivo de obviar algunos conjuntos. Veamos a continuación el orden definido por este nuevo algoritmo.

Definamos la notación $[X|Y]$ para representar una lista ordenada de rasgos, de forma que X es el primer rasgo de la lista e Y es la lista ordenada de los restantes elementos. Note que es relevante el orden de los rasgos dentro de la lista. Así, por ejemplo, si tenemos la lista $[X_2,X_5,X_6]$ según esta notación podemos expresarlo como $[X_2|[X_5, X_6]]$ y, a su vez, $[X_5,X_6]$ como $[X_5|[X_6]]$. [] es la lista vacía.

Definición 4.8. *El símbolo* « *denota una relación de orden sobre el conjunto potencia de* $\Re$ ($P(\Re)$), *tal que:*

1. *Sean* $Y = [X_i|Y']$ *y* $Z = [X_j|Z']$, *entonces* Y « Z *si* $i < j$.
2. *Sean* $Y = [X_i|Y']$ *y* $Z = [X_i|Z']$ *entonces* Y « Z *si* Y'« Z'.
3. *Para toda* Y *se cumple que* [] « Y.

A este orden le debe el algoritmo el nombre de *LEX*, pues es similar al orden lexicográfico en que son comparadas las cadenas de caracteres.

En la tabla 4.1 se muestra el ordenamiento sobre $P(\Re)$ de los distintos algoritmos de escala exterior. Para simplificar se han representado los rasgos por sus índices en $\Re$ y su representación binaria. Note las diferencias existentes entre cada uno de esos ordenamientos.

REC		***CER***		***BT***		***LEX***	
$X \subseteq \Re$	Bin.	$X \subseteq \Re$	Bin.	$X \subseteq \Re$	Bin.	$X \subseteq \Re$	Bin.
1,2,3	111	∅	000	∅	000	∅	000
1,2	110	1	100	3	001	1	100
1,3	101	2	010	2	010	1,2	110
1	100	3	001	2,3	011	1,2,3	111
2,3	011	1,2	110	1	100	1,3	101
2	010	1,3	101	1,3	101	2	010
3	001	2,3	011	1,2	110	2,3	011
∅	000	1,2,3	111	1,2,3	111	3	001

Tabla 4.1.- Ordenamientos de los algoritmos de escala exterior.

En lo adelante, cuando hablemos de lista nos estaremos refiriendo a una lista ordenada de rasgos.

Con el operador + definimos la concatenación de listas. Así: $[X_1, X_2]+[X_3, X_4] = [X_1, X_2, X_3, X_4]$.

Definición 4.9. *Sea una lista de rasgos* $l = [X_{j_1}, ..., X_{j_s}]$. *Denominaremos conjunto de filas típicas de* X_t *con respecto a* l *al conjunto de los índices de las filas de MB que tienen valores unitarios en la columna correspondiente a* X_t *y ceros en las correspondientes a* X_{j_k}, $1 \le k \le s$, $X_{j_k} \ne X_t$ *y lo denotaremos por* $F^{l}_{X_t}$. *Note que* X_t *no necesariamente debe pertenecer a* l.

Ejemplo 4.6. Dada la MB siguiente:

001100
100000
000101
000010
010001

y la lista l=[X_1, X_3] entonces $F^l_{X_6} = \{3,5\}$

Denotaremos por cr_i^l a la cantidad de rasgos de l que tienen valores unitarios en la i-ésima fila de MB.

Proposición 4.1. *Sean* $l = [X_{j_1}, ..., X_{j_s}]$ *y un rasgo cualquiera* $X_t \notin l$. *Sea U el conjunto de las filas de MB donde el rasgo* X_t *tiene valor unitario, esto es,* $\{i / MB_{it}=1\}$. *Si existe un rasgo* $X_{j_k} \in l$ *tal que* $F^l_{X_{jk}} \subset U$ *entonces* X_t *no formará parte de ningún testor típico junto con todos los rasgos de l.*

Demostración. Supongamos que $\tau = l+[X_t]+f$ es un testor típico, donde f es una lista ordenada de rasgos que no tiene elementos en común con l ni con $[X_t]$. Entonces, por definición de testor típico, si eliminamos, por ejemplo, el rasgo $X_{j_k} \in l$, $1 \le k \le s$ de τ, debe aparecer una fila p de MB completa de ceros en la matriz formada por todas las columnas correspondientes a los rasgos de τ, exceptuando a X_{j_k}. Por tanto, $p \in F^l_{X_{jk}}$ y $p \in U$. Pero esto se contradice con que $MB_{pt} = 1$. Por tanto, τ no puede ser un testor típico. ∎

Proposición 4.2. *Sean* $l = [X_{j_1}, ..., X_{j_s}]$ *una lista no vacía de rasgos de MB y un rasgo cualquiera* $X_t \notin l$. *Si* $\left|F^l_{X_t}\right| = 0$, *entonces* X_t *no formará parte de ningún testor típico junto con todos los rasgos de l.*

Demostración. Supongamos que $\tau = l+[X_t]+f$ es un testor típico, donde f es una lista ordenada de rasgos que no tiene elementos en común con l ni con $[X_t]$. Entonces, por definición de testor típico, debe existir al menos una fila i de MB que tenga un uno en la columna correspondiente a X_t y cero en todas las columnas correspondientes a los rasgos de $l+f$. Por tanto, por definición, i es una fila típica de X_t con respecto a $l+f$ y, en particular, i es una fila típica de X_t con respecto a l. Esto contradice el hecho de que $\left|F^l_{X_t}\right| = 0$. ∎

Si se cumple al menos una de las dos proposiciones anteriores diremos que X_t es *excluyente con l*.

Ejemplo 4.7. Dada la MB y la lista del ejemplo anterior tenemos que X_4 es excluyente con dicha lista, pues cumple la proposición 4.1 si tomamos el rasgo X_3. Note que, en este caso, no se cumple la proposición 4.2.

Definición 4.10. *Sea* $l = [X_{j_1}, ..., X_{j_s}]$ *una lista de rasgos. Denominaremos hueco de l al rasgo* $X_p \in l$, $j_1 \le p \le j_s$, *tal que* $p = \max\{j_q / X_{j_q}, X_{j_{q+1}} \in l \wedge j_{q+1} \neq j_q + 1\}$, *es decir, es el mayor índice de los elementos de l tal que el índice de su consecutivo en l no es su consecutivo en MB.*

Proposición 4.3. *Sea* $l = [X_{j_1}, ..., X_{j_s}]$ *una lista asociada a un testor típico tal que* $X_{j_s}=X_n$. *Si existe* X_p *hueco de* l, *sea* $l'= [X_{j_1}, ..., X_{j_k}, X_{p+1}]$, *donde* $j_1 \leq ... \leq j_k = p\text{-}1 < p < j_s$. *Entonces no existe ninguna lista* λ *tal que* $l \ll \lambda \ll l'$ *y* λ *esté asociada a un testor típico.*

Demostración. Denotemos l como $\eta + \delta$, donde $\delta = [X_{j_q}, X_{j_{q+1}}, ..., X_{j_s}]$ es el sufijo de l, $j_q = p+1$ y $\eta = [X_{j_1}, ..., X_{j_k}, X_p]$ es el prefijo de l. Entonces, por el orden definido, toda lista $l \ll \lambda \ll l'$ podemos denotarla como $\lambda = \eta + \nu$, donde ν es el sufijo de λ.

Como todos los rasgos de δ son consecutivos en MB y $X_{j_s} = X_n$, entonces todo rasgo de ν está en el sufijo de l. Esto implica que cualquiera sea ν, el conjunto de rasgos representado por ν es subconjunto del conjunto de rasgos representado por δ. Por tanto, todo subconjunto de rasgos representado por λ será subconjunto del subconjunto de rasgos τ representado por l y como τ es un testor típico ninguna lista λ podrá estar asociada a un testor típico. ∎

Corolario 1. *Sea* $l = [X_{j_1}, ..., X_{j_s}]$ *una lista asociada a un testor típico tal que* $X_{j_s}=X_n$. *Si no existe hueco de l, entonces no existe ninguna lista* λ *tal que* $l \ll \lambda$ *y* λ *esté asociada a un testor típico.*

Corolario 2. *Sea* $l + [X]$ *una lista asociada a un no testor tal que* $X=X_n$ *y* $l= [X_{j_1}, ..., X_{j_s}]$. *Si existe* X_p *hueco de l, sea* $l'=[X_{j_1}, ..., X_{j_k}, X_{p+1}]$, *donde* $j_1 \leq ... \leq j_k = p - 1 < p < j_s$. *Entonces no existe ninguna lista* λ *tal que* $l \ll \lambda \ll l'$ *y* λ *esté asociada a un testor.*

4.2.4 Algoritmo *LEX*

El algoritmo *LEX* va generando listas de rasgos siguiendo el orden explicado en el epígrafe anterior. La idea del algoritmo es ir construyendo listas de rasgos que posean la propiedad de tipicidad y luego, comprobar si este conjunto de rasgos constituye un testor. Cuando se obtiene un testor típico se producen saltos.

El algoritmo comienza analizando el primer rasgo de MB. Un nuevo rasgo se puede incorporar a la lista si se cumple que no es excluyente con la lista (puede coexistir con los rasgos de la lista para formar un testor típico y tiene filas típicas con respecto a la lista). Esta condición es la que garantiza la propiedad de tipicidad.

Luego, se comprueba si el conjunto de rasgos contenido en la lista junto con el nuevo rasgo forma un testor, verificando si no existe fila en la MB (considerando sólo esos rasgos) completa de ceros. Si se cumple que es testor, entonces estamos en presencia de un testor típico y se almacena.

Si se encontró un testor típico y éste contiene al último rasgo de MB, entonces se saltan todos sus subconjuntos consecutivos. Si no contiene al último rasgo de MB, para saltar todos los supraconjuntos del testor típico encontrado se elimina el último rasgo de la lista y se analiza si se puede incluir el próximo rasgo de MB.

Cuando un rasgo es aceptado como nueva componente de la lista actual l se actualizan adecuadamente los cr_i^l para todas las filas de MB y los conjuntos de filas típicas de todos los rasgos de l.

Si el rasgo analizado no se puede incorporar a la lista, se prosigue el análisis con el siguiente rasgo de la matriz básica.

Entrada: Matriz básica MB.

Salida: El conjunto de todos los testores típicos de MB.

Algoritmo

Paso 1. Ordenación de MB.

a) Encontrar la fila con cantidad mínima de unos; de existir más de una, escoger cualquiera de ellas. Ponerla como primera fila en MB.

b) Ordenar las columnas poniendo indistintamente como primeras las que tengan un 1 en la primera fila.

Paso 2. Inicialización.

a) l = []

b) $X = X_1$ (X es el primer candidato a elemento de l)

Paso 3. Evaluación del candidato X.

a) Si l = [] y la columna correspondiente a X tiene cero en la primera fila de MB entonces FIN.

b) Si X es excluyente con l entonces ir al paso 4; no se acepta el candidato (proposición 4.1 y 4.2).

c) Si para toda fila i de MB $cr_i^{l+[X]} > 0$, entonces

   i. Guardar l + [X]; es un testor típico.

   ii. Ir al paso 4.

d) Si $X = X_n$, entonces ir al paso 4 (corolario 2 de la proposición 4.3).

e) Hacer $l = l$ + [X]; se acepta el candidato.

f) Actualizar los valores de cr_i^l para todas las filas de MB y $F_{X_t}^l$ para todos los elementos X_t en l.

Paso 4. Selección del nuevo candidato.

a) Si $X \neq X_n$, entonces

   i. Sea j el índice de X en MB.

   ii. Hacer $X = X_{j+1}$.

   iii. Ir al paso 3.

b) Si l = [] entonces FIN.

c) Si l + [X] fue un testor típico o X no fue excluyente con l, entonces

   i. Si existe el hueco X_p de l+[X], entonces

      a. Hacer $X = X_{p+1}$.

      b. Eliminar de l todos los elementos desde X_p hasta X_{j_s} (proposición 4.3), actualizando cr_i^l para todas las filas de MB y $F_{X_t}^l$ para cada X_t de l.

      c. Ir al paso 3.

   ii. De no existir X_p entonces FIN (corolario 1 de la proposición 4.3).

d) Hacer $X = X_{j_s}$, donde X_{j_s} es el último rasgo de l.

e) Hacer $l = l \setminus [X_{j_s}]$.

f) Actualizar cr_i^l para todas las filas de MB y $F_{X_t}^l$ para cada X_t de l.

g) Ir al paso 4.

Algoritmo 4.1.- Algoritmo LEX.

Con el objetivo de lograr un mejor desempeño del algoritmo se realiza previamente un ordenamiento de MB que consiste en colocar como primera fila aquella que tenga la menor cantidad de unos y, en esta fila encontrada, situar todos los unos a la izquierda (lo cual implica, por supuesto, un reordenamiento de las columnas correspondientes a cada posición unitaria). Si existiera más de una fila con mínima cantidad de unos se toma cualquiera de ellas. De esta forma, el algoritmo puede terminar cuando, durante el proceso de retroceso, la lista actual esté vacía y el rasgo que le corresponda analizarse posea un cero en la primera fila, ya que ninguno de los posibles subconjuntos de rasgos que se pueden formar con los rasgos que restan, puede formar un testor típico por no cumplir la propiedad de ser testor (la primera fila de la MB es completa de ceros en estas columnas).

Los pasos del algoritmo *LEX* se describen en el algoritmo 4.1 [Sant03].

4.2.5 Comparación con otros algoritmos de testores típicos

Para comparar el desempeño de los algoritmos analizados y el propuesto se utilizaron matrices de diferencia de diferentes dimensiones generadas aleatoriamente. A pesar de que ellas no provienen de un problema práctico concreto cumplen los requerimientos según nuestro propósito.

En la tabla 4.2 se muestran algunas de las pruebas realizadas a los algoritmos, encaminadas a comparar el tiempo que consume cada uno en el cálculo de los testores típicos para matrices de diversas dimensiones. Estas corridas fueron realizadas en una Pentium 2 de 300 MHz con 128 RAM.

Como puede observarse el algoritmo de escala exterior *BT* tiene mejor desempeño que los de escala interior (*CC* y *CT*), incluso para matrices de pequeña dimensión. Aún siendo lento, el *CT* fue más rápido que el *CC* en todas las experimentaciones realizadas. El *CC* para matrices de una dimensión superior a 55x20 es muy lento y, por tanto, podemos descartarlo como posible solución a un problema práctico.

Precisamente por lo anterior, se hizo énfasis en la confrontación del algoritmo expuesto con el *BT*, no sólo por la similitud de las técnicas usadas, sino porque es precisamente el *BT* el de mejor desempeño. Es por eso que en la tabla, en la mayoría de los casos, sólo aparecen los tiempos para el *BT* y el *LEX*.

Orden de MB	*CC*	*CT*	*BT*	*LEX*	# de TT
22x15	14832 mseg.	571mseg.	30 mseg.	**10 mseg.**	165
24x16	39697 mseg.	2824 mseg.	70 mseg.	**30 mseg.**	230
29x17	3 min.	4457 mseg.	200 mseg.	**50 mseg.**	430
54x20	1h.	4 min.	2424 mseg.	**481 mseg.**	2294
59x25		2h.	47999 mseg.	**2103 mseg.**	11746
97x25			60033 mseg.	**8172 mseg.**	19973
58x30			10 min.	**30 seg.**	34900
65x35			Más de 1 h. 30 min.	**1 min.**	112219
70x40			Más de 2 h. 53 min.	**2 min.**	446986
100x60			Más de 12 h.	**2 h.**	16567230

Tabla 4.2.- Comparación de los algoritmos de testores típicos.

En la tabla 4.2 se muestra la diferencia significativa que existe en los tiempos de cálculo del *BT* y el *LEX* a medida que aumenta la dimensión de las matrices.

4.2.6 Conclusiones

En este epígrafe se propuso un nuevo algoritmo de escala exterior que tiene significativamente mejor desempeño que los algoritmos existentes. La tabla 4.2, mostrada en el epígrafe anterior, ha sido bastante explícita y sólo ha incluido parte de todas las pruebas realizadas. En todas el resultado fue el mismo: el orden de los algoritmos según su desempeño en forma descendente fue *LEX*, *BT*, *CT* y, por último, el *CC*.

LEX eliminó los dos inconvenientes del algoritmo *BT* señalados en el epígrafe 4.2.2. Precisamente por el orden de recorrido usado para el conjunto potencia de los rasgos, al encontrar un testor típico evita el análisis de todos los supraconjuntos que le suceden en el orden establecido, lo cual no se logra con el *BT* y evita también el análisis de todos los subconjuntos consecutivos en el orden establecido. Además la existencia y actualización para cada lista del conjunto de filas típicas de los rasgos de dicha lista y de los cr_i^l para todas las filas de MB es muy útil para impedir la repetición de cálculos y poder reutilizarlos en el proceso de conformación de los próximos testores típicos.

El ordenamiento previo que se realiza de la matriz básica en el *LEX* requiere de mucho menos cálculos que el que se realiza en el *BT*. Además, en el caso del *BT*, cuando la MB es compleja (en cuanto a la disposición de ceros y unos en ella) puede suceder que a pesar de hacer el reordenamiento no se logre saltar prácticamente ningún *n*-uplo. Este caso sucede si no se logra conformar una "buena" matriz triangular. Sin embargo, en el *LEX*, independientemente de la disposición de ceros y unos en la MB, la primera fila después del ordenamiento siempre garantiza la generación de la menor cantidad de listas, lo cual constituye el objetivo trazado para dicho ordenamiento.

A nuestro juicio, estos tres elementos determinan el mejor comportamiento del algoritmo expuesto.

4.3 Algoritmo para la construcción automática de resúmenes

En el epígrafe anterior presentamos un algoritmo que permite el cálculo eficiente de los testores típicos en cualquier problema del Reconocimiento de Patrones donde las clases sean disjuntas, es decir, este algoritmo trabaja con objetos descritos por variables de cualquier naturaleza ya sea cualitativas, cuantitativas, lingüísticas, etc.

En este epígrafe nos restringiremos a su uso en colecciones de documentos, con el objetivo de presentar un método para la construcción automática de resúmenes de grupos de documentos [Pons03b].

4.3.1 Conceptos básicos

Sea ζ una colección de documentos, donde cada uno de ellos está representado, utilizando el modelo vectorial, como una tupla (w_1^i , ... , w_n^i), donde $w_j^i = TF(t_j, d^i)$ es la frecuencia del término t_j en el documento d^i. Los términos representan los lemas de las palabras que ocurren en el documento, y las palabras de parada han sido eliminadas de esta tupla. Esta colección se supone particionada en un conjunto de grupos Π.

Para cada grupo $c \in \Pi$, calcularemos su representante, y lo denotaremos por $\bar{c}$, como la unión de todos los documentos del grupo, esto es:

$$\bar{c} = \left(w_1^{\bar{c}}, \ldots, w_n^{\bar{c}} \right)$$

donde $w_j^{\bar{c}}$ es la frecuencia relativa del término t_j en el vector suma de los documentos del grupo, es decir, $w_j^{\bar{c}} = \frac{\sum_{i=1}^{|c|} w_j^i}{len(\bar{c})}, j = 1, \ldots, n.$

Dado un grupo c, sea $T(c) = \{t_1,\ldots, t_{n_c}\}$ el conjunto de los términos más frecuentes en el representante $\bar{c}$, es decir, los términos t_j tales que $w_j^{\bar{c}} \geq \varepsilon$, $j = 1,\ldots, n_c$ y ε es un parámetro definido por el usuario.

Para cada grupo $c_k \in \Pi$, construiremos la matriz $MR(c_k)$, cuyas columnas son los términos de $T(c_k)$ y sus filas son los representantes de todos los grupos de Π descritos en términos de estas columnas. Note que esta matriz es diferente en cada grupo.

$$MR(c_k) = \begin{pmatrix} w_1^{\bar{c}_1} & . & . & . & w_{|T(c_k)|}^{\bar{c}_1} \\ . & & & & . \\ . & & & & . \\ . & & & & . \\ w_1^{\bar{c}_{|\Pi|}} & . & . & . & w_{|T(c_k)|}^{\bar{c}_{|\Pi|}} \end{pmatrix}$$

En cada una de las matrices $MR(c)$ calcularemos el conjunto de los testores típicos utilizando el algoritmo *LEX* explicado en el epígrafe 4.2.4. Para ello, consideraremos sólo dos clases. La primera está formada solamente por el representante del grupo c y la otra clase está formada por el resto de los representantes de todos los grupos de Π. Esto traerá como consecuencia que la matriz de diferencias MD estará formada por $|\Pi|$-1 filas.

Para la construcción de la matriz de diferencias, utilizaremos como criterio de comparación de disimilaridad el siguiente:

$$C_j(w_j^{\bar{c}}, w_j^{\bar{c}_k}) = \begin{cases} 1 & \textit{if } w_j^{\bar{c}} - w_j^{\bar{c}_k} \geq \delta \\ 0 & \textit{en otro caso} \end{cases},$$

donde $w_j^{\bar{c}}, w_j^{\bar{c}_k}$ son los pesos del término t_j correspondientes a los representantes de los grupos c y c_k respectivamente y δ es un parámetro definido por el usuario. Note que este criterio considerará diferentes a los valores (pesos) del término t_j, si éste es altamente frecuente en el representante del grupo c y muy poco frecuente en el representante del grupo c_k.

Por otra parte, sea una oración S de un documento d y T un conjunto de términos. Llamaremos *co-ocurrencia maximal de S con respecto a T*, y lo denotaremos como $mc(S,T)$, al conjunto de todos los términos de T que ocurren en S.

Sean $S_i = (t_1^i,\ldots t_{n_i}^i)$ y $S_j = (t_1^j,\ldots t_{n_j}^j)$ dos oraciones representadas como una tupla de los términos que contienen. Al igual que en los documentos, los términos representan los lemas de las palabras que ocurren en las oraciones y las palabras de parada no se consideran.

Compararemos estas dos oraciones usando una medida de semejanza basada en la función de *Jaccard* [Take94] de la siguiente forma:

$$Sem(S_i, S_j) = \begin{cases} 1 & si \ \dfrac{|S_i \cap S_j|}{|S_i| + |S_j| - |S_i \cap S_j|} \geq \beta_S \\ 0 & en\ otro\ caso \end{cases} .$$

Note que, según la función de *Jaccard*, dos oraciones se parecen en la medida que tengan más términos en común. Si esta función supera el umbral β_S, entonces las oraciones se consideran semejantes.

4.3.2 Método de cálculo de los resúmenes

En nuestra opinión, un resumen de un grupo de documentos debe incluir los términos que caracterizan a ese grupo, pero que además lo distingan del resto de los grupos, es decir, los términos que sean altamente frecuentes en el grupo y muy poco frecuentes en el resto. Por ello, para cada grupo *c* calculamos el conjunto de los testores típicos de longitud máxima de la matriz *MR*(*c*), utilizando como criterio de comparación el explicado en el epígrafe 4.3.1.

De esta forma, un resumen de un grupo *c* estará formado por el conjunto de las oraciones extraídas de los documentos de *c*, en las cuales ocurran la mayor cantidad de términos pertenecientes al conjunto de todos los testores típicos de longitud máxima de la matriz *MR*(*c*). Además, añadiremos al resumen aquellas oraciones que garanticen el cubrimiento del conjunto de testores típicos calculado.

Es conocido que un conjunto de documentos puede incluir oraciones redundantes. El conjunto de las oraciones seleccionadas para formar parte del resumen no está exento de este problema. Para eliminar la redundancia entre las oraciones, se comparan dos a dos todas ellas usando la función de semejanza entre oraciones definida en el epígrafe anterior. Si un par de oraciones es semejante, se elimina la de menor longitud.

Finalmente, con el objetivo de mejorar la coherencia y organización de los resúmenes, ordenamos las oraciones seleccionadas de acuerdo a un orden pre-establecido en el conjunto de documentos y a su posición en el texto.

El método de cálculo de los resúmenes se describe en el algoritmo 4.2 [Pons03b].

El párrafo es una unidad semántica útil para la construcción de resúmenes debido a que los escritores ven al párrafo como una unidad de tópicos y organizan sus pensamientos de acuerdo con ello. Por lo tanto, si queremos obtener resúmenes más extensos, podemos usar los párrafos en lugar de las oraciones. De esta forma, podríamos extraer los párrafos que cubren al conjunto de los testores típicos calculados.

En el capítulo 6 se mostrarán ejemplos de resúmenes obtenidos con el algoritmo propuesto en diferentes colecciones de prueba.

Entrada: Π un conjunto de grupos de una colección de documentos ζ.
ε, umbral de la frecuencia de los términos.
δ, parámetro del criterio de comparación entre rasgos.
β_S, semejanza mínima entre oraciones.

Salida: Resumen de cada grupo de documentos.

Algoritmo

Para cada grupo $c \in \Pi$:

Paso 1. Construir la matriz $MR(c)$.

Paso 2. Calcular los testores típicos de longitud máxima de la matriz $MR(c)$ usando el algoritmo *LEX*.

Paso 3. Sea T la unión de todos los testores típicos calculados.

Paso 4. Para cada documento d en el grupo c:
Para cada oración S en d:
a) Calcular la co-ocurrencia maximal $mc(S,T)$.

Paso 5. Ordenar en forma decreciente a todas las oraciones de los documentos de c en función del cardinal de $mc(S,T)$.
Sean $p_1 > p_2 > \ldots > p_s$ estos cardinales.

Paso 6. $Resumen(c) = \varnothing$

Paso 7. Añadir a $Resumen(c)$ todas las oraciones S que satisfacen $|mc(S,T)| = p_1$.

Paso 8. $W = \bigcup_{|mc(S,T)|=p_1} mc(S,T)$.

Paso 9. $i = 2$.

Paso 10. Mientras $T \setminus W \neq \varnothing$ hacer:
a) $V = \varnothing$.
b) Para cada $mc(S,T)$ de cardinal p_i:
i. Si $mc(S,T) \cap (T \setminus W) \neq \varnothing$ entonces:
a. Añadir S a $Resumen(c)$.
b. $V = V \cup mc(S,T)$.
c) $i = i + 1$.
d) $W = W \cup V$.

Paso 11. Comparar dos a dos todas las oraciones de $Resumen(c)$. Si dos oraciones son semejantes, eliminar la de menor longitud.

Paso 12. Ordenar las oraciones de $Resumen(c)$.

Algoritmo 4.2.- Algoritmo para el cálculo de los resúmenes.

4.4 CONCLUSIONES

En este capítulo hemos presentado un método efectivo para construir resúmenes a partir de una partición en grupos de una colección de documentos. El método propuesto emplea el cálculo de los testores típicos como su operación primaria y, a partir de ellos, construye el resumen de cada grupo.

A diferencia de los otros métodos propuestos en la literatura, la selección de las oraciones se basa en los términos frecuentes y discriminantes de cada grupo, los cuales pueden ser eficientemente calculados con el algoritmo *LEX* aquí presentado.

El algoritmo *LEX* tiene significativamente mejor desempeño que los algoritmos existentes para el cálculo de los testores típicos y su utilización no está restringida a colecciones de documentos, sino que puede emplearse para la selección de variables en cualquier problema del Reconocimiento de Patrones con clases disjuntas.

La mayor novedad del método presentado para la construcción de resúmenes es, precisamente, el uso de los testores típicos que, combinado con diferentes técnicas y heurísticas, produce buenos resúmenes como será mostrado en el capítulo 6. Este método es robusto, independiente del dominio y puede ser aplicado a documentos escritos en cualquier idioma.

Adicionalmente, el método propuesto puede ser utilizado para la construcción de resúmenes de un solo documento (página Web, libro, etc.). Por ejemplo, en un libro podemos considerar como grupos algunos elementos estructurales del documento (capítulos, secciones, etc.), mientras que sus miembros serían las diferentes sub-estructuras que contienen (sub-secciones, párrafos, etc.).

5 SISTEMA DE DETECCIÓN DE TÓPICOS *JERARTOP*

5.1. INTRODUCCIÓN

Como mencionamos en el capítulo 2 el objetivo de un sistema de detección es agrupar las noticias que abordan el mismo tópico. Un sistema de detección genérico está caracterizado por un modelo de representación de los documentos, una medida de semejanza para compararlos, el tratamiento de sus propiedades temporales y un algoritmo de agrupamiento no supervisado e incremental que permita agrupar los documentos.

Otro importante problema en los sistemas de detección de tópicos es el poder proporcionar resúmenes de los tópicos detectados. De esta forma, los usuarios podrían de forma eficaz captar su contenido y no tendrían que leer todos los documentos del grupo.

En este capítulo propondremos un sistema de detección de tópicos en línea, denominado *JERARTOP* que va más allá de un sistema tradicional de detección, ya que extrae o infiere de forma completamente automática e incremental el conocimiento implícito en el flujo de documentos. Este conocimiento se expresa en forma de una jerarquía de tópicos/subtópicos, es decir, el sistema permite identificar no sólo los tópicos presentes en un flujo de noticias sino también los posibles sucesos más pequeños que ellos comprenden.

A diferencia de los sistemas de detección propuestos en la literatura, esta aproximación evita la definición de qué es un tópico, ya que es el usuario el que navegando por la jerarquía creada decide en qué nivel explora el conocimiento de su interés.

El sistema de detección propuesto categoriza, además, en un conjunto de temáticas pre-definidas a cada nodo de la jerarquía creada y proporciona un resumen de los contenidos asociados en cada uno de ellos.

Para la construcción de la jerarquía de tópicos y la descripción de cada uno de ellos, el sistema propuesto utiliza los algoritmos explicados en los capítulos anteriores.

Este capítulo está estructurado como sigue. En el epígrafe 5.2 se describe de manera general la arquitectura del sistema *JERARTOP*. Los epígrafes 5.3, 5.4 y 5.5 se dedicarán al estudio del modelo de representación de los documentos que proponemos. Primeramente, se estudiará en el epígrafe 5.3 la representación de las noticias a un nivel más simple, el de términos y en el epígrafe 5.5 explicaremos la representación conceptual de las noticias. Para lograr esta representación proponemos en el epígrafe 5.4 un nuevo algoritmo de desambiguación del sentido de las palabras.

Teniendo en cuenta la representación propuesta, en el epígrafe 5.6 estudiaremos la medida de semejanza que nos permitirá comparar a las noticias. Luego, en el epígrafe 5.7 explicaremos los aspectos relacionados con los algoritmos para la construcción de la

jerarquía de tópicos y la descripción de cada uno de ellos. Finalmente, daremos las conclusiones de este capítulo.

5.2. ARQUITECTURA GLOBAL DEL SISTEMA DE DETECCIÓN *JERARTOP*

La figura 5.1 muestra la arquitectura global de nuestro sistema de detección. Dado un flujo continuo de noticias, primeramente éstas son procesadas aplicando un conjunto de técnicas de indexación (analizador morfo-sintáctico, eliminación de la palabras de parada, extracción de lemas y reconocimiento de estructuras multipalabras). Además se reconocerán las entidades temporales y espaciales contenidas en el texto de las noticias.

Una vez realizado esto, se obtiene la representación de cada noticia. En nuestro sistema, esta representación no estará limitada a los términos que ocurren en las noticias sino que tendrá en cuenta, además, sus propiedades temporales y espaciales (vea epígrafe 5.3). Si se desea trabajar en un nivel de abstracción superior, es decir, representando a las noticias a través de los conceptos que ocurren en ellas, debemos aplicar los algoritmos de desambiguación y generalización que explicaremos en los epígrafes 5.4 y 5.5.

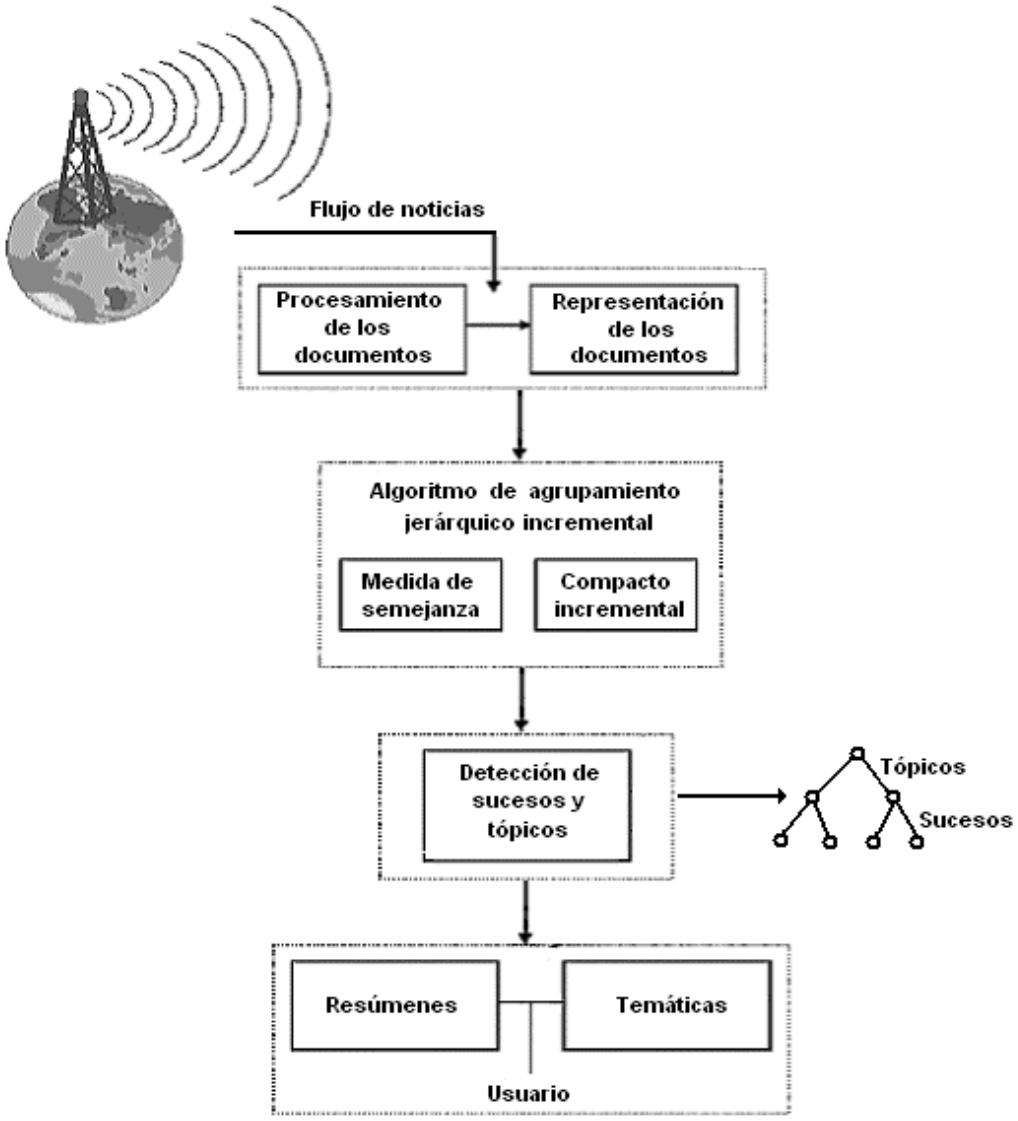


Fig. 5.1.- Arquitectura global del sistema de detección JERARTOP.

A partir de la representación obtenida definiremos una medida de semejanza global que comparará a las noticias teniendo en cuenta todas sus componentes (vea el epígrafe 5.6). El sistema de detección *JERARTOP* será capaz de obtener una jerarquía de sucesos y tópicos con diferentes niveles de granularidad, aplicando el algoritmo jerárquico incremental que utiliza como rutina de agrupamiento el algoritmo compacto incremental propuesto en el capítulo 3.

De cada tópico obtenido podemos obtener la temática que aborda o su resumen. Para la construcción automática de los resúmenes utilizaremos el algoritmo explicado en el epígrafe 4.3.

A continuación explicaremos en detalle todos los elementos que conforman la arquitectura de nuestro sistema.

5.3. MODELO DE REPRESENTACIÓN DE LOS DOCUMENTOS

Hacer la distinción entre dos desastres aéreos diferentes o dos elecciones políticas no es una tarea sencilla. Los términos de las noticias que abordan el mismo tipo de suceso tienden a converger y, por lo tanto, un vector de términos no siempre es capaz de representar la sutil diferencia que existe entre dos noticias que abordan sucesos similares pero que no son el mismo suceso.

Desde la perspectiva de un periodista, una noticia acerca de un suceso típicamente especifica las siguientes preguntas: ¿cuándo el suceso ocurrió?, ¿quiénes están involucrados?, ¿dónde ocurrió? y ¿qué ocurrió? Por tanto, nosotros pensamos que la efectividad de un sistema de detección de sucesos podría mejorarse si los documentos se representaran de tal forma que captaran estas propiedades.

Las aproximaciones actuales a la detección de sucesos no distinguen estas propiedades, considerando todo el documento como una bolsa de términos. En cambio, nosotros proponemos una representación específica para cada una de las propiedades semánticas de las noticias: las acciones y protagonistas (¿qué y quiénes?), sus propiedades temporales (¿cuándo?) y sus propiedades espaciales (¿dónde?).

Así, en nuestro sistema, cada noticia (documento) d_i se representa mediante tres tuplas:

- Una tupla de los pesos de los términos que componen el texto de la noticia.

  Los términos representan los lemas de las palabras que aparecen en el texto. Además se eliminan las palabras de parada y se reconocen algunas estructuras multipalabras como nombres de personas, organizaciones, etc. (ver epígrafe 2.3.3). Los términos se pesan estadísticamente utilizando la frecuencia *TF* normalizada por la longitud del vector (ver epígrafe 2.3.2). No utilizamos el esquema de pesado *IDF* debido a la naturaleza en línea de un sistema de detección, lo que requeriría un corpus de entrenamiento para estimar los pesos. Esta tupla se expresa de la siguiente forma:

  $$T^i = (t_1 : TF_1^i, ..., t_n : TF_n^i)$$

  donde $TF_k^i = TF(t_k, d_i)$ es la frecuencia relativa del término t_k en el documento d_i y $k = 1, ..., n$.

  Por ejemplo: (comicio: 0.006, oposición: 0.002, thabo_mbeki: 0.012, congreso_nacional_africano: 0.015, …)

- Una tupla correspondiente a las entidades temporales, las cuales representan fechas o intervalos de fechas, expresados en el calendario gregoriano. Esta tupla puede ser de dos formas:
  - Considerando solamente la fecha de publicación de la noticia, es decir,

    $F^i = (fecha_i)$

    donde $fecha_i$ es la fecha de publicación de la noticia d_i.

  - Una tupla de los pesos de las entidades temporales de la noticia.

    Estas entidades temporales son extraídas automáticamente de los textos usando el algoritmo presentado en [Llid01]. Este algoritmo, primeramente, aplica un analizador gramatical para detectar las expresiones temporales en el texto. Luego, estas expresiones se traducen a entidades temporales de un modelo formal del tiempo. Finalmente, se seleccionan aquellas entidades que representan fechas específicas o intervalos de fechas. Es bueno señalar que este algoritmo es capaz de reconocer tanto referencias absolutas (ejemplo: "10 de abril de 2003") como relativas (ejemplo: "hoy", "esta semana", etc.).

    Las entidades temporales son pesadas estadísticamente usando la frecuencia absoluta TF (sin normalizar).

    Esta tupla se expresa de la siguiente forma:

    $F^i = (f_1 : TF_{f_1^i}, ..., f_{m_i} : TF_{f_{m_i}^i})$

    donde $TF_{f_k^i}$ es la frecuencia absoluta de la entidad temporal f_k del documento d_i y $k = 1,..., m_i$.

Como podemos ver, el tratamiento de las propiedades temporales de las noticias no está restringido en nuestro sistema de detección a un ordenamiento cronológico de las mismas. Es en la propia representación de la noticia donde reflejamos estas propiedades temporales.

- Una tupla de los pesos de los lugares mencionados en el texto de la noticia.

  Estos lugares son automáticamente extraídos del texto usando un tesauro de topónimos. Los lugares se representan mediante su camino dentro del tesauro y cada nivel indica una región geográfica (país, región administrativa, etc.). Por ejemplo, "Soweto" se representa por 4P.04, donde 4P indica el país Suráfrica y 04, la región de Soweto.

  Esta tupla se expresa de la siguiente forma:

  $P^i = (p_1 : TF_{p_1^i}, ..., p_{l_i} : TF_{p_{l_i}^i})$

  donde $TF_{p_k^i}$ es la frecuencia absoluta del lugar p_k en el documento d_i y $k = 1,\ldots, l_i$.

  En este modelo no se tiene en cuenta las regiones geográficas superiores a los países, por producir una generalización demasiado fuerte en las medidas de similitud.

En la figura 5.2 vemos un ejemplo de una representación de un documento mediante estas tuplas.

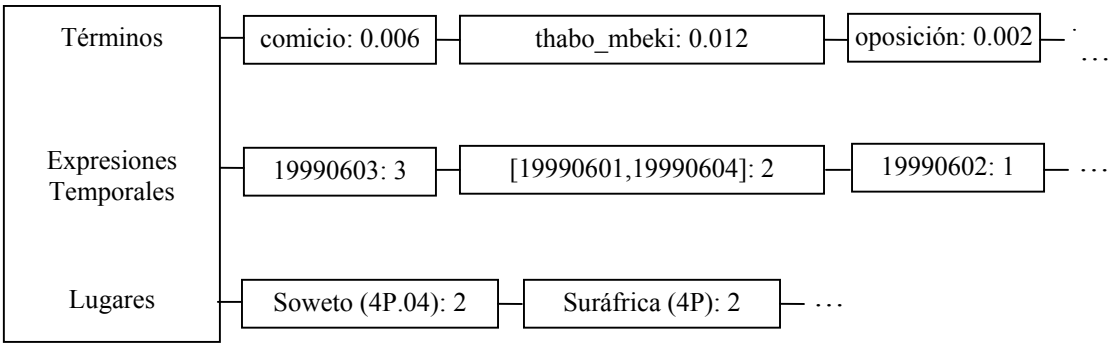


Fig. 5.2.- Ejemplo de representación a nivel de término de un documento.

A la representación del documento explicada anteriormente le llamaremos *representación a nivel de término*. En esta representación podemos reemplazar la tupla correspondiente a los términos para considerar las frases o los conceptos del documento.

Así, si en lugar de los pesos de todos los términos, consideramos solamente los pesos de los sustantivos sin modificadores, de los verbos y de las frases sustantivas del tipo sustantivo+sustantivo, adjetivo+sustantivo y sustantivo+adjetivo diremos que tenemos una *representación a nivel de frase*. Por ejemplo, si tenemos el texto "La participación de los surafricanos es el factor decisivo en estas elecciones multirraciales", su representación a nivel de término contemplaría el siguiente conjunto: {participación, surafricano, factor, decisivo, elección, multirracial}. En cambio, su representación a nivel de frase sería el conjunto: {participación, surafricano, factor_decisivo, elección_multirracial}.

Por último, a la representación del documento donde, en lugar de los pesos de todos los términos, consideramos los pesos de los conceptos que ocurren en el texto de la noticia, le llamaremos *representación conceptual*. Un término puede representar varios conceptos o acepciones. Los conceptos, en cambio, no son ambiguos. Por ejemplo: el término "estrella" representa los siguientes conceptos:

- El símbolo asterisco.
- Cuerpo sideral.
- Protagonista de una obra de teatro, película, etc.
- Figura plana de 5 o más puntas, usado como emblema.
- Cerveza estrella.
- As, ducho, genio, diestro.
- Romper, destrozar, quebrar (como verbo)

Para obtener la representación conceptual de las noticias es preciso, primeramente desambiguar los términos que ocurren en ella. Por eso, en el epígrafe siguiente proponemos un algoritmo con este fin.

5.4. ALGORITMO DE DESAMBIGUACIÓN DEL SENTIDO DE LAS PALABRAS

5.4.1 Revisión de la tecnología actual

La desambiguación del sentido de las palabras (WSD, siglas en inglés de *Word Sense Disambiguation*) es el problema de asignar el significado apropiado (sentido) a una palabra dada en un texto. Resolver la ambigüedad de las palabras es un problema de vital importancia dentro del Procesamiento del Lenguaje Natural (PLN). La desambiguación del sentido de las palabras constituye una "tarea intermedia" que sirve de ayuda en muchas aplicaciones del PLN como Traducción Automática, Recuperación de Información, Clasificación de Textos, Análisis del Discurso y Extracción de Información.

Existen dos tipos de WSD: la desambiguación basada en corpus (supervisados) y la desambiguación basada en el conocimiento (no supervisados) [Berg96]. Los métodos supervisados requieren corpus semánticamente etiquetados por un humano para posteriormente entrenar al sistema, además de un recurso léxico que suministra el sentido que se corresponde con la anotación realizada. Los métodos no supervisados, en cambio, no necesitan de estos ejemplos para realizar la desambiguación, sino que obtienen esta información automáticamente, basándose en recursos de conocimiento externos (diccionarios, tesauros, bases de conocimientos, etc.).

Generalmente las aproximaciones supervisadas han obtenido mejores resultados que los métodos no supervisados. Sin embargo, los métodos supervisados tienen la desventaja de su dependencia al corpus de entrenamiento, tanto por su disponibilidad e idoneidad, como por su discutible aplicación a diferentes dominios. Éstos necesitan de una colección de entrenamiento lo suficientemente grande y representativa del problema a resolver. Además los métodos supervisados sufren el problema del embotellamiento en la adquisición del conocimiento. Se estima que el etiquetado manual de un corpus extenso requiere más de 16 personas-año con dedicación exclusiva. Esta limitación se acrecienta si consideramos que este esfuerzo debe ser realizado antes de aplicar el método supervisado a un nuevo problema. En [Escu00] se hace un estudio empírico de la dependencia de los métodos supervisados al dominio del problema.

En los métodos basados en el conocimiento, uno de los recursos externos más utilizados actualmente es la base de conocimientos léxica *WordNet* [Mill90]. *WordNet* combina las características de otros recursos explotados comúnmente en trabajos de desambiguación, ya que incluye definiciones de términos en cada uno de sus sentidos, tal y como lo hace un diccionario y define conjuntos de palabras sinónimas y diferentes relaciones semánticas entre ellas, tal y como lo hace un tesauro. Además constituye un recurso de amplia cobertura y de libre distribución.

Se han desarrollado diversos métodos de desambiguación basados en *WordNet*. Uno de los primeros trabajos en esta dirección fue el de Voorhees [Voor93], el cual utilizó la relación de hiperonimia para sustantivos, definió la caperuza (*hood*, en inglés) de cada sentido de una palabra y propone un criterio basado en la frecuencia de aparición de las caperuzas en la colección de documentos y en el texto considerado. Sussna [Suss93] le asigna un peso a cada relación entre sustantivos de *WordNet* y define la distancia semántica entre un sustantivo del contexto y un sentido de la palabra a desambiguar como la suma de los pesos de las relaciones que conforman el camino más corto entre ellos. Agirre y Rigau [Agir96] presentan un método para desambiguar sustantivos, basado en la noción de densidad conceptual y se apoyan en la relación de hiperonimia de *WordNet*. Fernández-Amorós [Fern01] ha introducido varios parámetros con el

objetivo de mejorar la propuesta de Agirre. Magnini [Magn01] define un método de desambiguación basado en los dominios de los sentidos; así, en lugar de asignar un sentido a una palabra le asigna un dominio. Montoyo [Mont01] presenta un método para desambiguar sustantivos utilizando la relación de hiperonimia que está basado en las marcas de especificidad y un conjunto de heurísticas. Una marca de especificidad indica la información común que comparten dos conceptos en la relación de hiperonimia.

En el sistema de detección *JERARTOP* para obtener la representación conceptual de las noticias es necesario un algoritmo de desambiguación del sentido de las palabras. Por tanto, esto impone a este algoritmo ciertas restricciones. En primer lugar, debido a la naturaleza en-línea del sistema de detección, el algoritmo de desambiguación no debe ser costoso computacionalmente. En segundo lugar, el vocabulario no es restringido y varía en la medida que aumenta el flujo de noticias, por lo que no es posible contar con una colección de entrenamiento previamente etiquetada que pueda ayudar en el proceso de desambiguación.

En este epígrafe, presentamos un nuevo método eficiente de desambiguación basado en el conocimiento [Pons03a]. A diferencia de los métodos propuestos en la literatura, este método utiliza todas las relaciones semánticas de *WordNet*, y se aplica tanto a nombres, como a adjetivos y verbos. Adicionalmente, utiliza como heurística los dominios de Magnini [Magn00] y los tópicos IPTC[5] utilizados en el dominio periodístico para reducir la polisemia de las palabras en las noticias.

5.4.2 Conceptos básicos

El elemento básico utilizado por *WordNet* para representar conceptos como conjuntos de sinónimos se denomina *synset*. Por lo tanto, *WordNet* une en un mismo *synset* aquellas palabras (en realidad, formas canónicas de las palabras) que tienen un significado común. Una palabra puede formar parte de varios *synsets* y en cada uno de ellos representa un significado distinto.

Los *synsets* están enlazados entre sí para formar una red semántica que contiene relaciones jerárquicas como la *hiperonimia*/*hiponimia* y *meronimia*/*holonimia* y otras relaciones como la *antonimia*, *causa de*, *rol*, etc. Cada *synset* tiene asociado, además, un glosario, es decir, una definición de ese significado, tal y como lo hace un diccionario. *WordNet*, por tanto, puede considerarse como un grafo orientado donde los vértices son los *synsets* y los arcos representan las relaciones entre ellos.

Contextos de un término

Los métodos de desambiguación no supervisados, comúnmente desambiguan una palabra utilizando su contexto. El contexto de una palabra a desambiguar es el conjunto de palabras que aparecen próximas a ella en el texto del documento. Este contexto pudiera ser la oración, el párrafo, el texto completo, etc. Muchos trabajos de desambiguación contemplan sólo en el contexto a los sustantivos que co-ocurren con la palabra a desambiguar. En nuestro caso, nosotros consideramos a todas las palabras.

Definición 5.1. *Llamaremos SContexto de un término t en un contexto C, y lo denotaremos como* $SContexto_C(t)$*, al conjunto de todos los synsets correspondientes a los términos de C.*

[5] http://www.iptc.org

Definición 5.2. *Sean G = <S, R>, el grafo no orientado asociado al grafo de WordNet y s ∈ S, un synset. Llamaremos r-vecindad de s al conjunto de todos los synsets s' tal que existe una cadena de s a s' de longitud r en G.*

En la tabla 5.1 mostramos los *synsets* (con sus términos y relaciones correspondientes) que pertenecen a la 1-vecindad del *synset* 06342992 del sustantivo "hermano". Las relaciones inversas aparecen indicadas con *. Por otro lado, a partir de *WordNet* también puede comprobarse que el *synset* 05138072 correspondiente al término "pareja" pertenece a la 2-vecindad de 06342992.

Synset	**Términos**	**Relación**
05129724	familia	has_mero_member*
06124889	hermanastro	has_hyponym
06293748	cuatrillizo	has_hyponym
06294444	quintillizo	has_hyponym
06405110	trillizo	has_hyponym
06406919	gemelo, mellizo	has_hyponym
06163124	pariente	has_hyponym*

Tabla 5.1.- 1-vecindad del synset 06342992 del término "hermano".

El proceso de desambiguación propuesto en este epígrafe se basa en un sistema de votación de los *synsets* de acuerdo al número de *r*-vecinos que co-ocurren en un contexto dado. Dado que cada relación semántica de *WordNet* tiene un impacto diferente a la hora de desambiguar un término, se propone una función de pesado $w(R)$ sobre estas relaciones. Ésta se define del siguiente modo:

- Si R es la relación de *sinonimia*, entonces $w(R) = 10$.
- Si R es una de las relaciones siguientes: *near_synonym*, *xpos_near_synonym*, *relational_adj_wn15*, entonces $w(R) = 8$.
- Si R es la relación de *hiperonimia* o *meronimia*, entonces $w(R) = 5$.
- Si R es la relación de *antonimia*, entonces $w(R) = 1$.
- En caso contrario, $w(R) = 2$.

Los valores de estos pesos han sido determinados empíricamente, siguiendo la idea intuitiva de que las relaciones de sinonimia son las que tienen un mayor impacto en el proceso de desambiguación. Si en el contexto de un término aparece un sinónimo de éste en una de sus acepciones, es casi seguro que ésta sea su acepción. Del mismo modo, consideramos que los hiperónimos y merónimos de un término son más importantes que el resto de las relaciones. Finalmente, pensamos que la presencia de un antónimo en el contexto de un término es la que menos incidencia tiene en su desambiguación.

La votación que recibe un *synset s* del término *t* con respecto al contexto *C* y la *r*-vecindad de *s* se define como:

$$V_s(C, r) = \sum_{s' \in \Omega} v(s')$$

$$\Omega = SContexto_C(t) \cap r - vecindad(s)$$

$$v(s') = \prod_{j=1}^{r} w(R_j)$$

En las fórmulas anteriores R_j, j=1,…,r representan las relaciones que etiquetan las aristas que forman la cadena de longitud r entre los *synsets* s y s'.

Nótese, que dado un término y un sentido posible de ese término, en la medida que exista mayor cantidad de términos en su contexto que posean *synsets* "fuertemente" relacionados con ese sentido, este sentido recibirá mayor votación.

Utilización de dominios y tópicos IPTC

Los dominios de Magnini (*WordNet Domains*) [Magn00] constituyen una extensión de *WordNet*, donde cada *synset* está etiquetado con uno o más dominios. Existen aproximadamente 250 etiquetas de dominio, las cuales están organizadas en una jerarquía.

Una etiqueta de dominio puede agrupar términos de categorías sintácticas diferentes, puede incluir *synsets* de varias subjerarquías de *WordNet* y puede agrupar sentidos de una misma palabra en grupos homogéneos. Las etiquetas de dominio establecen relaciones semánticas entre los sentidos de las palabras que pueden ser aprovechadas en el proceso de desambiguación.

Al aplicar los dominios se reduce la polisemia de las palabras, ya que el número de dominios de una palabra es, generalmente, mucho menor que el número de sus sentidos. Magnini, en [Magn02], propone que se utilicen 43 de estos dominios con propósitos de desambiguación. Ejemplos de estos dominios son: arte, deporte, religión, economía, etc. Existen *synsets* genéricos que son muy difíciles de etiquetar con un dominio como, por ejemplo, el sentido de "hombre" como persona masculina adulta y otros *synsets* que aparecen frecuentemente en diferentes contextos como los días de la semana, colores, etc. Para ellos se ha creado una etiqueta de dominio denominada *Factotum* [Magn02].

Dado un texto T y un dominio D, calcularemos la frecuencia del dominio D en T como sigue:

$$Frec(D,T) = \frac{Cantidad\ \ de\ términos\ \ de\ T\ con\ \ al\ \ menos\ \ un\ \ synset\ \ etiquetado\ con\ \ D}{Cantidad\ \ total\ \ de\ términos}$$

Es importante aclarar que si un término tiene más de una ocurrencia en T se cuenta tantas veces como ocurrencias tenga.

El Consejo de Telecomunicaciones y Prensa Internacional (*International Press and Telecommunications Council, IPTC*) ha desarrollado el Sistema de Referencia de Temáticas para permitir que los proveedores de información accedan a una codificación universal independiente del idioma, de las temáticas abordadas por las noticias. Este sistema consiste en una taxonomía bastante amplia de tópicos IPTC que contienen todas las temáticas abordadas por las noticias. Por ejemplo, arte, cultura y espectáculo, desastres y accidentes, salud, política, disturbios, conflictos y guerras, entre otras. Cada una de estas temáticas se descompone, a su vez, en temáticas más específicas. La figura 5.3 muestra una parte de esta taxonomía.

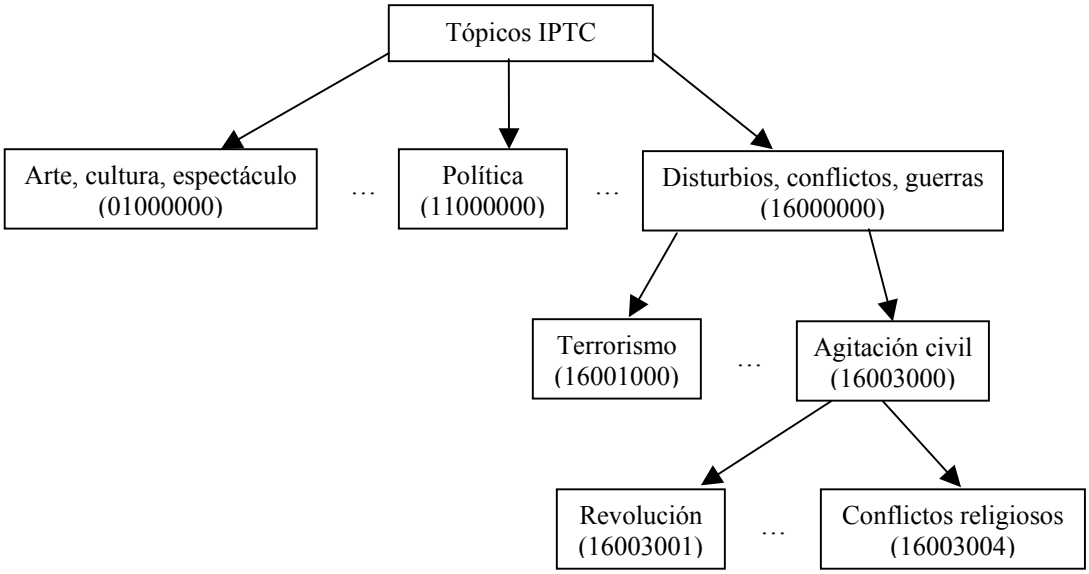


Fig. 5.3.- Parte de la taxonomía de Tópicos IPTC.

Para poder emplear este listado de temáticas (tópicos IPTC), se desambiguó a mano cada una de ellas, es decir, se le asignó *synsets* de la *WordNet*. Cada uno de estos *synsets* fue etiquetado con el tópico IPTC correspondiente, conjuntamente con todos los *synsets* relacionados directamente con él (1-vecindad) y todos sus hipónimos. De esta forma, construimos una herramienta similar a los dominios de Magnini, donde cada *synset* de *WordNet* está etiquetado con uno o más tópicos IPTC. Es preciso aclarar que no necesariamente todos los *synsets* de *WordNet* están etiquetados por un tópico IPTC.

De la misma manera que calculamos la frecuencia del dominio D en un texto T, podemos calcular la frecuencia del tópico IPTC en el texto.

Utilización de Glosarios

Aunque los diccionarios contienen múltiples inconsistencias y no se crearon para su explotación computacional, lo cierto es que proporcionan una información muy valiosa para la distinción de los sentidos de una palabra, a través de sus definiciones. Por este motivo, los diccionarios y el glosario de *WordNet* se han utilizado como recurso externo en varios métodos de desambiguación [Lesk86, Mont01].

Dado G_s, el glosario de un *synset* s asociado a un término t de un texto T, calcularemos la votación $V_s(G_s)$ que recibe s como la cantidad de términos de G_s que ocurren en T.

5.4.3 Algoritmo de desambiguación

El algoritmo de desambiguación propuesto está compuesto por dos heurísticas, y un proceso de desambiguación que se efectúa a diferentes niveles de granularidad del texto.

La primera heurística se apoya en la hipótesis de que en un texto prevalecen N_T tópicos IPTC y N_D dominios de Magnini y que ellos pueden ser utilizados para identificar el sentido de las palabras polisémicas. Por eso, lo primero que se hace es determinar los N_T tópicos IPTC y los N_D dominios de Magnini más frecuentes en el texto, y eliminar los sentidos de las palabras polisémicas que no están etiquetados con

estos tópicos o dominios (Pasos 3 y 4). Primero se realiza la poda por los tópicos IPTC por ser éstos más específicos que los dominios de Magnini.

En una segunda etapa, se comienzan a desambiguar las palabras variando el contexto y el radio de la vecindad de cada *synset* a analizar (Paso 5). De esta forma, el contexto puede ser la oración, el párrafo, el conjunto de términos desambiguados o el conjunto de términos que aún quedan por desambiguar en todo el texto. Del mismo modo, el radio de la vecindad varía de 1 a 4. Cada *synset* de la palabra a desambiguar recibirá un voto en función de los términos del contexto que están relacionados con él y de estas relaciones. En cada paso, sólo nos quedamos con los *synsets* de cada palabra que han recibido la máxima votación.

Al finalizar este proceso, se aplica la segunda heurística (*Desambigua_por_Glosario*), la cual se apoya en el glosario de cada *synset*. De esta forma, a una palabra se le asigna el *synset s* cuya $V_s(G_s)$ sea máxima.

A continuación se describe el algoritmo de desambiguación, conjuntamente con los procedimientos que utiliza [Pons03a].

Procedimiento *Desambigua_por_Contexto*(*C*: *Contexto, r*: *Radio de la vecindad*)

Para cada término *t* no desambiguado:

a) Hallar $M = \{s \in S(t) / V_s(C,r) = \max_{s' \in S(t)} \{V_{s'}(C,r)\}\}$

b) Si *M* es unitario, entonces
   i. *S*(*t*) = *M*.
   ii. Incorporar a *t* a la lista de ya desambiguados.

c) Si no,
   i. Si *r* = 1 y los *synsets* de *M* tienen un hiperónimo inmediato común *s'*, entonces
      a. S(*t*) = {*s'*}.
      b. Incorporar a *t* a la lista de ya desambiguados.
   ii. Si no, *S*(*t*) = *M*.

Algoritmo 5.1.- Procedimiento Desambigua_por_Contexto.

Procedimiento *Desambigua_por_Glosario*(*G*: *Glosario*)

Para cada término *t* no desambiguado:

a) Seleccionar el *synset s* ∈ *S*(*t*) cuya $V_s(G_s)$ sea máxima. G_s es el glosario asociado a *s*.

b) Si este *synset* existe y es único, entonces
   i. *S*(*t*) ={*s*}.
   ii. Incorporar a *t* a la lista de ya desambiguados.

c) Si no,
   i. *S*(*t*) es el conjunto de todos los *synsets* cuya $V_s(G_s)$ fue máxima.

Algoritmo 5.2.- Procedimiento Desambigua_por_Glosario.

Entrada: Un texto representado por una tupla de términos.

Cantidad de dominios de Magnini a considerar N_D.

Cantidad de tópicos IPTC a considerar N_T.

Glosario de *synsets* *G*.

Salida: *Synsets* asociados a cada uno de los términos del texto.

Algoritmo

Paso 1. Asignar a cada término *t* del texto la lista de todos sus *synsets* *S*(*t*). Si no tiene *synset* asociado en *WordNet* quedarse con el propio término.

Paso 2. Pasar a la lista de ya desambiguados a todos los términos *t* que tengan un solo *synset* en *S*(*t*).

Paso 3. Determinar los N_T tópicos IPTC y los N_D dominios de Magnini más frecuentes en el texto.

Paso 4. Para cada término *t* no desambiguado:

a) Si existe un *synset* $s \in S(t)$ etiquetado con uno de los N_T tópicos IPTC seleccionados en el paso 3, entonces

i. Eliminar de *S*(*t*) a todos aquellos *synsets* que no estén etiquetados con uno de esos N_T tópicos.

ii. Si *S*(*t*) es unitario, es decir, contiene un solo *synset*, pasar a *t* a la lista de ya desambiguados.

b) Si no, si existe un *synset* $s \in S(t)$ etiquetado con uno de los N_D dominios de Magnini seleccionados en el paso 3, entonces

i. Eliminar de *S*(*t*) a todos aquellos *synsets* que no estén etiquetados con uno de esos N_D dominios.

ii. Si *S*(*t*) es unitario, es decir, contiene un solo *synset*, pasar a *t* a la lista de ya desambiguados.

Paso 5. Para *i* en {1, 3} hacer:

a) Para *r* en {*i*, *i*+1} hacer:

i. *Desambigua_por_Contexto*(*C_Oracion*, *r*)

ii. *Desambigua_por_Contexto*(*C_Parrafo*, *r*)

b) Para *r* en {*i*, *i*+1} hacer

i. *Desambigua_por_Contexto*(*C_Desambiguados*, *r*)

ii. *Desambigua_por_Contexto*(*C_SinDesambiguar*, *r*)

Paso 6. *Desambigua_por_Glosario*(*G*)

Algoritmo 5.3.- Algoritmo de desambiguación del sentido de las palabras.

5.4.4 Experimentos

SENSEVAL es una competición con el objetivo de evaluar los sistemas de desambiguación [Kilg98]. Esta competencia propone una colección de textos de evaluación, un conjunto de definiciones y un conjunto de medidas de evaluación. En ella, los sistemas se clasifican en supervisados y no supervisados.

Para la evaluación de los sistemas en SENSEVAL se utilizan las medidas de Precisión, Relevancia (*recall*) y Cobertura, definidas de la siguiente forma:

$$Precisión = \frac{Sentidos\ desambiguados\ correctamente}{Número\ total\ de\ sentidos\ contestados}$$

$$Relevancia = \frac{Sentidos\ desambiguados\ correctamente}{Número\ total\ de\ sentidos}$$

$$Cobertura = \frac{Número\ total\ de\ sentidos\ contestados}{Número\ total\ de\ sentidos}$$

Para evaluar la efectividad del algoritmo de desambiguación propuesto utilizamos la colección de SENSEVAL-2[6] para la tarea "*lexical-sample*" en español. En esta tarea la evaluación se realiza sobre una palabra concreta de los ejemplos del texto. La colección utilizada contiene textos para desambiguar 40 palabras (18 nombres, 13 verbos y 9 adjetivos).

Para realizar la desambiguación de las palabras, primeramente, se obtienen los lemas y se eliminan las palabras de parada del texto. Como recurso externo utilizamos la base de conocimientos léxica *WordNet*, en particular, *EuroWordNet* [Voss98]. Para la aplicación de la heurística del glosario se tradujeron los glosarios de *WordNet* al español y se enriquecieron con las definiciones dadas en diccionarios y en la propia colección del SENSEVAL.

Para utilizar los dominios de Magnini fue necesario utilizar la traslación de *WordNet* 1.5 a la 1.6[7], debido a que los *synsets* de *EuroWordNet* coinciden con los de *WordNet* 1.5, mientras que los dominios se corresponden con los *synsets* de *WordNet* 1.6. En la colección de SENSEVAL-2 se determinaron los 2 dominios más frecuentes ($N_D = 2$), debido a que los textos son pequeños y el dominio más frecuente casi siempre resultó ser *Factotum*. La heurística basada en los tópicos IPTC no fue utilizada en esta colección.

Palabra	Número de ejemplos	Relevancia	Precisión	Cobertura
Arte	110	77.58	80.5	96.36
autoridad	122	34.84	36.02	96.72
bomba	113	75.66	75.66	100
canal	156	37.93	37.93	100
circuito	123	42.01	63.79	65.85
corazón	146	36.39	36.39	100
corona	119	75.28	75.28	100
gracia	160	62.86	62.86	100
grano	78	62.39	62.39	100
hermano	135	62.04	62.04	100
masa	131	46.44	69.13	67.18
naturaleza	167	34.91	39.66	88.02
operación	142	50	50	100
órgano	212	58.02	58.02	100
partido	159	83.02	83.02	100
pasaje	112	52.32	52.32	100
programa	142	42.57	42.57	100
tabla	119	74.65	78.61	94.96
TOTAL	**2446**	**55.64**	**58.48**	**95.13**

Tabla 5.2.- Resultados obtenidos con los sustantivos.

Los resultados obtenidos para los nombres se muestran en la tabla 5.2, para los adjetivos, en la tabla 5.3 y para los verbos, en la tabla 5.4. Cada una de las tablas

[6] http://www.sle.sharp.co.uk/senseval2/
[7] http://www.lsi.upc.es/~nlp/wn-map

contiene el número de ejemplos (incluyendo los datos de ensayo, entrenamiento y prueba proporcionados por SENSEVAL) y los valores de las tres medidas de evaluación de la calidad.

En la colección SENSEVAL existen sentidos que no tienen asociado *synsets* en *WordNet*, por lo que estos ejemplos se cuentan como no contestados por el algoritmo; de ahí los casos en que la cobertura no es 100 %.

Palabra	Número de ejemplos	Relevancia	Precisión	Cobertura
brillante	256	57.42	57.42	100
ciego	114	69.74	69.74	100
claro	204	57.34	57.34	100
local	139	83.74	83.74	100
natural	137	37.57	37.57	100
popular	661	53.45	57.08	93.65
simple	217	50.86	50.86	100
verde	109	58.59	68.66	85.32
vital	256	46.61	46.61	100
TOTAL	**2093**	**55.34**	**56.92**	**97.23**

Tabla 5.3.- Resultados obtenidos con los adjetivos.

Palabra	Número de ejemplos	Relevancia	Precisión	Cobertura
actuar	155	37.26	37.26	100
apoyar	210	48.62	48.62	100
apuntar	191	27.11	27.11	100
clavar	131	25.57	25.57	100
conducir	150	24.23	24.23	100
copiar	147	35.33	35.33	100
coronar	244	31.71	46.06	68.85
explotar	133	40.99	40.99	100
saltar	137	12.08	14.27	84.67
tocar	236	12.58	15.14	83.05
tratar	192	27.06	27.06	100
usar	167	48.5	48.5	100
vencer	183	31.18	31.18	100
TOTAL	**2276**	**30.82**	**32.8**	**93.98**

Tabla 5.4.- Resultados obtenidos con los verbos.

Como puede observarse en las tablas, los resultados obtenidos para los sustantivos y adjetivos son claramente superiores a los obtenidos en los verbos. Los peores resultados en los verbos los explicamos debido a la inferior cantidad de relaciones que involucran a los verbos en *EuroWordNet*. De un total de 134058 relaciones en *EuroWordNet*, en 90842 aparece al menos un sustantivo; en 33036 aparece al menos un adjetivo y en sólo 14781 aparece al menos un verbo. De estas últimas, 13427 relaciones se establecen entre dos verbos y 1354 entre un verbo y un sustantivo. Esto trae como consecuencia, que se disponga de muy poca información para poder asignar el sentido correcto a los verbos polisémicos.

5.4.5 Resumen

En este epígrafe presentamos un algoritmo de desambiguación basado en la base de conocimientos léxica *WordNet* que puede ser aplicado a cualquier dominio de

problemas sin requerir colecciones etiquetadas manualmente ni ningún proceso de entrenamiento.

El proceso de desambiguación se realiza en diferentes niveles de granularidad y utiliza todas las relaciones semánticas existentes en *WordNet*. Además, este algoritmo aprovecha los dominios de Magnini, los tópicos IPTC y los glosarios de los sentidos de las palabras para desambiguar las palabras polisémicas.

En los experimentos realizados sobre 6815 ficheros de la colección en español de la tarea "*lexical-sample*" de SENSEVAL-2 se obtuvo un 49.54% de precisión, un 47.26% de relevancia y un 95.39% de cobertura. El algoritmo propuesto obtiene resultados satisfactorios, considerando la dificultad de la tarea de desambiguación no supervisada.

En SENSEVAL-2 no se presentaron sistemas puramente no supervisados para la desambiguación de la colección en español. No obstante, la utilización de esta colección posibilita la comparación de los resultados obtenidos con otros sistemas no supervisados desarrollados.

5.5. OBTENCIÓN DE LA REPRESENTACIÓN CONCEPTUAL

Para obtener la representación conceptual de un documento nos apoyaremos en los códigos de *synsets* de la *WordNet*. Así, en lugar de la tupla de los pesos de los términos, un documento se representa por una tupla de los pesos de los conceptos (términos o *synsets*) correspondientes a los términos extraídos del texto de la noticia.

Como el algoritmo de desambiguación propuesto en el epígrafe anterior obtiene pobres resultados con los verbos, éstos no se desambiguarán y se quedarán tal cual (no se transformarán en *synsets*) en la representación conceptual.

Entrada: El texto de una noticia.

Salida: Representación conceptual del texto.

Algoritmo

Paso 1. Obtener la representación a nivel de término de la noticia.

Paso 2. Dividir la tupla T^i en dos tuplas: T_t^i formada por los pesos de los términos que son nombres propios, verbos o no tienen *synset* asociado en *WordNet* y T_s^i formada por los pesos del resto de los términos.

Paso 3. Dada la tupla de términos T_s^i obtenida en el paso anterior, aplicar el algoritmo de desambiguación explicado en el epígrafe 5.4 para obtener la tupla de *synsets* asociados a estos términos.

Paso 4. Si algún término de T_s^i no fue desambiguado, entonces pasar ese término con su peso para la tupla T_t^i. De esta forma, una noticia estaría representada por dos tuplas: T_t^i formada por los pesos de los términos (no son *synsets*) y T_s^i, por los pesos de los *synsets*.

Paso 5. Generalizar la tupla de *synsets* T_s^i, aplicando el algoritmo de generalización.

Paso 6. $T^i = T_t^i \cup T_s^i$.

Algoritmo 5.4.- Obtención de la representación conceptual.

La obtención de la representación conceptual requiere de los pasos mostrados en el algoritmo 5.4.

Es importante mencionar que cuando en el proceso de desambiguación o generalización un término se convierte en un *synset*, a este *synset* se le asocia la frecuencia del término. Del mismo modo, si más de un término se transforma en un mismo *synset*, la frecuencia de este *synset* será la suma de las frecuencias de dichos términos.

El objetivo del proceso de generalización (paso 5) es reducir la cantidad de *synsets* de la tupla obtenida en la desambiguación y lograr una mayor independencia semántica entre estos *synsets*. Además, con él tratamos de modelar la habilidad que tienen los humanos de generalizar cuando interpretan el contenido de una noticia e intentan detectar un suceso.

En este proceso se aplica la regla de generalización que a continuación definimos.

Definición 5.3. *Sean d un documento representado por una tupla de synsets S, s un synset cualquiera y H_s el conjunto de los hijos de s en una relación R de WordNet. Diremos que s generaliza a sus hijos si se cumple una de las condiciones siguientes:*

- *$s \in S$ y $\exists\ s' \in H_s$ tal que $s' \in S$, es decir, s es un synset del documento y uno de sus hijos es también un synset del documento.*
- *$s \notin S$ y la mitad más uno de sus hijos son synsets del documento o generalizan a su vez a sus hijos.*

Esta regla de generalización se aplicará a las relaciones de *hiperonimia* y *meronimia* de *WordNet* por tener estructuras semánticas jerárquicas. Un concepto representado por un *synset s* es un *hipónimo* del concepto representado por *s'* si *s es un s'*. En este caso, se dice que *s'* es un *hiperónimo* de *s*. Por ejemplo, el concepto "cubano" es un hipónimo del concepto "caribeño" y éste, a su vez, es un hipónimo de "latinoamericano".

Un concepto representado por un *synset s* es un *merónimo* del concepto representado por *s'* si *s es parte de s'*. En este caso, se dice que *s'* es un *holónimo* de *s*. Por ejemplo, el concepto "proa" es un merónimo de "barco".

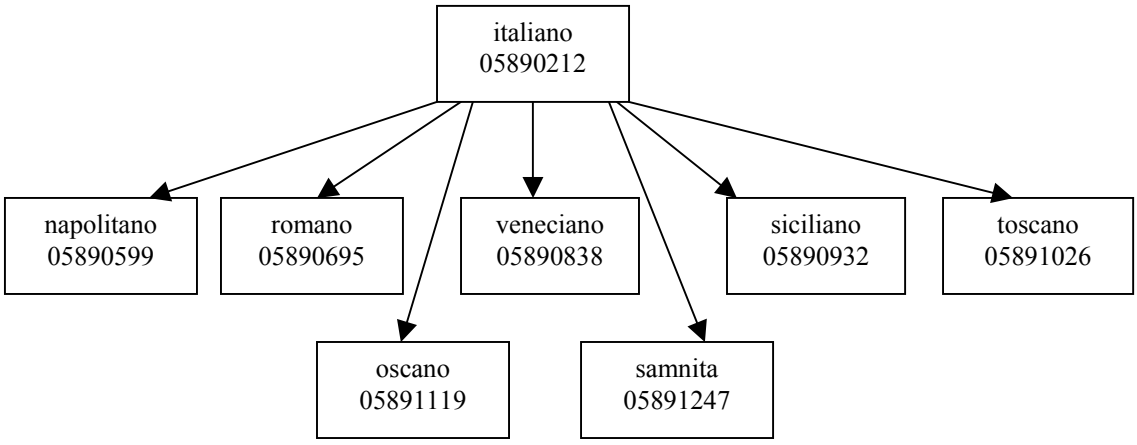


Fig. 5.4.- Ejemplo de relación de hiperonimia.

Veamos un ejemplo de aplicación de la regla de generalización. Supongamos que se tiene un documento que contiene a los términos "napolitano", "romano", "veneciano" y "siciliano". Una vez desambiguados estos términos se transformaron en los *synsets*

05890599, 05890695, 05890838 y 05890932, respectivamente. La figura 5.4 muestra las relaciones de hiperonimia que existen entre ellos.

Como puede observarse el concepto "italiano" tiene 7 hijos en la relación de hiperonimia y 4 de ellos están presentes en el documento. Por tanto, el concepto "italiano" generaliza a "napolitano", "romano", "veneciano" y "siciliano" y los *synsets* 05890599, 05890695, 05890838 y 05890932 se transforman en un solo *synset*: 05890212. La idea intuitiva de este proceso de generalización es que si un documento habla sobre napolitanos, romanos, venecianos y sicilianos podemos decir que habla sobre italianos.

En *WordNet* existen, además, algunos conceptos que están etiquetados como conceptos base. Por ejemplo, "acto", "célula" y "posición_social". Estos conceptos son muy generales, es decir, están en los primeros niveles de la jerarquía. En nuestro método de generalización usamos los conceptos base como condición de parada.

A continuación se describe el algoritmo de generalización empleado y que tiene en cuenta la definición anterior.

Entrada: Un documento representado por una tupla de *synsets*.

Salida: Un documento generalizado, representado por una tupla de *synsets*.

Algoritmo

Paso 1. Se construyen los conjuntos *RutaH* y *RutaM*.

RutaH = conjunto de todos los ancestros (que no son conceptos base) de los *synsets* del documento a través de la relación de hiperonimia.

RutaM = conjunto de todos los ancestros (que no son conceptos base) de los *synsets* del documento a través de la relación de meronimia.

Paso 2. Se determinan en *RutaH* y *RutaM* los *synsets* que generalizan a sus hijos. Todos los *synsets* que generalizan a sus hijos y no son generalizados por ningún *synset* se quedan en *RutaH* y *RutaM*. El resto es eliminado.

Paso 3. Para cada *synset s* del documento:

a) Si *s* fue generalizado por algún *synset* de *RutaH* o de *RutaM*, *s* se sustituye por su generalización. Si en ambos conjuntos *s* se generalizó, entonces *s* se reemplaza por sus dos generalizaciones.

b) Si *s* no fue generalizado por ningún *synset* de *RutaH* ni de *RutaM* nos quedamos con el propio *synset*.

Algoritmo 5.5.- Algoritmo de generalización.

5.6. MEDIDA DE SEMEJANZA

Un sistema de detección de tópicos requiere de una medida de semejanza para comparar a las noticias durante el proceso de agrupamiento. La mayoría de los algoritmos presentados en la literatura utilizan la medida del coseno. En nuestra aproximación, además de esta medida, introduciremos nuevas medidas de semejanza parcial para cada componente de la representación de los documentos. De esta forma, diremos que dos noticias abordan el mismo tópico si coinciden aproximadamente en su contenido, lugares y propiedades temporales.

Para definir una medida de semejanza global es necesario combinar de cierta forma las medidas de semejanza parcial que se definan en cada una de las componentes de la

representación de las noticias. Es usual, definir una combinación lineal de las diferentes medidas de semejanza parcial como, por ejemplo, se hace en [Makk03]. Esta forma no es conceptualmente correcta, pues cada una de las semejanzas parciales se refieren a universos diferentes, por lo que la unidad de semejanza en una no tiene por qué ser la misma en el otro universo. El hecho de que dos noticias con respecto a sus propiedades temporales se parezcan en 0.3 y con respecto a sus propiedades espaciales, en 0.5 no significa que ellas se parezcan en 0.8. Es necesario, por tanto, homogeneizar esas mediciones que provienen de universos diferentes.

La medida de semejanza global, que propondremos en este epígrafe, se construirá sobre la base de filtros. Todas las noticias que describen sucesos que ocurren en un mismo lugar conforman un universo homogéneo. Todas las noticias de este universo se pueden particionar, a su vez, en universos homogéneos de sucesos que ocurriendo en el mismo lugar, ocurren en un intervalo de tiempo preestablecido. Por último, en cada uno de los universos anteriores se pueden realizar estructuraciones de las noticias según los términos que ellas contienen. De esta forma, los universos de las propiedades espaciales, temporales y semánticas de las noticias se han sumergido en un mismo universo, donde tiene sentido hacer operaciones con la unidad de semejanza.

A continuación explicaremos con más detalle cómo se construye esta medida.

5.6.1 Semejanza semántica

A la medida de semejanza parcial que compara las tuplas correspondientes a los términos le llamaremos *semejanza semántica* y la denotaremos por S_T. Esta medida dependerá de la representación de los documentos.

Si tenemos la representación a nivel de términos o de frases de los documentos, utilizaremos la medida del coseno para comparar a los documentos d_i y d_j, es decir,

$$S_T(d_i, d_j) = cos(T^i, T^j) = \frac{\sum_{k=1}^{n} TF_k^i \cdot TF_k^j}{\sqrt{\sum_{k=1}^{n} TF_k^{i\,2}} \cdot \sqrt{\sum_{k=1}^{n} TF_k^{j\,2}}}$$

Por otra parte, si estamos trabajando con la representación conceptual de los documentos, definiremos una medida de semejanza basada en la medida del coseno pero que intenta captar las relaciones semánticas que existen entre los *synsets* de ambos documentos.

Dada una relación R de *WordNet* y dos *synsets* s_1 y s_2, la notación s_1 R s_2 significa que s_1 y s_2 están relacionados directamente por la relación R.

Sea ζ el espacio de representación de los documentos. Definamos, además, una función σ: ζ x $\zeta \rightarrow \Lambda$, tal que para dos documentos d_i y $d_j \in \zeta$:

$$\sigma(d_i, d_j) = \{(t_k, t_l) / t_k \in T^i, t_l \in T^j, t_k \succ t_l\}$$

donde $t_k \succ t_l$ si se cumple una de las condiciones siguientes:

1. $t_k = t_l$ (t_k y t_l pueden ser términos o *synsets*).
2. t_k y t_l son *synsets*, $t_k \neq t_l$ y se cumple una de las dos condiciones siguientes:

- $t_l \, R \, t_k \wedge TF_l^j = max \{TF_q^j / t_q \in T^j \wedge t_q R \, t_k \}$.
- $t_k \, R \, t_l \wedge TF_k^i = max \{TF_q^i / t_q \in T^i \wedge t_q R \, t_l \}$.

Note que tanto t_k como t_l pueden pertenecer a más de un par en $\sigma(d_i, d_j)$.

Denotaremos como $\Pi(d_i / d_j)$ a la tupla formada por:

- Todos los $t_k \in T^i$ tales que $\exists t_l \in T^j \, (t_k, t_l) \in \sigma(d_i, d_j)$, donde t_k puede estar varias veces repetido en dicha tupla si aparece en más de un par de $\sigma(d_i, d_j)$.
- Todos los $t_k \in T^i$ que no pertenecen a ningún par de $\sigma(d_i, d_j)$.

Entonces la semejanza semántica entre las representaciones conceptuales de dos documentos puede calcularse del siguiente modo:

$$S_T(d_i, d_j) = \frac{\sum_{(t_k, t_l) \in \sigma(d_i, d_j)} TF_k^i \cdot TF_l^j}{\left\| \Pi(d_i / d_j) \right\| \cdot \left\| \Pi(d_j / d_i) \right\|}$$

Esta medida trata de seguir la misma idea de la medida del coseno, pero aquí los documentos se transforman de su espacio de representación inicial en un nuevo espacio que resulta del pareo semántico entre los documentos que se comparan.

Por ejemplo, supongamos que tenemos dos documentos d_1 y d_2 representados conceptualmente por las tuplas de términos siguientes:

$$T^1 = (\text{participar} : 2, 02093370\text{a} : 3, 00380975\text{n} : 1, 00351389n : 1)$$

donde 02093370a es el *synset* correspondiente al adjetivo “electoral”, 00380975n es el *synset* asociado al sustantivo “conteo” y 00351389n es el *synset* asociado al sustantivo “presidencia”.

$$T^2 = (\text{controlar} : 1, \text{participar} : 2, 00101987\text{n} : 1, 00382029\text{n} : 2)$$

donde 00101987n es el *synset* correspondiente al sustantivo “elección” y 00382029n es el *synset* asociado al sustantivo “escrutinio”.

Entre el *synset* 02093370a y el *synset* 00101987n existe una relación de adjetivo relacional (*relational_adj*) y entre los *synsets* 00380975n y 00382029n existe una relación de hiperonimia (*has_hyponym*), las cuales se tomarán en cuenta en el cálculo de la semejanza de estos documentos.

$$\sigma(d_1, d_2) = \{(\text{participar}, \text{participar}), (02093370\text{a}, 00101987\text{n}), (00380975\text{n}, 00382029\text{n})\}$$

Por tanto, la semejanza semántica entre d_1 y d_2 se calcula como sigue:

$$S_T(d_1, d_2) = \frac{2 \cdot 2 + 3 \cdot 1 + 1 \cdot 2}{\sqrt{2^2 + 3^2 + 1^2 + 1^2} \cdot \sqrt{1^2 + 2^2 + 1^2 + 2^2}} = 0.73$$

5.6.2 Distancia temporal

Si utilizamos solamente la fecha de publicación de las noticias para caracterizarlas temporalmente, entonces la distancia temporal se define como el número de días entre sus fechas de publicación, es decir,

$$D_F(d_i, d_j) = \textit{número de días entre } \textit{fecha}_i \textit{ y } \textit{fecha}_j$$

Para comparar las tuplas correspondientes a las entidades temporales extraídas automáticamente de los documentos d_i y d_j proponemos una distancia basada en la tradicional distancia entre conjuntos, la cual se define como sigue:

$$D_F(d_i, d_j) = \min_{f^i \in FR_i, f^j \in FR_j} \{dist(f^i, f^j)\}$$

donde:

- $dist(f^i, f^j)$ es la distancia entre las entidades temporales f^i y f^j y se define de la siguiente forma:
  - Si f^i y f^j son fechas, entonces $dist(f^i, f^j)$ es el número de días entre f^i y f^j.
  - Si $f^i = [a,b]$ y $f^j = [c,d]$ son intervalos de fechas, entonces $dist(f^i, f^j) = \min_{f_1 \in [a,b], f_2 \in [c,d]} \{dist(f_1, f_2)\}$.

    Aquí nuevamente seguimos la idea intuitiva de que los intervalos de fechas son realmente conjuntos de fechas y, por tanto, podemos definir la distancia en términos de la tradicional distancia entre conjuntos.
  - Si f^i es una fecha y f^j es un intervalo, entonces $dist(f^i, f^j) = dist([f^i, f^i], f^j)$.
  - Si f^i es un intervalo y f^j es una fecha, esta función se define de forma similar al caso anterior.
- FR_i es el conjunto de todas las entidades temporales f^i del documento d_i que satisfacen las siguientes condiciones:
  - f^i tiene la máxima frecuencia en d_i, es decir, $TF_{f^i} = \max_{k=1,\ldots,m_i} \{TF_{f_k^i}\}$.
  - f^i tiene la mínima distancia *dist* a la fecha de publicación $fecha_i$ del documento d_i (esta condición es ignorada cuando se compara los representantes de los grupos, en lugar de los documentos).

  No es difícil notar que el conjunto FR_i no necesariamente es unitario.

5.6.3 Semejanza espacial

Para comparar las tuplas correspondientes a los lugares de los documentos d_i y d_j proponemos la siguiente función:

$$S_P(d_i,d_j)=\begin{cases}1 & si\ \exists\ p_q^i, p_t^j\ tal\ que\ tienen\ un\ prefijo\ común\\ 0 & en\ otro\ caso\end{cases}$$

Por ejemplo, "Kosovo" representado por 5G.02 será semejante a "Yugoslavia" representado por 5G, pues ambos tienen el prefijo común 5G.

Es bueno mencionar que la semejanza espacial propuesta sólo considera regiones geográficas inferiores o iguales a países, para que sea lo suficientemente discriminante. Por otro lado, se ha flexibilizado al máximo la comparación entre regiones (basta con que aparezca un mismo país de forma directa o indirecta en ambos documentos para que la semejanza sea 1) para evitar que dos noticias semejantes conceptualmente no se relacionen a causa de los lugares mencionados en el texto. Este aspecto es importante ya que muchos sucesos de relevancia implican varios lugares distintos.

5.6.4 Semejanza global

Teniendo en cuenta las medidas de semejanza parcial definidas anteriormente para cada una de las componentes de la representación de los documentos, definimos la medida de semejanza global como sigue:

$$sem(d_i,d_j)=\begin{cases}S_T(d_i,d_j) & si\ D_F(d_i,d_j)\le\beta_{tiempo}\wedge S_P(d_i,d_j)=1\\ 0 & en\ otro\ caso\end{cases}$$

donde β_{tiempo} es el número máximo de días requerido para determinar si dos noticias abordan o no el mismo tópico.

Esta medida trata de capturar la idea de que dos noticias abordan el mismo tópico si tienen una alta semejanza semántica, proximidad temporal y coincidencia en lugar.

Si para representar a los documentos sólo tuviéramos en cuenta la tupla de los términos la medida de semejanza global sería:

$$sem(d_i,d_j)=S_T(d_i,d_j)$$

Si, por el contrario, ignorásemos la tupla correspondiente a los lugares, entonces:

$$sem(d_i,d_j)=\begin{cases}S_T(d_i,d_j) & si\ D_F(d_i,d_j)\le\beta_{tiempo}\\ 0 & en\ otro\ caso\end{cases}$$

Finalmente, si ignorásemos la tupla correspondiente a las entidades temporales, entonces la medida de semejanza global sería:

$$sem(d_i,d_j)=\begin{cases}S_T(d_i,d_j) & si\ S_P(d_i,d_j)=1\\ 0 & en\ otro\ caso\end{cases}$$

En el próximo capítulo analizaremos la repercusión de cada una de estas medidas en la efectividad de un sistema de detección de tópicos.

5.7. SISTEMA DE DETECCIÓN *JERARTOP*

Nuestro sistema de detección de tópicos está caracterizado por los siguientes aspectos:

- Se representarán los documentos no sólo teniendo en cuenta su contenido sino también sus propiedades temporales y espaciales.
- Para comparar a las noticias se utilizará una medida de semejanza que contempla sus propiedades semánticas, temporales y espaciales.
- Se identificarán no sólo los tópicos sino también los sucesos en los que ellos se pueden descomponer, es decir, se obtendrá una jerarquía de sucesos y tópicos.
- Adicionalmente, se determinará la temática (tópico IPTC) a la que se refiere cada suceso o tópico detectado.
- De cada uno de los tópicos detectados se proporcionará un resumen que permita captar sus ideas fundamentales.

A continuación explicaremos estos aspectos con más detalle.

5.7.1 Jerarquía de tópicos

El concepto de suceso es problemático; a pesar de que parece intuitivamente muy claro, es difícil establecer una definición sólida. Inicialmente, como mencionamos en el capítulo 2, en TDT se definió un suceso como algún hecho que ocurre en un instante de tiempo y lugar específicos y luego, se amplió la noción de suceso a la de tópico para incluir además a todos los sucesos, hechos o actividades directamente relacionados con él.

Las relaciones entre sucesos y tópicos no son simples ni están bien definidas en la vida real. Algunos tópicos pueden ser más amplios que otros e incluso algunos tópicos contienen sub-tópicos, por lo que sería interesante poder captar estos niveles de detalles. Por ejemplo, la figura 5.5 presenta la estructura de sucesos del tópico "Guerra de Kosovo". En el primer nivel de la jerarquía se muestran pequeños sucesos (el número de documentos de cada suceso aparece entre paréntesis). En los niveles superiores, se muestran los sucesos compuestos y tópicos que pueden construirse a partir de ellos.

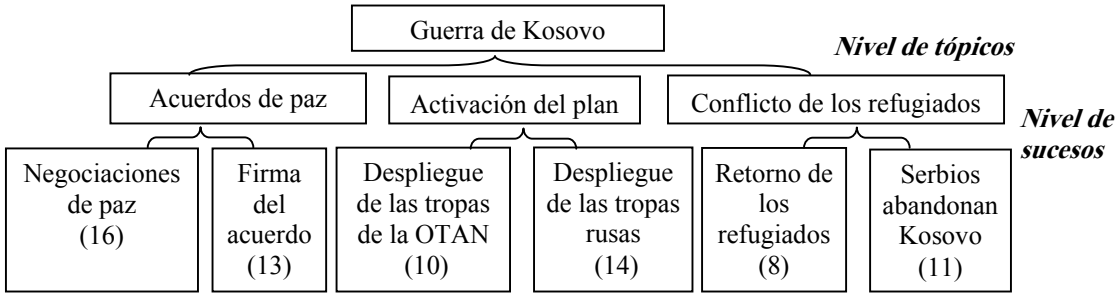


Fig. 5.5.- Estructura de sucesos del tópico "Guerra de Kosovo".

Los sistemas de detección actuales no son capaces de manipular estas estructuras de sucesos y tópicos. Nuestro sistema de detección intentará descubrir la estructura de los tópicos y sucesos reportados por una colección de noticias, es decir, no sólo obtener los tópicos sino también los posibles sucesos en los que ellos se pueden descomponer. Así, en lugar de obtener un conjunto de grupos de documentos, donde cada uno representa

un tópico detectado, nuestro sistema construirá una jerarquía de documentos. El nivel inferior de esta jerarquía representará a los sucesos más simples, conformados por las noticias de la colección. Los niveles superiores de la jerarquía contendrán sucesos más complejos y tópicos que pueden construirse a partir de los niveles inferiores.

Esta aproximación, además, evita la definición de qué es un tópico (como en los sistemas actuales), ya que es el usuario el que navegando por la jerarquía creada decide en qué nivel explora el conocimiento de su interés.

Con este objetivo, utilizaremos el algoritmo jerárquico incremental propuesto en el epígrafe 3.4. Este algoritmo recibe como entrada:

- La medida de semejanza propuesta en el epígrafe 5.6 que, como analizamos anteriormente, tiene en cuenta no sólo el contenido de las noticias sino también sus propiedades temporales y espaciales.
- Como rutina de agrupamiento utilizaremos el algoritmo compacto incremental propuesto en el epígrafe 3.2.
- El representante $\bar{c}$ de cada grupo *c* será, al igual que los documentos, una tupla de 3 elementos $(T^{\bar{c}}, F^{\bar{c}}, P^{\bar{c}})$. Como método de cálculo de los representantes de los grupos utilizaremos la unión de los documentos del grupo, es decir:
  - $T^{\bar{c}} = (t_1 : TF_1^{\bar{c}}, ..., t_n : TF_n^{\bar{c}})$, donde $TF_j^{\bar{c}}$ es la frecuencia relativa del término t_j en el vector suma de las tuplas T^i de los documentos d_i del grupo *c*; esto es, $TF_j^{\bar{c}} = \dfrac{\sum_{d_i \in c} len(d_i) \cdot TF_j^i}{len(\bar{c})}$.
  - $F^{\bar{c}} = (f_1 : TF_{f1}^{\bar{c}}, ..., f_{m_{\bar{c}}} : TF_{f_{m_{\bar{c}}}}^{\bar{c}})$, donde $TF_{f_j}^{\bar{c}}$ es la frecuencia absoluta de la entidad temporal f_j (ya sea extraída automáticamente o la fecha de publicación) en el grupo *c*, es decir, es el número de documentos en el grupo que contienen a la entidad temporal y $m_{\bar{c}}$ es el número total de entidades temporales que describen a los documentos del grupo.
  - $P^{\bar{c}} = (p_1 : TF_{p1}^{\bar{c}}, ..., p_{l_{\bar{c}}} : TF_{p_{l_{\bar{c}}}}^{\bar{c}})$, donde $TF_{p_j}^{\bar{c}}$ es la frecuencia absoluta del lugar p_j en el grupo *c*, es decir, es el número de documentos en el grupo que contienen al lugar y $l_{\bar{c}}$ es el número total de lugares que describen a los documentos del grupo.

  Para reducir la longitud de los representantes de los grupos truncamos las tuplas, eliminando los términos (entidades temporales, lugares) cuya frecuencia sea menor que la décima parte de la máxima frecuencia.

El algoritmo compacto incremental es apropiado para la detección de tópicos debido a:

- Es no supervisado, es decir, no requiere de muestras de entrenamiento etiquetadas.
- Su naturaleza incremental.

- Los tópicos (grupos) detectados por el algoritmo no dependen del orden de presentación de los documentos.
- No realiza una asignación irrevocable de los documentos a los grupos. Esto permite que pueda realizar reasignaciones de los documentos a los grupos en la medida en que los sucesos evolucionan.
- Presenta un efecto de encadenamiento restringido. Cuando un suceso evoluciona, muchas de sus propiedades son inicialmente desconocidas o se asumen conocidas por la audiencia, por lo que no necesariamente están explícitas en el texto de la noticia. Además nuevos rasgos léxicos de un suceso aparecen en la medida en que el suceso evoluciona. Por ejemplo, en el suceso acerca del desastre causado por una bomba colocada por un grupo terrorista en una ciudad, inicialmente las noticias informan sobre los detalles del desastre y los esfuerzos realizados en la recuperación del sitio del desastre y luego, cuando el suceso evoluciona, las noticias sobre el suceso contienen los detalles del grupo terrorista que se adjudica el hecho, los detalles de su captura, etc. La forma en que se construyen los grupos compactos como componentes conexas en el grafo de máxima β_0-semejanza permite de cierta forma relacionar estas noticias.

5.7.2 Obtención de las temáticas

Como mencionamos en el epígrafe 5.4.2, el Consejo de Telecomunicaciones y Prensa Internacional (*International Press and Telecommunications Council, IPTC*) ha desarrollado una taxonomía bastante amplia de tópicos IPTC que contienen todas las temáticas abordadas por las noticias. A cada uno de los sucesos y tópicos detectados en nuestra jerarquía le asignaremos un tópico IPTC. Esto sin dudas, sería muy útil para un analista de información, pues además de la estructuración obtenida de los tópicos en diferentes niveles de granularidad, tendría la temática abordada por cada uno de ellos. Por ejemplo, el tópico de la guerra de Kosovo (ver figura 5.5) pertenece a la temática 16 sobre disturbios, conflictos y guerra.

Para obtener la temática de un grupo (tópico) detectado por nuestro sistema, se obtiene primeramente la representación conceptual de su representante, es decir, calculamos el representante del grupo y lo desambiguamos. De esta forma, el representante estará caracterizado por una tupla de *synsets*.

Como mencionamos en el epígrafe 5.4.2, cada *synset* de *WordNet* fue etiquetado por tópicos IPTC y, por lo tanto, podemos determinar los 10 tópicos IPTC que más *synsets* etiquetan en el representante.

Los tópicos IPTC están organizados en una taxonomía. Cada uno de los tópicos IPTC seleccionados votarán por todos sus ancestros en la taxonomía. El voto que le otorgan es la cantidad de *synsets* del representante que ellos etiquetan. Por ejemplo, si uno de los 10 tópicos seleccionados fue el 16003001 (revolución), éste votará por 16003000 (agitación civil) y por 16000000 (disturbios, conflictos y guerras) (ver figura 5.3).

Finalmente, se selecciona el tópico IPTC que recibió mayor votación. La temática asignada al grupo será el tópico IPTC que cumpla con las condiciones siguientes:

- Es uno de los 10 tópicos IPTC seleccionados inicialmente.
- Es un descendiente del tópico IPTC de mayor votación.

- Entre todos los descendientes del tópico IPTC de mayor votación es el que etiqueta a la mayor cantidad de *synsets* en el representante.

La idea intuitiva de este método es seleccionar, primeramente el tópico IPTC más general que etiqueta a la mayor cantidad de términos en el grupo de documentos y luego, dentro de esta temática seleccionar la más particular.

Es fácil ver que pudiera ocurrir que más de un tópico IPTC cumpla con las condiciones anteriores. Si esto ocurre, el grupo es etiquetado con todos los tópicos IPTC que satisfagan estas condiciones, es decir, al grupo se le asignará más de una temática.

A continuación se describe el algoritmo de obtención de las temáticas.

Entrada: Un grupo de documentos que representa un tópico detectado.

Salida: Temática (tópico IPTC) del tópico.

Algoritmo

Paso 1. Calcular el representante del grupo por el método de la unión explicado en el epígrafe 5.7.1.

Paso 2. Desambiguar el representante aplicando el algoritmo propuesto en el epígrafe 5.4.

Paso 3. Determinar los 10 tópicos IPTC que etiquetan a más *synsets* en el representante desambiguado. Sea *L* la lista de estos 10 tópicos.

Paso 4. Cada tópico IPTC de *L* vota por todos sus ancestros.

Paso 5. Sea *topic* el tópico IPTC que recibió mayor voto.

Paso 6. Seleccionar el tópico IPTC de *L* que sea el descendiente de *topic* de mayor voto.

Algoritmo 5.6.- Algoritmo de obtención de las temáticas.

De la misma forma que hemos asignado una temática a un grupo de documentos, podemos también asignarle una temática a un documento en particular. En este caso, basta obtener la representación conceptual del documento en lugar del representante y aplicar los pasos del 3 al 6.

Adicionalmente, como mencionamos anteriormente, de cada tópico detectado podemos obtener un resumen aplicando el algoritmo propuesto en el epígrafe 4.3.

En la figura 5.6 mostramos un fragmento de una jerarquía creada por el sistema *JERARTOP* con dos niveles. Esta jerarquía se representa mediante un fichero XML que contiene para cada tópico los sucesos en los que se descompone y la temática que aborda cada uno. A su vez, cada suceso contiene los titulares de las noticias que lo conforman, además de la temática, el resumen, las fechas y los lugares más representativos del suceso. Cuando un suceso o tópico es unitario, no construimos su resumen.

En el capítulo 6 se mostrarán los resultados obtenidos por el sistema de detección propuesto en varias colecciones de noticias.

```
- <Topic Id= "Cluster_134" Level= "1">
    <Description Testores= "['negociación', 'unión_europea', 'presidente']" />
    <Subject IPTC= "Política & Tratados y Organizaciones & alliance" />
  - <Event Id= "Cluster_166" Level= "1">
      <Description Fechas= "19990604" Lugares= "MEXICO, RIO, COLONIA" Resumen=
      "Ni siquiera una conversación telefónica celebrada la semana pasada entre Chirac, Aznar
      y el presidente argentino, Carlos Menem, ha modificado la decisión francesa de retrasar
      la apertura de negociaciones con Chile y con el Mercosur. El viceministro de Exteriores
      alemán, Günther Verheugen, expresó el deseo de la presidencia alemana de aprobar los
      mandatos de negociación antes de esa cita que reunirá a los jefes de Estado o de
      Gobierno de todos los países de América Latina y del Caribe y de la UE. Los países de
      Mercosur y Chile advirtieron a la UE el pasado fin de semana en México que quieren un
      acuerdo de libre comercio y que esperan que la cumbre de Río adopte compromisos
      claros en ese sentido. Fue él quien lanzó en 1997 la idea de reunir en una cumbre a los
      líderes de América Latina y Europa. La diplomacia española ha mantenido un breve y
      discreto silencio sobre el enfrentamiento que se registró el pasado viernes en Colonia,
      Alemania, entre el presidente del Gobierno español, José María Aznar, y el presidente
      francés, Jacques Chirac, sobre el mandato de negociación con Mercosur. Aunque la
      cumbre es mucho más que Mercosur -asistirán 17 Jefes de Estado y de Gobierno de
      Latinoamérica, 16 del Caribe y 15 de la UE-, las negociaciones con el bloque en el que
      están Brasil y Argentina son el escaparate de las relaciones de Europa con el resto del
      continente." Testores= "['europa', 'américa_latina', 'presidente', 'país', 'cumbre',
      'negociación', 'mercosur']" />
      <Subject IPTC= "Política & Tratados y Organizaciones & alliance" />
      <Document SCode= "19990602_PAIS(0).sect(1).Sect(0).artl(17).News(0).titl(0).
      STR(0)" href="NoticiasInternacionales/Junio/i39_19990602.txt">Francia trata de
      retrasar la negociación de un pacto comercial de la UE con América Latina.
      </Document>
      <Document SCode= "19990609_PAIS(0).sect(1).Sect(0).artl(15).News(0).titl(0).
      STR(0)" href="NoticiasInternacionales/Junio/i174_19990609.txt">El portazo de la UE a
      Mercosur amenaza el futuro de las relaciones con América Latina.</Document>
      <Document SCode= "19990601_PAIS(0).sect(1).Sect(0).artl(16).News(0).titl(0).
      STR(0)" href= "NoticiasInternacionales/Junio/i17_19990601.txt">América Latina busca
      en la UE la salida a las crisis financieras cíclicas.</Document>
    </Event>
  - <Event Id= "Cluster_343" Level= "1">
      <Subject IPTC= "Política & Tratados y Organizaciones & alliance" />
      <Document SCode= "19990618_PAIS(0).sect(1).Sect(0).artl(13).News(0).titl(0).
      STR(0)" href= "NoticiasInternacionales/Junio/i342_19990618.txt">Francia culpa a
      España del bloqueo de la negociación entre la UE y Mercosur.</Document>
    </Event>
```

Fig. 5.6.- Fragmento de una jerarquía creada.

5.8. CONCLUSIONES

En este capítulo presentamos nuestro sistema de detección de tópicos. En particular, mostramos un método para representar los documentos que incorpora las propiedades temporales, espaciales y semánticas de las noticias. De igual forma, definimos medidas de semejanza parciales para cada una de las componentes de la representación de las noticias y una medida de semejanza global que expresa que dos noticias abordan el mismo tópico si tienen una alta semejanza semántica, proximidad temporal y coincidencia en lugar.

Nuestro sistema de detección no se limita a encontrar un conjunto de tópicos, sino que construye una jerarquía de sucesos y tópicos con diferentes niveles de granularidad. En el primer nivel de la jerarquía se identifican los sucesos individuales. En los siguientes niveles estos sucesos son sucesivamente agrupados para formar sucesos más complejos y tópicos. Esta jerarquía obtenida permite a un analista de información identificar no sólo los tópicos sino además los sucesos en los que ellos se pueden descomponer.

Con el objetivo de facilitar el análisis de los tópicos detectados por nuestro sistema brindamos dos herramientas: la construcción automática de los resúmenes de cada tópico detectado, así como la identificación de la temática de cada uno de ellos.

El algoritmo de desambiguación propuesto en este capítulo no sólo se utiliza para obtener la representación conceptual de los documentos sino también para la obtención de las temáticas (tópicos IPTC) abordadas por los grupos detectados por el sistema.

Como hemos visto en este capítulo, en nuestro sistema de detección se utilizan los algoritmos que hemos propuesto en los capítulos anteriores.

En la tabla 5.5 ubicamos nuestro sistema de detección en la tabla de características generales de los sistemas existentes que mostramos anteriormente en el capítulo 2. A esta tabla le hemos añadido dos nuevas columnas que expresan si el sistema de detección realiza una asignación irrevocable de los documentos a los grupos y si los tópicos detectados dependen del orden de presentación de los documentos.

Sistema	**Modelo de representación**	**Medida de semejanza**	**Algoritmo de agrupamiento**	**Propiedades temporales**	**Dependencia del orden**	**Asignación irrevocable**
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	Sí	Sí
UMASS	Vectorial con pesos *TF-IDF*, *IDF* estimados a partir de corpus retrospectivo	Coseno	*1-NN*	Orden cronológico	Sí	Sí
Papka	Vectorial con variante del esquema *Okapi tf*	Suma pesada	*Single-Pass*	Modelo de umbral	Sí	Sí
UPENN	Vectorial con pesos *TF-IDF*	Coseno	*Single-Link*	No	No	No
IBM	Vectorial con variante de *TF-IDF*	Medida de *Okapi*	*Single-Pass*	Orden cronológico	Sí	Sí
Iowa	Vectorial con pesos *TF-IDF*	Coseno	Modelo de tubería	Orden cronológico	No	No

Kurt	Vectorial con pesos *tfc*	Coseno penalizada por el tiempo	*K-NN*	Ventana temporal y parámetros en la medida de semejanza	Sí	Sí
Brants	Vectorial con variante del *TF-IDF* incremental	Variante de la medida de Hellinger	*1-NN*	No	Sí	Sí
Dragón	Modelo del lenguaje	Distancia probabilística	*K-Means*	Parámetros en la medida de distancia	Sí	No
BBN	Modelo del lenguaje	BBN y métrica de selección	*K-Means Incremental*	No	No	No
TNO	Modelo del lenguaje	Semejanza probabilística	Variante del *Single-Pass*	No	Sí	No
JERARTOP	Vectorial	Medida que toma en cuenta las propiedades semánticas, espaciales y temporales.	Jerárquico incremental que utiliza como rutina el compacto incremental.	Parámetros en la medida de semejanza	No	No

Tabla 5.5.- Características generales de los sistemas de detección en línea.

Precisamente, por utilizar como rutina de agrupamiento el algoritmo compacto incremental, nuestro sistema de detección no realiza una asignación irrevocable de los documentos a los grupos ni depende del orden de presentación de los documentos.

A pesar de ubicarlo en esta tabla, debemos destacar que *JERARTOP* es algo más que un sistema de detección, pues a diferencia de todos los existentes, construye de forma completamente automática e incremental una jerarquía de tópicos/subtópicos y proporciona para cada uno de ellos su temática y un resumen.

6 EXPERIMENTOS

6.1 INTRODUCCIÓN

En este capítulo se describe el entorno experimental con el objetivo de evaluar la efectividad del sistema de detección presentado en el capítulo anterior.

En primer lugar, se discutirán los recursos utilizados en los experimentos: las colecciones de prueba, las herramientas utilizadas y las medidas de evaluación de la calidad que emplearemos.

Finalmente, mostraremos los experimentos realizados y las conclusiones a las que hemos llegado. En estos experimentos analizaremos el impacto en la detección que tiene cada una de las componentes de la representación de las noticias, así como de las diferentes variantes de funciones de semejanza que pueden utilizarse. Además, optimizaremos los parámetros que utilizan nuestros algoritmos en función de las medidas de evaluación de la calidad.

6.2 COLECCIONES DE PRUEBA

Para los experimentos utilizamos tres colecciones de prueba. La primera contiene 554 noticias publicadas en la esfera internacional del periódico español El País durante el mes de junio de 1999. Manualmente hemos construido una jerarquía de tópicos con dos niveles. En el primer nivel se han identificado 86 sucesos no unitarios, cuyo tamaño máximo es de 16 documentos. Estos sucesos han sido agrupados, a su vez, en 58 tópicos.

Ejemplos de tópicos detectados manualmente son:

- Elecciones en Sudáfrica.
- Elecciones en Indonesia.
- El caso Pinochet en Chile.
- El asesinato del obispo Gerardi en Guatemala.
- Un terremoto en México.
- Los estragos del Mitch en Centroamérica.
- 21 sucesos relacionados con la crisis de Kosovo y el conflicto serbio.
- Elecciones europeas.
- Juicio al líder kurdo Ocalan.
- Secuestros de la guerrilla colombiana.
- La visita del Papa a Polonia.
- Sustitución de Robaina en Cuba.

La tabla 6.1 muestra ejemplos de sucesos manuales detectados en el tópico de la guerra de Kosovo.

Descripción del suceso	Rango de fechas	#Docs.
Negociaciones del acuerdo de paz	Junio 07-08	16
Firma del acuerdo de paz	Junio 09-11	13
Reacciones políticas en Yugoslavia	Junio 14-19	16
Las tropas serbias abandonan Kosovo	Junio 10-19	6
Despliegue de las tropas de la OTAN	Junio 11-19	13
Retorno de los refugiados albaneses a Kosovo	Junio 16-24	8
Refugiados serbios escapan de Kosovo	Junio 14-24	11

Tabla 6.1.- Sucesos del tópico "Guerra de Kosovo".

Al primer nivel de la jerarquía manual construida le llamaremos *Nivel de sucesos* y al segundo, *Nivel de tópicos*. Esta jerarquía describirá la estructura de tópicos y sucesos de esta colección.

La segunda colección de prueba utilizada fue la versión 4.0 de TDT2. Como mencionamos en el capítulo 2, esta colección contiene noticias en inglés publicadas en 1998 y 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*).

En nuestros experimentos trabajamos con la colección completa, es decir, sin dividirla en conjunto de entrenamiento, de prueba y de evaluación. En total, nuestra colección está conformada por 9824 noticias etiquetadas en 193 tópicos. De ellas, existen 153 noticias que están etiquetadas en más de un tópico.

La colección TDT2 no está etiquetada en diferentes niveles de granularidad, por lo que sólo contamos con la clasificación manual a nivel de tópicos.

Por último, la tercera colección utilizada en nuestros experimentos fue la TREC-5 en castellano[8]. Esta colección contiene 693 noticias publicadas por la agencia AFP durante 1994 y clasificadas en 23 tópicos. La tabla 6.2 muestra algunos tópicos de esta colección y la cantidad de documentos etiquetados con él.

Tópico	Descripción del tópico	# Docs.
SP52	Guerra de los rebeldes vascos	13
SP53	Estado de la mujer en América Latina	46
SP54	Recursos marinos mundiales	37
SP55	Destino de Carlos Andrés Pérez	108
SP58	Financiamiento de la campaña electoral de Samper	49
SP60	Métodos usados por los narcotraficantes	47
SP64	Extinción de la iguana verde	9
SP65	Actividades de Raúl Castro	29
SP68	SIDA en Argentina	12

Tabla 6.2.- Ejemplos de tópicos de TREC-5.

[8] http://trec.nist.gov

6.3 HERRAMIENTAS UTILIZADAS

El sistema de detección *JERARTOP* utiliza los recursos y herramientas que describiremos a continuación:

- Etiquetador léxico-morfológico: genera las categorías gramaticales de cada palabra junto con su información morfológica.
- *WordNet*.
- Dominios de Magnini.
- Tópicos IPTC.

El sistema de detección propuesto es independiente del idioma de las noticias que se van a procesar. En este caso, lo que cambia son las herramientas que se utilizan en cada idioma.

Para procesar las noticias en castellano, utilizamos como analizador morfológico *MACO+* [Carm98], desarrollado por el Grupo de Procesamiento de Lenguaje Natural de la Universidad Politécnica de Catalunya. MACO+ se basa en los modelos estocásticos ECGI extendidos [PlaF00]. Una descripción detallada de este analizador puede encontrarse en http://www.lsi.upc.es/~nlp.

El etiquetador léxico-morfológico para textos en inglés, utilizado en el sistema propuesto, se conoce con el nombre de *TreeTagger* [Schm94]. Este etiquetador fue desarrollado dentro del Proyecto "*Textual corpora and tools for their exploration*" en el Instituto de Lingüística Computacional de la Universidad de Stuttgart. La característica fundamental del *TreeTagger* es analizar y desambiguar las categorías léxicas de las palabras para textos no restringidos en inglés. Para ello se basa en los modelos de Markov y en los árboles de decisión con el objetivo de obtener estimaciones más fiables. Una descripción detallada de este analizador puede encontrarse en http://www.cis.upenn.edu/~treebank.

Ambos analizadores reciben como entrada un texto plano y obtienen como salida las palabras con sus lemas y una etiqueta que indica la categoría gramatical de cada palabra junto con su información morfológica.

La base de conocimientos léxica utilizada para los textos en inglés fue la versión 1.6 de *WordNet* y para el castellano, la base *EuroWordNet*. Éstos, conjuntamente con los Dominios de Magnini y los tópicos IPTC son los recursos externos que utilizamos en el algoritmo de desambiguación propuesto en el capítulo 5.

6.4 MEDIDAS DE EVALUACIÓN DE LA CALIDAD

Para evaluar la calidad de nuestro sistema de detección utilizaremos las siguientes medidas:

- F1 macro-promediada.
- Coste de detección macro-promediado.
- Coste de detección normalizado.

Estas medidas fueron explicadas en el epígrafe 2.2.4. En el caso del coste de detección, para evaluar la colección de TDT2 se usó $P_{tópico} = 0.02$. En la colección de El País se estimaron valores diferentes de $P_{tópico}$ en cada nivel de agrupamiento. Esto se

debe a que los grupos son mayores en los niveles superiores de la jerarquía. Cuanto mayor es el tamaño de los grupos, mayor es la probabilidad de que un documento pertenezca a un grupo dado. De esta forma, en nuestros experimentos se estimó $P_{tópico}$=0.012 en el nivel de sucesos y $P_{tópico}$ = 0.017 en el nivel de tópicos.

En ambas colecciones se usaron las constantes $coste_{f_a}$ = 0.1 y $coste_m$ = 1.

La medida F1 exige que los grupos detectados por el sistema sean prácticamente iguales que las clases manuales, mientras que el coste de detección exige con más fuerza que los errores por omisión sean mínimos, es decir, exige que para cada clase manual exista un grupo que la contenga, aunque este grupo incluya otros documentos que no pertenecen a la clase. Es por eso, que en la medida que los grupos sean más grandes se obtienen costes de detección menores.

La clasificación manual de la colección TDT2 contiene, como mencionamos anteriormente, noticias multi-tópicos. Tal y como recomienda la metodología de evaluación de TDT, estas noticias son agrupadas pero se ignoran al evaluar la calidad del proceso de detección [TDT03].

6.5 EXPERIMENTOS GENERALES

En este epígrafe explicaremos los experimentos realizados en relación con la eficiencia y estabilidad del algoritmo compacto incremental en el proceso de detección de tópicos.

6.5.1 Eficiencia del algoritmo

En el capítulo 3 analizamos que la complejidad temporal en el caso peor del algoritmo compacto incremental es cuadrática. En este epígrafe mostraremos cuál es el comportamiento real del algoritmo.

Cuando utilizamos la medida de semejanza global propuesta en el epígrafe 5.6.4 podemos optimizar el algoritmo compacto incremental, de forma tal que no tengamos que comparar cada nuevo documento con todos los existentes, sino sólo con los documentos separados en el tiempo del nuevo en un intervalo no mayor de β_{tiempo} días. Note que la semejanza con el resto de los documentos siempre sería cero.

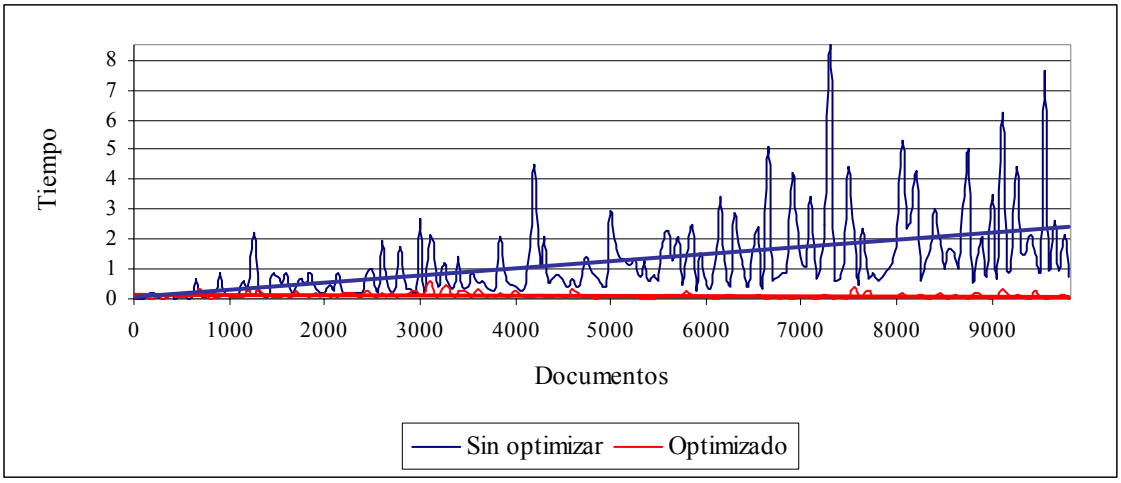


Fig. 6.1.- Comportamiento del algoritmo compacto incremental.

La figura 6.1 muestra el tiempo (en segundos) que consume el algoritmo ante la llegada de cada documento, considerando y sin considerar la optimización de la ventana temporal en la colección de TDT2.

Como podemos observar, el algoritmo compacto optimizado se comporta prácticamente lineal, lo cual es una característica deseada cuando se trabaja con grandes volúmenes de noticias.

6.5.2 Estabilidad del algoritmo

El algoritmo compacto incremental utiliza un parámetro: el umbral de semejanza β_0.. De este valor depende la cantidad de conjuntos compactos que se formarán. Cuanto mayor es el β_0 más se le exige a los documentos para concluir que abordan el mismo tópico y, por tanto, mayor es la cantidad de grupos que se forman. Por el contrario, mientras menor es el β_0, mayor cantidad de documentos se unirán y, por tanto, se formarán menos grupos.

En este epígrafe analizaremos el impacto que tiene la variación de este umbral en la calidad de los resultados obtenidos en el proceso de detección de tópicos en la colección El País. Para ello, variamos el parámetro β_0 en los valores 0.1, 0.15, 0.2, 0.23, 0.25, 0.3, 0.33, 0.4, 0.5, 0.55 y 0.6. En este caso, utilizamos como medida de semejanza entre documentos la medida del coseno.

En las figuras 6.2 y 6.3 mostramos los valores de la medida F1 y el coste de detección en el nivel de sucesos y el nivel de tópicos con respecto al umbral β_0, respectivamente.

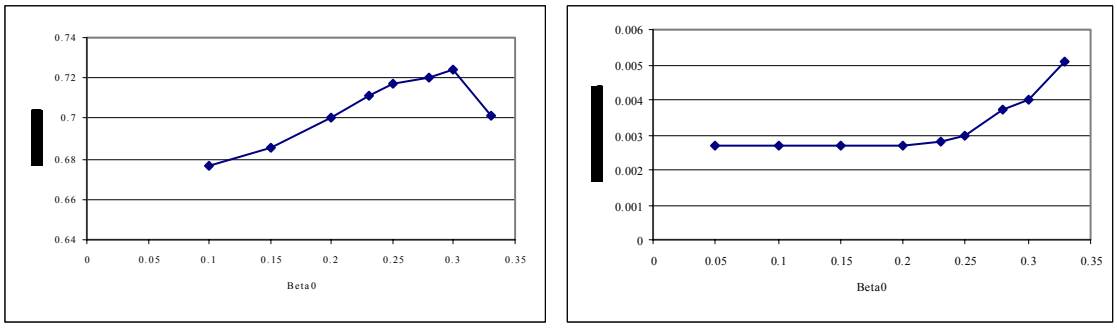


Fig. 6.2.- Medida F1 y Coste de detección en el nivel de sucesos.

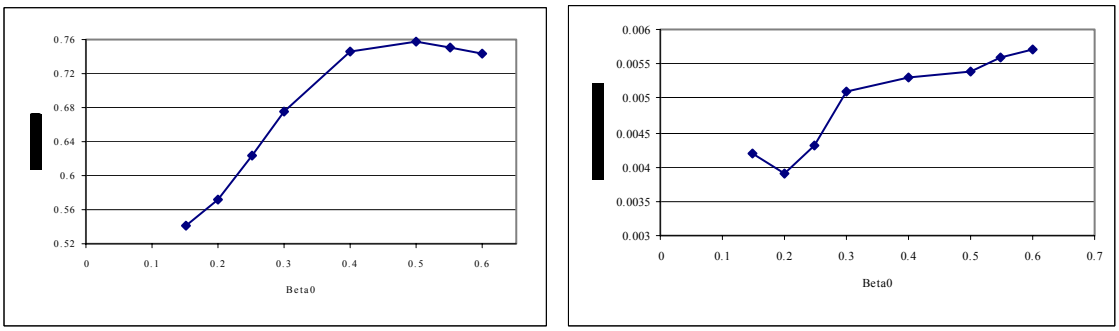


Fig. 6.3.- Medida F1 y Coste de detección en el nivel de tópicos.

Como puede observarse, los valores de la medida F1 se mantienen bastante estables en el rango de 0.23 a 0.3 en el nivel de sucesos y de 0.4 a 0.6 en el nivel de tópicos. Por lo tanto, pensamos que utilizando un valor de β_0 en este rango pueden obtenerse resultados satisfactorios, sin necesidad de invertir mucho esfuerzo optimizando este parámetro.

En el caso del coste de detección en el nivel de sucesos se mantiene un valor constante de 0.0027 para los primeros valores de β_0. Por lo general, los óptimos por coste de detección obtienen bajos valores de F1, y como veremos más adelante, esto es una constante en todos nuestros experimentos. En estos casos, el umbral de semejanza óptimo es mucho menor que con respecto a la medida F1 con el objetivo de que se formen grupos más grandes.

Por último, nótese que el algoritmo muestra mayor estabilidad con respecto al parámetro β_0 en el nivel de sucesos, donde los grupos son mucho más pequeños y coherentes.

6.6 Experimentos en la colección El País

En este epígrafe analizaremos los experimentos realizados con la colección de El País. Inicialmente, para realizar un estudio completo de la eficacia de nuestro sistema de detección se realizaron experimentos con todas las variantes de la medida de semejanza analizada en el epígrafe 5.6.4 y diferentes valores de β_0 y β_{tiempo}. Se seleccionaron en cada caso como parámetros óptimos, aquellos que daban el mayor valor en la medida F1 y el más bajo en el coste de detección.

Primeramente, analizaremos en el nivel de sucesos el impacto de cada una de las componentes de la representación de las noticias y las diferentes medidas de semejanza que pueden utilizarse. Luego, estudiaremos cómo se comporta nuestro sistema de detección en el nivel de tópicos. Por último, compararemos los resultados obtenidos en esta colección con otros sistemas de detección existentes.

6.6.1 Impacto de la componente temporal

En el capítulo 5 explicamos dos variantes de representación de las propiedades temporales de las noticias: considerando sólo las fechas de publicación o extrayendo automáticamente las entidades temporales del texto de las noticias. Teniendo en cuenta, lo anterior, definimos en el epígrafe 5.6.2 dos funciones de distancia para comparar las propiedades temporales de las noticias.

En este experimento evaluaremos el impacto de la componente temporal de las noticias en la calidad de un sistema de detección y determinaremos cuál de las variantes propuestas obtiene mejores resultados. Para ello, utilizaremos como medida de semejanza global aquella que sólo tiene en cuenta las componentes de términos y temporal de las noticias, ignorando la componente de lugar (vea epígrafe 5.6.4) y variaremos el umbral de tiempo β_{tiempo}.

En las figuras 6.4 y 6.5 mostramos los resultados de la medida F1 y el coste de detección con respecto al umbral β_{tiempo}, respectivamente. Estos gráficos muestran una notable merma en la efectividad del sistema cuando no se considera la componente temporal ($\beta_{tiempo} = \infty$). Como consecuencia, concluimos que la componente temporal mejora sensiblemente la calidad de los tópicos generados por el sistema. Note que el umbral óptimo de tiempo es aproximadamente 6 días.

La distancia temporal que sólo contempla las fechas de publicación obtiene mejores resultados que la que utiliza las fechas extraídas del texto, por lo que consideramos que en esta colección no vale la pena invertir tiempo ni recursos extrayendo las referencias temporales. A partir de ahora, en todos los experimentos siempre consideraremos las propiedades temporales de las noticias como sus fechas de publicación.

El impacto de la componente temporal en nuestro caso no es tan importante como en otros trabajos, como por ejemplo [Llid01], debido en nuestra opinión a la calidad del algoritmo de agrupamiento. Así, para algoritmos con decisiones irrevocables como el *Single-Pass*, la aportación de la componente temporal es significativa, ya que mejora la toma de dichas decisiones. Esto no sucede en el algoritmo propuesto, ya que cada documento se asocia al grupo donde se encuentra el documento que más se le parece y éste puede cambiar a lo largo del proceso de agrupamiento. Aquí el tiempo simplemente aporta un poco más de semántica a la medida de semejanza, que puede ser decisiva en aquellos documentos donde exista cierto grado de ambigüedad, como es el caso de los grupos referentes a la crisis de Kosovo o las cumbres políticas.

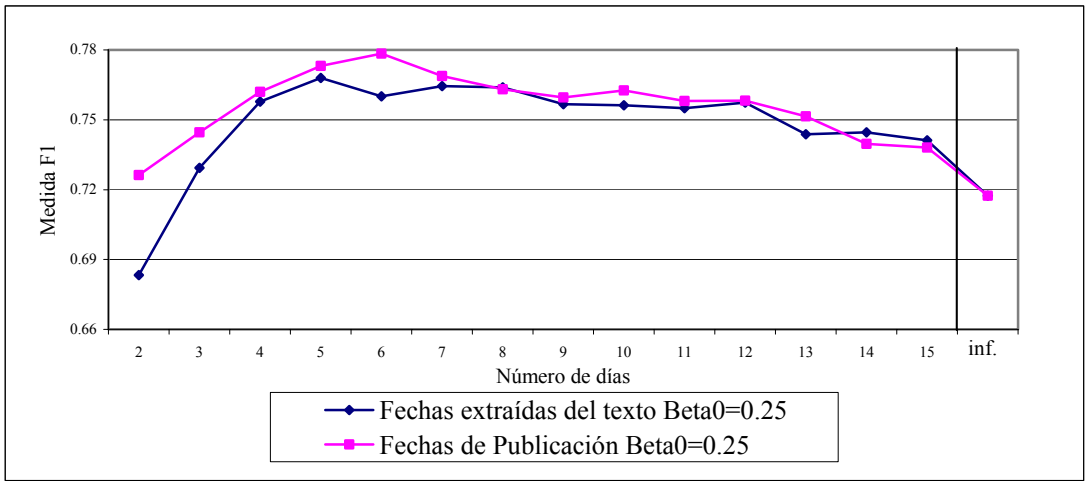


Fig. 6.4.- Impacto de la componente temporal con respecto a la medida F1.

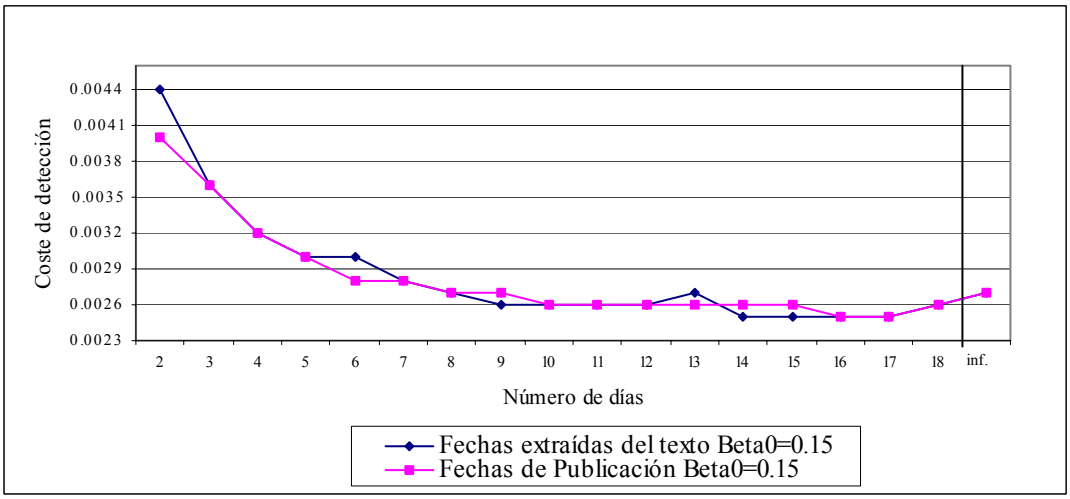


Fig. 6.5.- Impacto de la componente temporal con respecto al coste de detección.

6.6.2 Impacto de la componente de lugar

En este experimento evaluaremos el impacto de la componente de lugar en la calidad del proceso de detección. Las figuras 6.6 y 6.7 muestran los resultados obtenidos en la medida F1 y el coste de detección respectivamente, cuando utilizamos como función de semejanza la que contempla las tres componentes de la representación de las noticias (representada con el color rosado) y aquella que ignora la componente de lugar (representada con el color azul). En estas gráficas se varió el umbral de tiempo de 2 a 18 días.

Como podemos observar, considerando las tres componentes de la representación de las noticias se obtienen mejores resultados que ignorando el lugar. Nuevamente el valor óptimo del umbral β_0 es menor en el caso del coste de detección que cuando consideramos la medida F1.

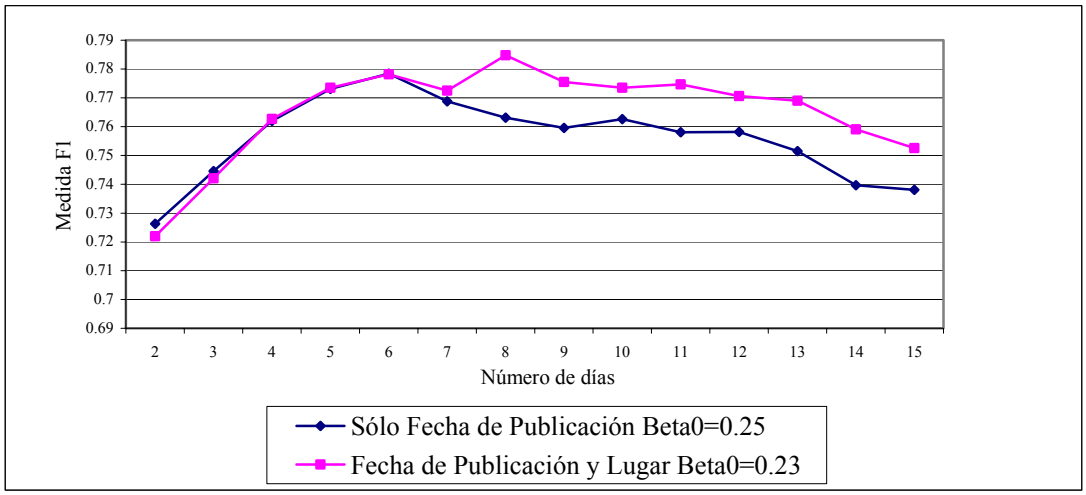


Fig. 6.6.- Impacto de la componente espacial con respecto a la medida F1.

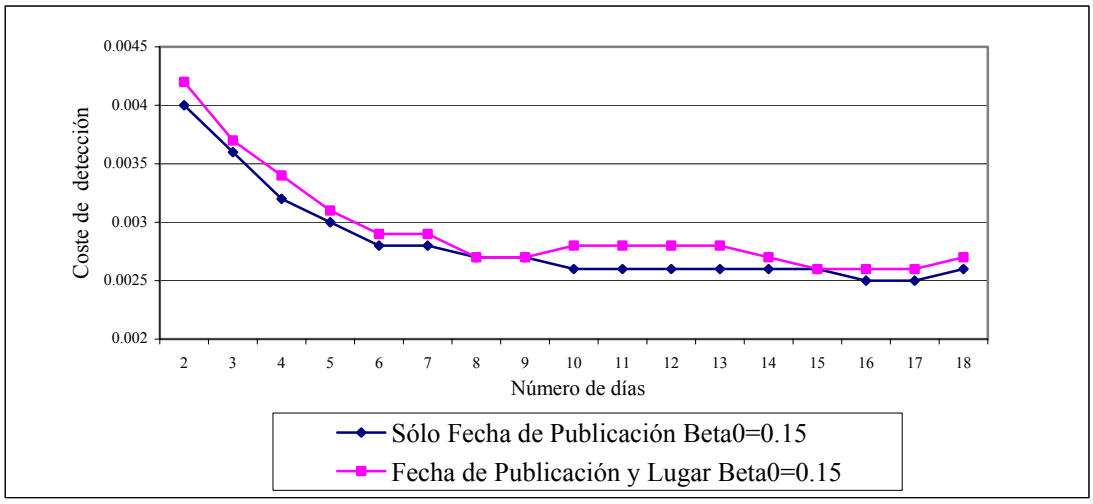


Fig. 6.7.- Impacto de la componente espacial con respecto al coste de detección.

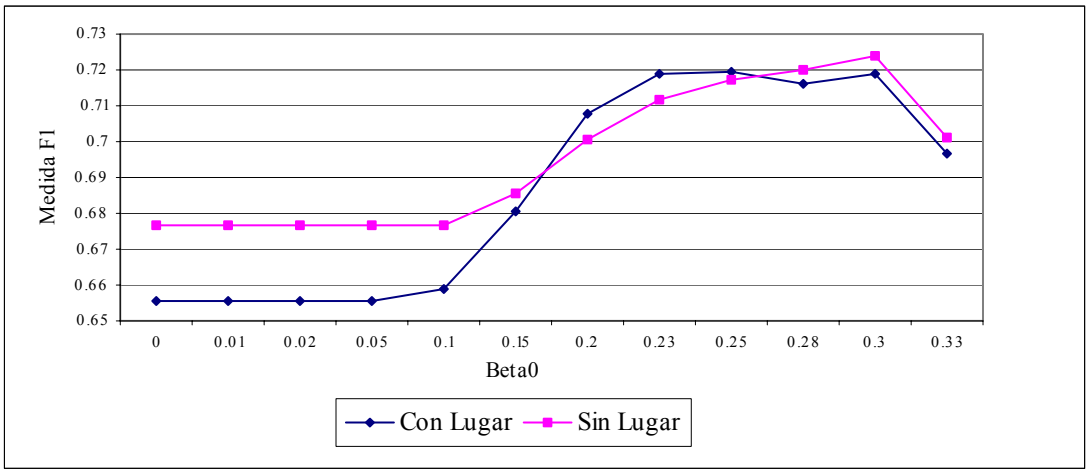


Fig. 6.8.-Considerando sólo la componente espacial con respecto a la medida F1.

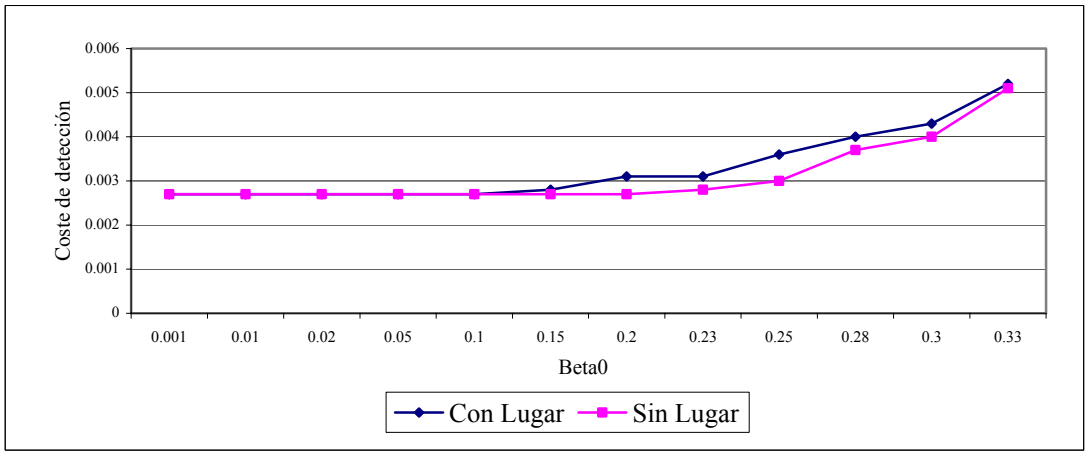


Fig. 6.9.- Considerando sólo la componente espacial con respecto al coste de detección.

Para analizar si la contribución aislada de la componente de lugar mejora la calidad del proceso de detección, comparamos los resultados obtenidos considerando como función de semejanza aquella que ignora la componente temporal con los obtenidos por la medida del coseno. Las figuras 6.8 y 6.9 muestran los resultados de las medidas de calidad en estos casos cuando variamos el umbral β_0.

En estas figuras se observa que la medida del coseno (sin lugar) obtiene resultados superiores a la medida que contempla la componente de lugar solamente. Por lo tanto, podemos concluir que la componente de lugar sólo es útil si se combina con la componente temporal. Esto nos lleva a pensar que existe una fuerte correlación entre el lugar y el tiempo, tal y como cabría esperar en la descripción de los sucesos.

6.6.3 Impacto de las frases

Como hemos analizado anteriormente, los mejores resultados se obtienen cuando utilizamos como medida de semejanza aquella que tiene en cuenta las componentes de términos, temporal (considerando las fechas de publicación) y espacial en la representación de las noticias (vea epígrafe 5.6.4). Utilizando esta medida, se obtuvo un valor óptimo de F1 de 0.7848 y un coste de detección de 0.0026. En el primer caso, los

valores óptimos se obtuvieron para $\beta_0 = 0.23$ y $\beta_{tiempo} = 8$ y en el segundo para $\beta_0 = 0.15$ y $\beta_{tiempo} = 15$. Estos experimentos se realizaron con la representación a nivel de términos (lemas de las palabras) de las noticias.

Como mencionamos en el epígrafe 5.3 también podemos utilizar la representación a nivel de frase o la representación conceptual de las noticias.

En este epígrafe analizaremos qué impacto tiene la representación a nivel de frase en la calidad del proceso de detección. Para ello, utilizamos la medida de semejanza y los parámetros que obtuvieron los mejores resultados en los experimentos anteriores. Las figuras 6.10 y 6.11 muestran los resultados obtenidos cuando variamos el umbral del tiempo.

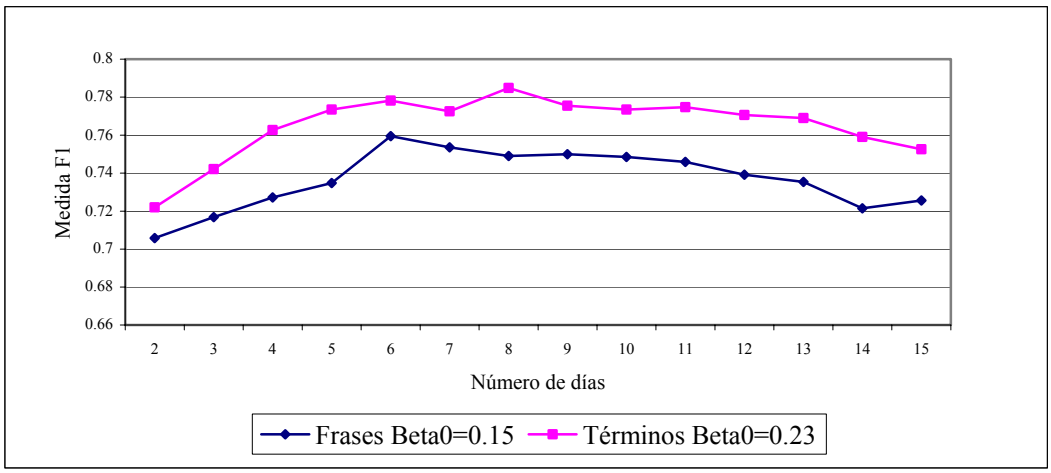


Fig. 6.10.- Impacto de las frases con respecto a la medida F1.

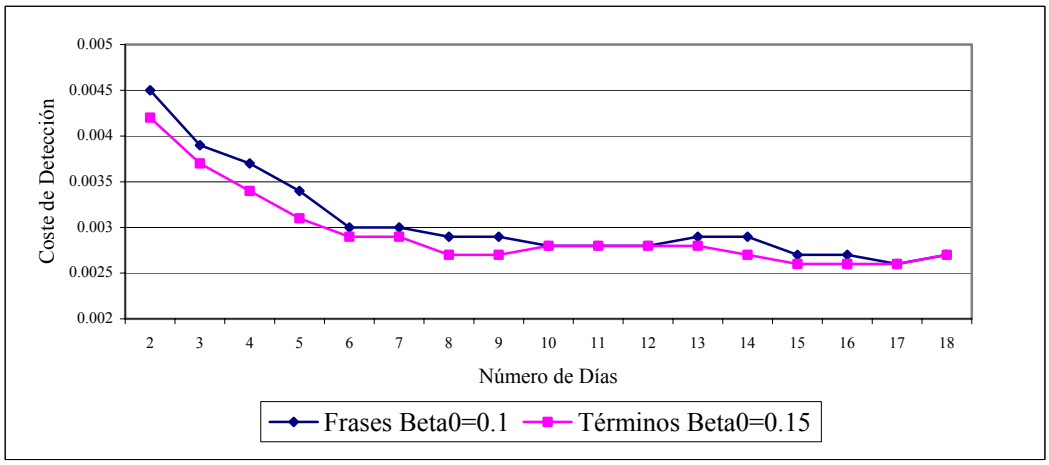


Fig. 6.11.- Impacto de las frases con respecto al coste de detección.

En estas gráficas se puede observar claramente que la detección de tópicos es superior en la representación a nivel de términos que en la representación a nivel de frases. No obstante, consideramos que la representación a nivel de frases obtenida es demasiado simple, por lo que esto requiere de un estudio más profundo en el futuro. En cualquier caso, estos resultados están en consonancia con los obtenidos en otros trabajos

de Recuperación de Información, como por ejemplo [Kost03], donde la inclusión de frases no consigue mejorar la calidad de estos sistemas.

6.6.4 Impacto de los conceptos

De la misma forma que hicimos en el epígrafe anterior, evaluamos el impacto de la representación conceptual de las noticias en nuestro sistema de detección. Para la obtención de la representación conceptual, en el algoritmo de desambiguación se utilizaron los 5 dominios de Magnini y los 5 tópicos IPTC más frecuentes y no se desambiguaron ni los adjetivos ni los verbos.

Utilizando la medida de semejanza semántica que compara las representaciones conceptuales de las noticias (ver epígrafe 5.6.1) conjuntamente con la componente temporal (fechas de publicación) y la componente espacial, se obtuvo un valor de F1 de 0.7234 y un coste de detección de 0.0042 para $\beta_0 = 0.25$ y $\beta_{tiempo} = 6$.

A pesar de que los resultados obtenidos son inferiores a los que obtuvimos con la representación a nivel de términos de las noticias, éstos pueden considerarse aceptables y, sobre todo, pudieran mejorarse. Basta recordar que en estos resultados influyen considerablemente el algoritmo de desambiguación y la cobertura de la *EuroWordNet*. Pensamos que si se elevara la precisión y la relevancia del algoritmo de desambiguación conjuntamente con una mayor información de relaciones semánticas en la *EuroWordNet* estos resultados podrían ser mejores. En este sentido, los resultados confirman los obtenidos en otros trabajos sobre el impacto de la desambiguación de los términos en tareas de Recuperación de la Información [Gonz99].

Es bueno señalar, además, que el uso de la representación conceptual de las noticias tiene una ventaja adicional, pues permite el procesamiento de colecciones multilingües al homogeneizar la representación de las noticias en función de los *synsets* de *WordNet*.

6.6.5 Comportamiento a nivel de tópico

En el capítulo 5 explicamos cómo el sistema de detección *JERARTOP* obtiene una jerarquía de tópicos con diferentes niveles de granularidad. En los epígrafes anteriores estudiamos el comportamiento de nuestro sistema en el nivel de sucesos, es decir, en el primer nivel de la jerarquía. En este epígrafe mostraremos los resultados obtenidos en el nivel de tópicos (segundo nivel).

Utilizando los parámetros óptimos obtenidos en los experimentos anteriores en el nivel de sucesos, las tablas 6.3 y 6.4 muestran para cada una de las variantes de la medida de semejanza los mejores resultados obtenidos en el nivel de tópicos.

Medida	β_{tiempo}	β_0	Medida F1
Coseno	-	0.5	0.8025
Temporal	10	0.5	0.8114
Temporal y espacial	10	0.5	0.8114

Tabla 6.3.- Mejores resultados con respecto a la medida F1.

Medida	β_{tiempo}	β_0	CDET Macro	CDET Norm.
Coseno	-	0.4	0.0037	0.2204
Temporal	25	0.4	0.0037	0.2204
Temporal y espacial	25	0.4	0.0039	0.2288

Tabla 6.4.- Mejores resultados con respecto al coste de detección.

Como podemos observar, en el nivel de tópicos, aunque aún son importantes, las componentes temporal y espacial tienen un menor impacto en la medida F1 y no afectan al coste de detección. Consideramos, además, que los valores de la medida F1 obtenidos en el nivel de tópicos son muy buenos, comparados con los de la literatura.

6.6.6 Comparación con otros métodos

Finalmente, se ha implementado tres de los sistemas mencionados en el capítulo 2 y se ha evaluado su efectividad (optimizando sus parámetros) usando la colección de El País. En las tablas 6.5 y 6.6 comparamos la eficacia de nuestro sistema de detección con las otras aproximaciones respecto a la medida F1 y al coste de detección.

En particular, comparamos los resultados obtenidos por nuestro sistema en los niveles de sucesos y de tópicos con los sistemas UMASS, CMU y el de Papka. Además lo comparamos con nuestro propio algoritmo de detección pero en lugar de las frecuencias *TF* utilizando los *TF-IDF* estimados incrementalmente a partir de un corpus retrospectivo (usamos una colección de noticias del propio periódico publicadas en el mes de mayo de 1999).

Como puede notarse, nuestro sistema supera claramente a los otros sistemas en el proceso de detección de tópicos en ambos niveles.

Nivel de sucesos		
Sistema	**Parámetros**	**Medida F1**
UMASS	Umbral = 0.25	0.6515
CMU	m = 100 t_n= 0.2	0.6892
Papka	θ = 0.15, β = 0.000005 y n = 400	0.4636
JERARTOP con *TF-IDF*	β_0 = 0.1, β_{tiempo} = 8	0.7669
JERARTOP	β_0 = 0.23, β_{tiempo} = 8	0.7848
Nivel de tópicos		
Sistema	**Parámetros**	**Medida F1**
UMASS	Umbral = 0.2	0.7128
CMU	m = 500 t_n= 0.2	0.7246
Papka	θ = 0.15, β = 0.000005 y n = 400	0.4314
JERARTOP con *TF-IDF*	β_0 = 0.1, β_{tiempo} = 8	0.7698
JERARTOP	β_0 = 0.5, β_{tiempo} = 10	0.8114

Tabla 6.5.- Comparación con otros sistemas de detección respecto a la medida F1.

Nivel de sucesos		
Sistema	**Parámetros**	**CDET**
UMASS	Umbral = 0.15	0.0038
CMU	m = 300 t_n= 0.15	0.0039
Papka	θ = 0.15, β = 0.000005 y n = 400	0.0123
JERARTOP con *TF-IDF*	β_0 = 0, β_{tiempo} = 8	0.0029
JERARTOP	β_0 = 0.15, β_{tiempo} = 15	0.0026
Nivel de tópicos		
Sistema	**Parámetros**	**CDET**
UMASS	Umbral = 0.15	0.0035
CMU	m = 400 t_n= 0.1	0.0042
Papka	θ = 0.15, β = 0.000005 y n = 400	0.0173
JERARTOP con *TF-IDF*	β_0 = 0.05, β_{tiempo} = 12	0.0027
JERARTOP	β_0 = 0.4, β_{tiempo} = 25	0.0039

Tabla 6.6.- Comparación con otros sistemas de detección respecto al coste de detección.

6.7 EXPERIMENTOS EN LA COLECCIÓN TDT

En los epígrafes anteriores mostramos los experimentos realizados con la colección El País con el objetivo de analizar el impacto de cada una de las componentes de la representación de las noticias, seleccionar la medida de semejanza más adecuada y optimizar los parámetros de los algoritmos. Como vimos anteriormente, la medida de semejanza que obtiene mejores resultados es la que tiene en cuenta las siguientes componentes:

- Componente de términos.
- Componente temporal expresada a partir de las fechas de publicación.
- Componente espacial.

Teniendo en cuenta lo anterior, realizamos varios experimentos con la colección TDT2 con el objetivo de crear una jerarquía de dos niveles: uno de sucesos y otro de tópicos. A pesar de que realizamos varios experimentos con jerarquías de mayor cantidad de niveles los resultados fueron inferiores.

En el nivel de sucesos utilizamos la medida de semejanza global óptima y los valores de $\beta_0 = 0.25$ y $\beta_{tiempo} = 20$. Las tablas 6.7 y 6.8 muestran los resultados obtenidos en el nivel de tópicos con respecto a la medida F1 y el coste de detección, respectivamente. De forma similar a lo ocurrido en la colección El País, las componentes temporal y espacial tienen un menor impacto en el nivel de tópicos.

Medida	β_{tiempo}	β_0	Medida F1
Coseno	-	0.4	0.7643
Temporal	50	0.4	0.779
Temporal y especial	50	0.4	0.779

Tabla 6.7.- Mejores resultados con respecto a la medida F1.

Medida	β_{tiempo}	β_0	CDET Macro	CDET Norm.
Coseno	-	0.4	0.0052	0.2577
Temporal	50	0.4	0.0054	0.2688
Temporal y espacial	50	0.4	0.0054	0.2688

Tabla 6.8.- Mejores resultados con respecto al coste de detección.

Desafortunadamente, el analizador morfológico usado para colecciones en inglés tiene algunos inconvenientes en la detección de estructuras multipalabras. Además muchos nombres propios no pueden ser detectados por problemas en el corpus TDT2 debido a la ausencia de las mayúsculas en muchas noticias. Este problema, por ejemplo, impide la detección de muchos topónimos. No obstante, los resultados obtenidos por nuestro sistema de detección han sido buenos.

Si observamos la tabla 2.2 del capítulo 2 donde se muestran los costes de detección normalizados obtenidos en el corpus de evaluación TDT2 por los sistemas de detección existentes, nos damos cuenta de que nuestro sistema de detección obtiene resultados satisfactorios. Es bueno señalar que los resultados de la tabla 2.2 se corresponden con el corpus de evaluación mientras que los nuestros se corresponden con la colección TDT2 completa. Además, es preciso recordar que nuestro sistema de detección no sólo obtiene resultados similares a los mejores sistemas sino que también brinda una información adicional: la jerarquía de tópicos con diferentes niveles de granularidad.

6.8 EXPERIMENTOS DE OBTENCIÓN DE LAS TEMÁTICAS

Para evaluar la calidad de nuestro método de obtención de temáticas (ver epígrafe 5.7.2) utilizamos la colección de El País. La tabla 6.9 muestra para los tópicos de esta colección, la temática en que fueron clasificados y una evaluación hecha por nosotros de esta clasificación.

Tópico	Temática	Eval.	Tópico	Temática	Eval.
Elecciones en Suráfrica	Política/Gobierno/Presidente	R	Vínculos gobierno mexicano con narcotráfico	Política/Gobierno/Presidente	B
Elecciones en Indonesia	Política/Sistema político	R	Cinturón eléctrico en EUA.	Economía y Finanzas/Vestuario	M
Elecciones europeas	Política/Elecciones	B	Torturas de la policía en México	Policía y Justicia	B
Dimisión gobierno belga	Política y Partidos/ Sistema político	B	Dimisión del jefe de policía en Brasil	Política/Sistema político	B
Elecciones en Francia	Política/Gobierno/Presidente	R	Juicio de Ocalan	Policía y Justicia/Juicio	B
Elecciones en alcaldía de Bolonia	Política/Autoridades locales	B	Posiciones de los gobiernos en el caso Pinochet.	Policía y Justicia/ Derechos	B
Referendum en Timor Oriental	Política/Elecciones/Votación	B	Crímenes de oficiales chilenos	Policía y Justicia /Criminalidad/ Homicidio	B
Caso Pey	Economía y Finanzas/Negocios	B	Asesinato del obispo Gerardi	Policía y Justicia /Criminalidad/ Homicidio	B
Financiación de la campaña del presidente mexicano	Política/Gobierno/Presidente	B	Escuadrones de la muerte y policía de El Salvador	Policía y Justicia	B
Investigación de la matanza de Tlatelolco	Política/Gobierno/Presidente	R	Terremoto en México	Desastres naturales/Terremoto	B
Huracanes en Centroamérica	Desastres naturales	B	Final guerra de Kosovo	Política/Tratados /Alianza	B
Fosas comunes en Kosovo y envío de tropas OTAN	Política/Tratados /Alianza	B	Retorno de refugiados en Kosovo	Política/Tratados /Alianza	R
Reconstrucción en Yugoslavia	Política/Tratados /Alianza	B	Acusación de Milosevic en el Tribunal Internacional	Política/Gobierno/Presidente	R
Conflicto de Cachemira	Guerras y conflictos	B	Guerrilla colombiana	Guerras y conflictos	B
Diálogo de paz en Irlanda del Norte	Política/Tratados	B	Guerra Israel-Líbano	Guerras y conflictos/Guerrilla	R
Aniversario de Tiananmen	Política/Autoridades locales	R	Situación en Argelia	Política/Gobierno	B
Matanza de jueces en Líbano	Política/Gobierno/Ejecutivo	M	Conflicto entre ambas Coreas	Guerras y conflictos	B
Drogas y miskitos	Medio ambiente/ Recursos naturales/ Río	M	Proliferación de armas en El Salvador	Guerras y conflictos/ Armas	B
UE-Mercosur	Política/Tratados /Alianza	B	Europa-América Latina	Política/Tratados /Alianza	B
OEA-América Latina	Política/Tratados /Alianza	R	Cumbre G-8	Economía y Finanzas	M
Relaciones Fidel y Aznar	Política/Gobierno/Presidente	B	Visita del Papa a Polonia	Asuntos sociales y Familia/Padre	M

Asesinato de Paco Stanley	Policía y Justicia /Criminalidad/ Homicidio	B	Viaje de Salinas a México	Política/Gobierno/Presidente	B
Cambios gobierno de Yugoslavia	Política y Gobierno	B	Destitución de Robaina	Política y Gobierno / Ministro	B
Cambios gobierno ruso	Política/Gobierno/Presidente	B	Defensa común europea (Elección Mr. PESC)	Política/Tratados /Alianza	B
Formación de gobierno en Israel	Política/Gobierno/Departamento de Gobierno	B	Campaña de Chávez	Política/Gobierno/Presidente	B
Destitución de funcionarios del turismo en Cuba	Economía y Finanzas / Negocios	R	Elecciones en Nicaragua	Política y Elecciones	B

Tabla 6.9.- Evaluación de la temática asignada a cada tópico.

Como podemos ver, en la mayoría de los casos la temática asignada a cada tópico es la correcta, por lo que consideramos que el método obtiene resultados satisfactorios. Así, considerando solo las respuestas buenas, el método consigue una precisión del 71%, mientras que si consideramos también como buenas las clasificadas como regular, obtenemos una precisión del 88%.

6.9 EXPERIMENTOS DE OBTENCIÓN DE RESÚMENES

En el capítulo 4 explicamos el método de construcción automática de resúmenes utilizado en nuestro sistema de detección. Para evaluar su efectividad utilizamos las colecciones de El País y TREC-5.

Como mencionamos anteriormente en la colección El País se identificaron 86 sucesos no unitarios. Uno de ellos aborda el tópico del asesinato del famoso presentador mexicano Paco Stanley. Este grupo está formado por 5 noticias, cuyos títulos, fechas de publicación y longitud (número de palabras) se muestran en la tabla 6.10.

Fecha de publicación	Título	Longitud
1999-6-8	Conmoción en México por el asesinato a tiros de un famoso presentador.	531
1999-6-9	El espionaje mexicano afirma que el cómico asesinado tenía vínculos con el narcotráfico.	748
1999-6-9	Las televisiones espolean la indignación contra Cárdenas.	298
1999-6-10	'Se creó un ambiente de histeria colectiva'	203
1999-6-10	La muerte de Stanley crispa el ambiente político en México	540

Tabla 6.10.- Documentos acerca del asesinato de Paco Stanley.

La unión de los testores típicos de longitud máxima encontrados es: {asesinato, mexicano, muerte}. El resumen de este suceso obtenido por nuestro método se muestra en la figura 6.12.

Como puede verse, este resumen ofrece una visión concisa de los principales aspectos del asesinato de Paco Stanley. Este resumen tiene 139 palabras, en contraste con las 2320 palabras en total que tienen todos los documentos de este grupo, lo que representa un 94% de compresión.

Paco Stanley presentaba populares programas de variedades en Televisión Azteca, y su muerte ha causado honda conmoción en la sociedad mexicana, en la que era un personaje muy querido. Porfirio Muñoz Ledo, contrincante del alcalde en la lucha interna del PRD por la nominación presidencial del partido en las presidenciales del 2000, aventuró nuevos móviles en el asesinato del presentador y cómico cuando declaró que 'sería inmoral' que con la sonada muerte se pretendiera desestabilizar el proceso previo a esa cita con las urnas, en la que Cárdenas aspira a la presidencia de la República. Durante horas y horas, explicó el comentarista Raúl Trejo, lo único que se vio en las pantallas de televisión mexicanas, tras el asesinato de Paco Stanley, fue un desfile de lamentos, quejas y exigencias que nada aclaraban, pero se transformaron en un ariete.

Fig. 6.12.- Resumen del asesinato de Paco Stanley.

La figura 6.13 muestra los resultados del porcentaje de compresión en cada suceso con respecto a su tamaño (número total de palabras en los documentos del suceso). Como podemos ver, existe una tendencia de que mientras mayor es el tamaño del suceso, mayor es el porcentaje de compresión.

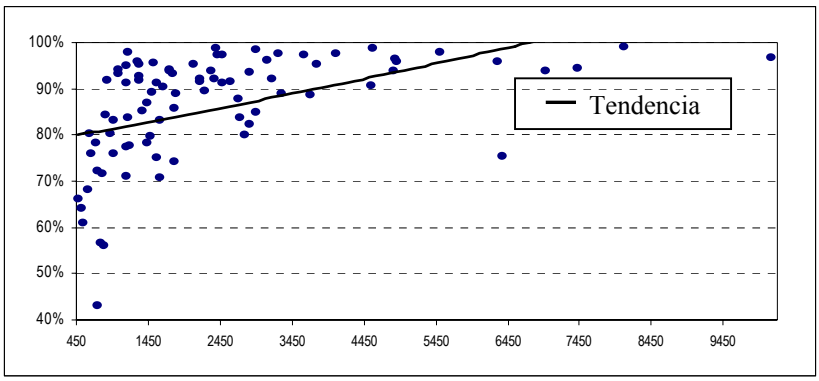


Fig. 6.13.- Compresión con respecto al número total de palabras en cada suceso.

Para evaluar la efectividad del método de construcción automática de resúmenes utilizamos además la colección TREC-5, la cual como dijimos anteriormente está clasificada en 23 tópicos. La tabla 6.11 muestra para cada tópico, su tamaño (número de documentos), el tamaño del conjunto unión de los testores típicos (número de términos) y el porcentaje de compresión obtenido con el resumen.

Tópico	Tamaño	TT	%	Tópico	Tamaño	TT	%	Tópico	Tamaño	TT	%
SP51	83	4	96.81	SP60	47	7	97.94	SP69	62	4	98.06
SP52	13	3	98.20	SP62	15	6	85.72	SP70	6	3	74.76
SP53	46	2	99.36	SP63	5	4	83.43	SP71	17	3	99.29
SP54	37	8	94.38	SP64	9	4	83.54	SP72	14	7	91.05
SP55	108	4	99.29	SP65	29	5	96.40	SP73	13	5	93.48
SP57	2	3	82.05	SP66	68	4	97.32	SP74	34	10	97.65
SP58	49	9	97.91	SP67	13	9	87.20	SP75	20	3	94.09
SP59	7	6	93.74	SP68	12	8	93.78				

Tabla 6.11.- Resultados obtenidos en la colección TREC-5.

Nuevamente, los resúmenes obtenidos captan las ideas principales de cada tópico y se logra una apreciable reducción en el número de palabras del resumen. Por ejemplo, la

descripción dada por TREC al tópico SP68 es “La situación del SIDA en Argentina y qué pasos está tomando el gobierno argentino para combatir esta enfermedad”. La unión de los testores típicos encontrados es: {Argentina, campaña, prevención, enfermedad, población, droga, caso, persona}. El resumen obtenido para este tópico se muestra en la figura 6.14.

> La asociación de las drogas DDI e hidroxamatos contra el SIDA podría llegar a ser "la única capaz de eliminar el virus en personas seropositivas" manifestó este domingo el médico argentino Julio Vila, que dirige un equipo de investigación en Francia. El gobierno argentino lanzará una campaña educativa de la población para prevenir el Síndrome de Inmunodeficiencia Adquirida (Sida), informaron este viernes autoridades sanitarias. El ministro de Salud y Acción Social, Alberto Mazza, había anunciado el viernes que el gobierno lanzará una campaña educativa, cuyo objetivo será "instruir a la población sobre las medidas de prevención que deben adoptarse para combatir el Sida". La campaña de educación preventiva será puesta en marcha después que se conozcan los resultados de una encuesta nacional sobre el grado de conocimiento que la población tiene sobre la enfermedad, había indicado Mazza. Los casos, aunque detectados en los últimos años, sólo fueron conocidos ahora, indicándose que la situación dio lugar a una campaña de prevención ordenada por el jefe del ejército, general Martín Balza, que presume estar ante "un problema grave pero controlable". Asimismo, en el Hospital Militar Central se creó el Centro de Inmunodeficiencia del Ejército (CEIDE), una institución que se encargará de llevar adelante las campañas de prevención, el tratamiento y seguimiento de los casos de Sida.

Fig. 6.14.- Resumen del tópico SP68 de TREC-5.

En realidad, es difícil evaluar la calidad de un método de construcción automática de resúmenes. No obstante, consideramos que los resúmenes obtenidos por nuestro método son legibles, coherentes y captan los temas fundamentales del conjunto de documentos. Por lo tanto, consideramos que estos resúmenes pueden ser presentados a un usuario como una descripción significativa del contenido de un tópico.

6.10 CONCLUSIONES

A lo largo de este capítulo hemos evaluado la calidad del sistema de detección propuesto. En particular, hemos analizado la utilidad de las diferentes componentes de representación de las noticias y de las diferentes variantes de función de semejanza que pueden emplearse. Hemos evaluado, además, las temáticas y los resúmenes obtenidos. Para ello, hemos utilizado tres colecciones de prueba: El País, TDT2 y TREC-5.

Nuestros experimentos han demostrado el positivo impacto de las componentes temporal y espacial en la calidad de los grupos generados por el sistema. Hemos comprobado que su utilidad es mayor en el nivel de sucesos que en el nivel de tópicos. En el caso, de la componente temporal concluimos que es mejor utilizar la fecha de publicación que invertir recursos extrayendo las referencias temporales de los textos.

Como hemos visto, los mejores resultados se obtienen cuando utilizamos la representación a nivel de término de las noticias y la medida de semejanza que contempla las componentes semántica, temporal y espacial. Los valores de la medida F1 y el coste de detección obtenidos en las colecciones de prueba utilizadas demuestran la validez de nuestro algoritmo para las tareas de detección de tópicos. Nuestro sistema de detección no sólo obtiene resultados superiores o similares a los sistemas existentes sino

que también brinda una información adicional: la jerarquía de tópicos con diferentes niveles de granularidad.

Al utilizar la función de semejanza propuesta, podemos optimizar el algoritmo compacto incremental y con ello, permitir la manipulación de grandes volúmenes de noticias al tener un coste lineal.

Los resultados obtenidos en la clasificación temática de cada uno de los tópicos generados por nuestro sistema de detección, así como la calidad de los resúmenes construidos brindan una ayuda inapreciable a los analistas de información en la interpretación de los tópicos generados.

Aunque no se obtuvieron mejores resultados, hemos dejado la puerta abierta para la investigación de métodos de representación conceptual de las noticias que permitan la detección de sucesos en colecciones multilingües.

7 CONCLUSIONES

En este trabajo se han desarrollado un conjunto de técnicas para el procesamiento dinámico de la información que incluyen algoritmos incrementales de agrupamiento y métodos de descripción de conjuntos de documentos. Se ha propuesto, además, un sistema de detección de tópicos que permite la construcción de una jerarquía de tópicos con diferentes niveles de abstracción y la descripción de cada uno de los tópicos obtenidos. A pesar de haber aplicado las técnicas desarrolladas a la detección de tópicos, su uso no está restringido a esta área.

A continuación se presentan las principales aportaciones de esta tesis, así como las posibles líneas de trabajo futuro y la producción científica obtenida en la realización de este trabajo.

7.1. APORTACIONES

Las principales aportaciones de esta tesis han sido:

- Un compendio claro y conciso de los principales sistemas de detección de tópicos existentes: CMU, UMASS, IBM, Dragón, UPENN, IOWA, BBN, TNO y los sistemas de Papka, Brants y Kurt que puede servir de literatura para aquellos que deseen iniciarse en esta temática.
- Un estudio comparativo de los sistemas de detección estudiados atendiendo al modelo de representación de documentos utilizado, la medida de semejanza, el algoritmo de agrupamiento y la forma en que manipulan las propiedades temporales de las noticias.
- Un análisis crítico de cada uno de los sistemas de detección estudiados, lo que permitió identificar las principales limitaciones existentes, las cuales son:
  - Dependencia de los tópicos detectados al orden de presentación de las noticias. Esta dependencia trae como consecuencia que los tópicos detectados pueden ser diferentes.
  - 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.
  - No tienen en cuenta las propiedades temporales de los documentos en el cálculo de la semejanza entre ellos o las propiedades temporales de los documentos que manipulan sólo están restringidas a su posición en la ventana temporal.
  - 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.

- Un nuevo algoritmo de agrupamiento no supervisado e incremental que permite la obtención de particiones, es decir, grupos disjuntos. Este algoritmo construye de forma incremental los conjuntos compactos y está basado en un conjunto de propiedades matemáticas que son demostradas en el trabajo.
- Un nuevo algoritmo de agrupamiento no supervisado e incremental que permite la obtención de cubrimientos, es decir, grupos solapados. Este algoritmo construye de forma incremental los conjuntos fuertemente compactos. De forma similar al algoritmo anterior, las propiedades matemáticas sobre las que se basa se han demostrado en el trabajo.
- Un nuevo algoritmo de agrupamiento no supervisado, jerárquico e incremental que crea jerarquías de grupos con diferentes niveles de abstracción. Para la obtención de los grupos en cada nivel de la jerarquía pudieran emplearse los algoritmos compacto y fuertemente compacto incrementales.

  Los tres algoritmos incrementales propuestos presentan las ventajas siguientes:

  - Permiten agrupar objetos de cualquier naturaleza descritos por rasgos métricos, espaciales, cualitativos y categóricos mezclados, incluso con ausencia de información. Su uso no está restringido al agrupamiento de documentos.
  - Sobre el espacio de representación de los objetos no se supone ninguna estructura algebraica o topológica, ni ninguna distancia (métrica) definida a priori.
  - El agrupamiento obtenido por los algoritmos propuestos es único, es decir, no depende del orden de presentación de los objetos. Esto constituía una limitación de la casi totalidad de los algoritmos incrementales existentes en la actualidad.
  - No necesitan fijar a priori el número de grupos a obtener.
  - Permiten encontrar grupos con formas arbitrarias en oposición a los algoritmos como *K-Means Incremental* y *Single-Pass* que utilizan medidas centrales para generar los grupos, y como consecuencia, se restringe a los grupos a tener formas esféricas o elipsoidales.
  - No necesitan almacenar la matriz de semejanzas completa (de orden n x n) de la colección de objetos.
  - La complejidad computacional de las variantes incrementales sigue siendo la misma que la de las variantes no incrementales, pero muy inferior a la aplicación reiterada de los respectivos algoritmos no incrementales.
- Un nuevo algoritmo de escala exterior para el cálculo de todos los testores típicos de una matriz de aprendizaje (*LEX*). Este algoritmo tiene significativamente mejor desempeño que los algoritmos existentes para el cálculo de los testores típicos. Su utilización no está restringida a colecciones de documentos, sino que puede emplearse para la selección de variables en cualquier problema del Reconocimiento de Patrones con clases disjuntas.

- Un método efectivo para construir resúmenes a partir de una partición en grupos de una colección de documentos. El método propuesto emplea el cálculo de los testores típicos como su operación primaria y, a partir de ellos, construye el resumen de cada grupo. A diferencia de los otros métodos propuestos en la literatura, la selección de las oraciones se basa en los términos frecuentes y discriminantes de cada grupo, los cuales pueden ser eficientemente calculados con el algoritmo *LEX* aquí presentado. Este método es robusto, independiente del dominio y puede ser aplicado a documentos escritos en cualquier idioma.
- Un sistema de detección de tópicos, denominado *JERARTOP*, que va más allá de un sistema tradicional de detección, ya que extrae o infiere de forma completamente automática e incremental el conocimiento implícito en el flujo de documentos. Este sistema se caracteriza por:
  - Un método de representación de las noticias que incorpora no sólo las propiedades semánticas (como hacían los sistemas actuales), sino también las propiedades espaciales y temporales. Los experimentos realizados demuestran la utilidad de considerar estas tres componentes pues elevan la calidad del sistema de detección.
  - La definición de una medida de semejanza para comparar a las noticias que contempla sus propiedades semánticas, temporales y espaciales. Con este objetivo, definimos medidas de semejanza parciales para cada una de las componentes de la representación de las noticias y una medida de semejanza global que expresa que dos noticias abordan el mismo tópico si tienen una alta semejanza semántica, proximidad temporal y coincidencia en lugar. Esta medida se construyó sobre la base de filtros y no como se realiza usualmente a partir de una combinación lineal de las semejanzas parciales, que es conceptualmente incorrecto.
  - El sistema no se limita a encontrar un conjunto de tópicos, sino que construye una jerarquía de sucesos y tópicos con diferentes niveles de granularidad. En el primer nivel de la jerarquía se identifican los sucesos individuales. En los siguientes niveles estos sucesos son sucesivamente agrupados para formar sucesos más complejos y tópicos. Esta jerarquía obtenida permite a un analista de información identificar no sólo los tópicos sino además los sucesos en los que ellos se pueden descomponer. Para la construcción de esta jerarquía se utiliza el algoritmo jerárquico incremental que emplea como rutina de agrupamiento el algoritmo compacto incremental, propuestos también en este trabajo.
  - A diferencia de los sistemas de detección propuestos en la literatura, esta aproximación evita la definición de qué es un tópico, ya que es el usuario el que navegando por la jerarquía creada decide en qué nivel explora el conocimiento de su interés.
  - Con el objetivo de facilitar el análisis de los tópicos detectados, el sistema de detección propuesto categoriza, además, en un conjunto de temáticas pre-definidas a cada tópico de la jerarquía creada y proporciona un resumen de los contenidos asociados en cada uno de ellos. Se utilizan como categorías los tópicos *IPTC* y para la construcción de los resúmenes se emplea el método propuesto en este trabajo.

  - No presenta ninguna de las limitaciones señaladas en el análisis crítico de los sistemas de detección existentes.
  - La experimentación realizada en el capítulo 6 permite constatar a partir de la comparación realizada con los sistemas de detección existentes, las posibilidades que brinda *JERARTOP* como nuevo sistema de detección de tópicos.

- Un nuevo algoritmo basado en el conocimiento para la desambiguación del sentido de las palabras que desambigua tanto sustantivos, como adjetivos y verbos, utiliza todas las relaciones semánticas existentes en *WordNet* y aprovecha los dominios de Magnini, los tópicos IPTC y los glosarios de los sentidos de las palabras para desambiguar las palabras polisémicas. Este método puede ser aplicado a cualquier dominio de problemas sin requerir colecciones etiquetadas manualmente ni ningún proceso de entrenamiento y es capaz de desambiguar textos en cualquier idioma siempre que tenga un *WordNet* definido. El algoritmo de desambiguación propuesto no sólo se utiliza para obtener la representación conceptual de los documentos sino también para la obtención de las temáticas (tópicos IPTC) abordadas por los grupos detectados por el sistema *JERARTOP*.

7.2. TRABAJO FUTURO

Como resultado de este trabajo proponemos que se continúe esta investigación en las siguientes direcciones:

- Profundizar en el estudio del impacto de la representación conceptual y a nivel de frase de las noticias, perfeccionando los métodos de construcción de estos dos tipos de representación y evaluando el impacto que tendrían en la calidad del sistema de detección.
- Profundizar en el estudio del método de construcción automática de resúmenes propuesto y evaluar su viabilidad para resumir documentos extensos que aborden cualquier temática.
- Mejorar la precisión y la relevancia del algoritmo de desambiguación propuesto. Para ello, es preciso introducir más heurísticas que combinen diferentes técnicas y recursos para producir todos juntos una mejor desambiguación del sentido de las palabras.
- Extensión de los algoritmos de agrupamiento incrementales desarrollados al caso difuso.
- El desarrollo de las variantes paralelas de los algoritmos de agrupamiento propuestos para permitir la manipulación de grandes volúmenes de datos.
- Crear una aplicación real de detección de tópicos con las técnicas desarrolladas en esta tesis.

7.3. PRODUCCIÓN CIENTÍFICA

En el marco del desarrollo de esta tesis se han realizado las siguientes publicaciones:

- Pons-Porrata, A.; Berlanga-Llavori, R. and Ruiz-Shulcloper, J. "On-line event and topic detection by using the compact sets clustering algorithm". *Journal of Intelligent and Fuzzy Systems*, Numbers 3-4, pp. 185-194, 2002.
- Pons-Porrata, A.; Berlanga-Llavori, R. and Ruiz-Shulcloper, J. "Detecting events and topic by using temporal references". *Lecture Notes in Artificial Intelligence 2527. Subseries of Lectures Notes in Computer Science*, Francisco J. Garijo, José C. Riquelme, Miguel Toro (Eds.), Springer-Verlag, pp.11-20, November 2002.
- 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.
- Pons-Porrata, A.; Ruiz-Shulcloper, J. and Berlanga-Llavori, R. "A method for the automatic summarization of topic-based clusters of documents". *Progress in Pattern Recognition, Speech and Image Analysis, Lecture Notes on Computer Sciences 2905*, Springer Verlag, Alberto Sanfeliu, José Ruiz Shulcloper (Eds.), pp. 596-603, Noviembre de 2003.
- 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.
- Santiesteban-Alganza, Y. y Pons-Porrata, A. "LEX: un nuevo algoritmo para el cálculo de los testores típicos". *Revista Ciencias Matemáticas*, Vol. 21, No. 1, 2003.
- Pons-Porrata, A.; Ruiz-Shulcloper, J.; Berlanga-Llavori, R. y Santiesteban-Alganza, Y. "Un algoritmo incremental para la obtención de particiones con datos mezclados". *Reconocimiento de Patrones. Avances y Perspectivas. Research on Computing Science, CIARP'2002*, pág. 265-276. Editores JL Díaz de León Santiago, C. Yánez Márquez, México D.F., Noviembre 19-22, 2002.
- Pons-Porrata, A.; Ruiz-Shulcloper, J.; Berlanga-Llavori, R. y Santiesteban-Alganza, Y. "Un algoritmo incremental para la obtención de cubrimientos con datos mezclados". *Reconocimiento de Patrones. Avances y Perspectivas. Research on Computing Science, CIARP'2002*, pág. 405-416. Editores JL Díaz de León Santiago, C. Yánez Márquez, México D.F., Noviembre 19-22, 2002.
- Pons, A.; Berlanga, R. y Llidó, D. M. "Técnicas de agrupamiento semántico-temporal para la identificación de sucesos en bases de noticias digitales". *Proceeding of IX Conferencia de la Asociación Española para la Inteligencia Artificial*, *CAEPIA'01*, Universidad de Oviedo, Gijón, España, pp.603-612, 2001.
- Pons-Porrata, A.; Berlanga-Llavori, R. y Ruiz-Shulcloper, J. "Un nuevo método de desambiguación del sentido de las palabras usando *WordNet*". *Proceedings de la X Conferencia de la Asociación Española para la Inteligencia Artificial CAEPIA 2003*, Volumen II, pág. 63-66, noviembre de 2003.

- Pons Porrata, A. y Santiesteban Alganza, Y. “Algoritmos Incrementales de Agrupamiento de Documentos”. *Proceedings of 2nd International Conference on Automatic Control AUT’2002*, Santiago de Cuba, Julio de 2002.
- Pons, A. y Berlanga, R. “Métodos y Algoritmos para la Agrupación Automática de Documentos”. *Informe Técnico, Universitat Jaume I*, España, DLSI 01/06/2001, 2001.

8 REFERENCIAS

[Agir96] Agirre, E. and Rigau, G. "Word Sense Disambiguation using Conceptual density". In *Proceedings of the 16th International Conference on Computational Linguistic* (COLING'96), pp. 16-22, Copenhagen, Denmark, 1996.

[Agui84] Águila, L. y Shulcloper, J. R. "Algoritmo CC para la elaboración de la información k-valente en problemas de Reconocimiento de Patrones". *Revista Ciencias Matemáticas*, Vol. 5, No. 3, 1984.

[AhoH83] Aho, A. V.; Hopcroft, J. E. and Ullman, J. D. "*Data Structures and Algorithms*". Addison-Wesley Publishing Company, 1983.

[Aize71] Aizenberg, N. N. y Tsipkin, A. I. "Testores primos". *Doklady Akademii Nauk SSSR* 201, pág. 801-802, 1971 (en ruso).

[Alba98] Alba Cabrera, E. y Lazo Cortés, M. "Una solución global para la utilización de los testores en problemas de reconocimiento de patrones". *Memorias del III Taller Iberoamericano de Reconocimiento de Patrones*, México, pp. 209-217, 1998.

[Alla98] 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.

[Alla00] 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.

[Alla00a] Allan, J.; Lavrenko, V. ; Frey, D. and Khandelwal, V. "UMASS at TDT 2000". *Proceedings TDT 2000 Workshop*, 2000.

[Alla00b] Allan, J.; Lavrenko, V. and Jin, H. "Comparing Effectiveness in TDT and IR". *CIIR Technical Report*, 2000.

[Andr81] Andreev, A. E. "On irredundant and minimal tests", *Soviet Mathematics Doklady*, Vol. 23, No. 1, pág. 92-96, 1981.

[Asla98] Aslam, J.; Pelekhov, K. and Rus, D. "Static and Dynamic Information Organization with Star Clusters". *Proceedings of the 1998 Conference on Information Knowledge Management*, CIKM'98, Baltimore, MD, 1998.

[Ayaq97] Ayaquica, I.O. "Un nuevo algoritmo de escala exterior para el cálculo de testores típicos". *Memorias del II Taller Iberoamericano de Reconocimiento de Patrones*, La Habana, pág. 141-148, 1997.

[Barz02] Barzilay, R.; Elhadad, N. and McKeown, K. "Inferring Strategies for Sentence Ordering in Multidocument News Summarization". *Journal of Artificial Intelligence Research* 17, pp. 35-55, 2002.

[Berg96] Berger, A.; Della Pietra, S. and Della Pietra, V. "A Maximum Entropy Approach to Natural Language Processing". *Computational Linguistics*, vol. 22 (1), 1996.

[Bran03] 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.

[Brav83] Bravo, A. "Algoritmo CT para el cálculo de los testores típicos de una matriz k-valente". *Revista Ciencias Matemáticas*, Vol. 4, No. 2, pág. 123-144, 1983.

[Call96] Callan, J. P. "Document Filtering with Inference Networks". *Proceedings of ACM SIGIR*, pp. 262-269, 1996.

[Carb99] 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.

[Carm98] Carmona, J.; Cervell, S.; Márquez, L.; Martí, M.A.; Padró, L.; Rodríguez, H.; Placer, R.; Taulé, M. and Turmo, J. "An Environment for Morphosyntactic Processing of Unrestricted Spanish text". In *Proceedings of the First International Conference on Language Resources and Evaluation*, LREC'98, 1998.

[Cheg55] Cheguis, I. A. y Yablonskii, S. V. "Acerca de los tests para esquemas eléctricos". *Uspieji matematiceskij Nauk*, tom 4, #10*,* pp. 182-184, Moscú, 1955 (en ruso).

[Cier99] Cieri, C.; Graff, D.; Liberman, M.; Martey, N. and Strassel, S. "The TDT-2 Text and Speech Corpus". *Proceedings DARPA Broadcast News Workshop*, pp. 57-60, 1999.

[Cier00] Cieri, C.; Graff, D.; Liberman, M.; Martey, N. and Strassel, S. "Large, Multilingual, Broadcast News Corpora for Cooperative Research in Topic Detection and Tracking: The TDT-2 and TDT-3 Corpus Efforts". *Proceedings of Language Resources and Evaluation Conference*, Athens, Greece, May 2000.

[Cutt92] 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.

[Cutt93] 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.

[Dhar99] Dharanipragada, S.; Franz, M.; McCarley, J.; Roukos, S. and Ward, T. "IBM Slides Presentation" by J.S. McCarley at *The DARPA Broadcast News Workshop*, Herndon, VA, 1999.

[Dhar99a] Dharanipragada, S.; Franz, M.; McCarley, J.; Roukos, S. and Ward, T. "Story Segmentation and Topic Detection in the Broadcast News Domain". *Proceedings of the DARPA Broadcast News Workshop*, Feb.-March, Herndon, VA, 1999.

[Dmit66] Dmitriev, A. N.; Zhuravlev, J. I. y Krendeleiev, F. P. "Acerca de los principios matemáticos de la clasificación de objetos y fenómenos". *Diskretnyi Analiz* 7, pág. 3-15, 1966 (en ruso).

[Dodd99] Doddington, G. "Presentation Slides". *The DARPA Broadcast News Workshop*, Herndon, VA, 1999.

[Dodd00] Doddington, G. and Fiscus, J. "Topic Detection and Tracking. TDT3 Detection Evaluation Results". *Sheraton Premiere at Tysons Corner*. Vienna, Virginia, USA, 2000.

[Eich99] 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.

[Escu00] Escudero, G.; Márquez, L. and Rigau, G. "An empirical study of the domain dependence of supervised word sense disambiguation systems". In *Proceedings of Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora* (EMNLP/VLC'00), Hong Kong, China, 2000.

[Fern01] Fernández-Amorós, D.; Gonzalo, J. and Verdejo, F. "The Role of Conceptual Relations in Word Sense Disambiguation". In *Proceedings of the 6th International Workshop on Applications of Natural Language for Information Systems*, 2001.

[Fisc99] Fiscus, J.; Doddington, G.; Garofolo, J. and Martin, A. "NIST'S 1998 Topic detection and tracking evaluation (TDT2)". *Proceedings of the 1999 DARPA Broadcast News Workshop*, pp. 19-24, Herdorn, Virginia, Feb.-March, 1999.

[Gill90] Gillick, L. "Probabilistic Models for Topic Spotting". *Workshop Notebook for the WHISPER Meeting at MIT Lincoln Laboratory*, July 25-26, pp. 206-211, 1990.

[Gold80] Goldman, R. S. "Problemas de la teoría de los test difusos". *Automática y Telemejánika*, No. 10, pág. 146-153, 1980 (en ruso).

[Gold00] Goldstein, J.; Mittal, V.; Kantrowitz, M. and Carbonell, J. "Multi-Document Summarization by Sentence Extraction, Automatic Summarization". *Proceedings of the ANLP/NAACL Workshop*, Seattle, WA, April 2000.

[Gonz99] Gonzalo, J.; Peñas, A. and Verdejo, F. "Lexical ambiguity and Information

Retrieval Revisited". In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora* (EMNLP/VLC-99), MaryLand, 1999.

[Graf00] Graff, D.; Cieri, C.; Strassel, S. and Martey, N. "The TDT-3 Text and Speech Corpus". *Topic Detection and Tracking Workshop*, Vienna, Virginia, February 28 - March 1, 2000.

[Gree00] Greengrass, Ed. "*Information Retrieval: a Survey*". November, 2000.

[Hill68] Hill, D. R. "*A vector clustering technique*". In Samuelson (ed.), Mechanized Information Storage, Retrieval and Dissemination, North-Holland, Amsterdam, 1968.

[Hira03] Hirao, T; Susuki, J.; Isozaki, H. and Maeda, E. "NTT's Multiple Document Summarization System for DUC2003". *Workshop on Text Summarization*, May 31-June 1, Edmonton, Canada, 2003.

[Horo75] Horowitz, E. and Sahni, S. "*Fundamentals of Data Structures*". Computer Science Press, Woodland Hills, California, 1975.

[Kake87] Kakes, A. y Casas, O. "*Teoría de grafos*". Editorial Pueblo y Educación, 1987.

[Kilg98] Kilgarriff, A. "Gold standard for evaluating word sense disambiguation programs". *Computer Speech and Language, Special Issue on evaluation*, 12(3), 1998.

[Kost03] Koster, Cornelis H.A. and Seutter, Marc. "Taming Wild Phrases". *Proceedings 25th European Conference on IR Research* ECIR 2003, Springer LNCS 2633, pp 161-176, 2003.

[Kurt01] 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.

[Lars99] 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.

[Lazo99] Lazo Cortés, M.; Sánchez Díaz, G. y Quintana Gómez, T. "Evaluación de la relevancia de los rasgos en un problema de clasificación supervisada a partir de todos los testores". *Memorias del IV Simposio Iberoamericano de Reconocimiento de Patrones SIARP'99*, Ciudad Habana, pág. 215-222, 1999.

[Lazo01] Lazo-Cortés, M.; Ruiz-Shulcloper, J. and Alba-Cabrera, E. "An overview of the concept testor". *Pattern Recognition*, Vol. 34, Issue 4, pp.13-21, 2001.

[Leek00] 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.

[Lesk86] Lesk, M. "Automated sense disambiguation using machine-readable dictionaries: How to tell a pine cone from an ice cream cone". In *Proceedings of SIGDOC Conference, Association for Computing Machinery*,

pp.24-26, Canada, 1986.

[Llid01] Llidó, D.; Berlanga, R. and Aramburu, M.J. "Extracting temporal references to automatically assign document event-time periods". In *Proc. Database and Expert System Applications 2001*, 62-71, Springer-Verlag, Munich, 2001.

[Llid02] Llidó Escrivá, D. M. "*Extracción y Recuperación de Información Temporal*". Tesis doctoral, Universitat Jaume I, 2002.

[Lowe99] 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.

[Magn00] Magnini, B. and Cavaglia, G. "Integrating subject field codes into WordNet". In *Proceedings of the LREC-2000*, Second International Conference on Language Resources and Evaluation, Athens, Greece, pp.1413-1418, 2000.

[Magn01] Magnini, B.; Strapparava, C.; Pezzulo, G. and Gliozzo, A. "Using Domain Information for Word Sense Disambiguation". In *Proceeding of SENSEVAL-2, Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pp.111-114, Toulouse, France, 5-6 July 2001.

[Magn02] Magnini,B.; Strapparava, C.; Pezzulo, G. and Gliozzo, A. "The Role of Domain Information in Word Sense Disambiguation". In *Evaluating Word Sense Disambiguation Systems. Special Issue of the Journal of Natural Language Engineering*, 2002.

[Mani97] Mani, I. and Bloedorn, E. "Multi-Document Summarization by Graph Search and Matching". *AAAI/IAAI* 1997, pp. 622-628, 1997.

[Mani01] Mani, I. "*Automatic Summarisation*". John Benjamins Publishing Company, 2001.

[Marc01] Marcu, D. "Discourse-based summarisation in DUC-2001". *Proceedings of Document Understanding Conference*, DUC-2001, 2001.

[Mart97] Martin, A.; Doddington, G.; Kamm, T.; Ordowski, M. and Przybocki, M. "The DET Curve in assessment of Detection Task Performance". *Proceedings of EuroSpeech 1997*, Volume 4, pp. 1895-1898, European Speech Communication Association (ESCA), 1997.

[Mart00] Martínez-Trinidad, J. F.; Ruiz-Shulcloper, J. and Lazo-Cortés, M. "Structuralization of Universes". *Fuzzy Sets and Systems*, vol. 112 (3), pp. 485-500, 2000.

[Mill90] Miller, G. "Five papers on WordNet". *Special Issue of International Journal of Lexicography* 3(4), 1990.

[Mont01] Montoyo, A. and Palomar, M. "Specifications Marks for Word Sense Disambiguation: New Development". In *Lecture Notes in Computer Science* vol. 2004:182-191, Springer Verlag. 2nd International Conference on Intelligent Text Processing and Computational Linguistics CICLing-2001,

México, 2001.

[Murt83] Murtagh, F. "A survey of recent advances in hierarchical clustering algorithms". *Computer Journal*, 26:354-359, 1983.

[Papk98] Papka, R. and Allan, J. "On line new event detection using Single Pass Clustering". *UMASS Computer Science Technical Report*, pp. 98-21, 1998.

[Papk99] Papka, R. "*On-line New Event Detection, Clustering and Tracking*". Ph.D. thesis, University of Massachusetts, Department of Computer Science, September 1999.

[Papk99a] Papka, R.; Allan, J. and Lavrenko, V. "UMASS Approaches to Detection and Tracking at TDT2". *Proceedings of the DARPA Broadcast News Workshop*, Herndon, VA, pp. 111-116, 1999.

[Pons01] Pons, A. y Berlanga, R. "Métodos y Algoritmos para la Agrupación Automática de Documentos". *Informe Técnico, Universitat Jaume I*, España, DLSI 01/06/2001, 2001.

[Pons01a] Pons, A.; Berlanga, R. y Llidó, D. M. "Técnicas de agrupamiento semántico-temporal para la identificación de sucesos en bases de noticias digitales". *Proceeding of IX Conferencia de la Asociación Española para la Inteligencia Artificial*, *CAEPIA'01*, Universidad de Oviedo, Gijón, España, pp.603-612, 2001.

[Pons02] 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.

[Pons02a] Pons Porrata, A. y Santiesteban Alganza, Y. "Algoritmos Incrementales de Agrupamiento de Documentos". *Proceedings of 2nd International Conference on Automatic Control AUT'2002*, Santiago de Cuba, Julio de 2002.

[Pons02b] Pons-Porrata, A.; Berlanga-Llavori, R. and Ruiz-Shulcloper, J. "Detecting events and topic by using temporal references". *Lecture Notes in Artificial Intelligence 2527. Subseries of Lectures Notes in Computer Science*, Francisco J. Garijo, José C. Riquelme, Miguel Toro (Eds.), Springer-Verlag, pp.11-20, November 2002.

[Pons02c] Pons-Porrata, A.; Ruiz-Shulcloper, J.; Berlanga-Llavori, R. y Santiesteban-Alganza, Y. "Un algoritmo incremental para la obtención de cubrimientos con datos mezclados". *Reconocimiento de Patrones. Avances y Perspectivas. Research on Computing Science, CIARP'2002*, pág. 405-416. Editores JL Díaz de León Santiago, C. Yánez Márquez, México D.F., Noviembre 19-22, 2002.

[Pons02d] Pons-Porrata, A.; Ruiz-Shulcloper, J.; Berlanga-Llavori, R. y Santiesteban-Alganza, Y. "Un algoritmo incremental para la obtención de particiones con datos mezclados". *Reconocimiento de Patrones. Avances y Perspectivas. Research on Computing Science, CIARP'2002*, pág. 265-276. Editores JL

Díaz de León Santiago, C. Yánez Márquez, México D.F., Noviembre 19-22, 2002.

[Pons02e] Pons-Porrata, A.; Berlanga-Llavori, R. and Ruiz-Shulcloper, J. "On-line event and topic detection by using the compact sets clustering algorithm". *Journal of Intelligent and Fuzzy Systems*, Numbers 3-4, pp. 185-194, 2002.

[Pons03] 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.

[Pons03a] Pons-Porrata, A.; Berlanga-Llavori, R. y Ruiz-Shulcloper, J. "Un nuevo método de desambiguación del sentido de las palabras usando WordNet". *Proceedings de la X Conferencia de la Asociación Española para la Inteligencia Artificial CAEPIA 2003*, Volumen II, pág. 63-66, noviembre de 2003.

[Pons03b] Pons-Porrata, A.; Ruiz-Shulcloper, J. and Berlanga-Llavori, R. "A method for the automatic summarization of topic-based clusters of documents". *Progress in Pattern Recognition, Speech and Image Analysis, Lecture Notes on Computer Sciences 2905*, Springer Verlag, Alberto Sanfeliu, José Ruiz Shulcloper (Eds.), pp. 596-603, Noviembre de 2003.

[Port80] Porter, M.F. "An Algorithm for Suffix Stripping". *Program*, v.14, No. 3, pp. 130-137, 1980.

[Ragh86] 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.

[Rijs79] van Rijsbergen, C. J. "*Information Retrieval*". Butterworths, London, second edition, 1979.

[Robe95] 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.

[Salt89] Salton, G. "*Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*", Addison-Wesley, 1989.

[Sanc97] Sánchez Díaz, G. "*Desarrollo y Programación de algoritmos eficientes (secuenciales y paralelos) para el cálculo de los testores típicos de una matriz básica*". Tesis de Maestría en Ciencias de la Computación, BUAP, Puebla, México, 1997.

[Sanc99] Sánchez Díaz, G.; Lazo Cortés, M. y Fuentes Chávez, O. "Algoritmo genético para calcular testores típicos de costo mínimo". *Memorias del IV Simposio Iberoamericano de Reconocimiento de Patrones SIARP'99*, Ciudad Habana, pág. 207-213, 1999.

[Sanc01] Sánchez-Díaz, G. "*Desarrollo de algoritmos para el agrupamiento de grandes volúmenes de datos mezclados*". Tesis doctoral, Centro de

Investigación en Computación, México, 2001.

[Sant03] Santiesteban-Alganza, Y. y Pons-Porrata, A. "LEX: un nuevo algoritmo para el cálculo de los testores típicos". *Revista Ciencias Matemáticas*, Vol. 21, No. 1, 2003.

[Schu99] 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.

[Shul82] Shulcloper, J. R.; Bravo, M. A. y Águila, F. L. "Algoritmos BT y TB para el cálculo de todos los tests típicos". *Revista Ciencias Matemáticas*, Vol. 6, No. 2, 1982.

[Shul91] Shulcloper, J. R. y Lazo Cortés, M. "*K*-testores primos", *Ciencias Técnicas, Físicas y Matemáticas* 9, pág.17-55, 1991.

[Shul95] Shulcloper, J.R.; Alba, E. y Lazo, M. "*Introducción al Reconocimiento de Patrones*". Grupo de Reconocimiento de Patrones Cuba-México, Centro de Investigación y Estudios Avanzados del I.P.N., México, Serie Verde No. 51, 1995.

[Shul95a] Shulcloper, J. R.; Alba Cabrera, E. y Lazo Cortés, M. "*Introducción a la teoría de Testores Típicos*". Serie Verde No. 50, I.P.N., México, 1995.

[Shul99] Shulcloper, J. R. and Lazo, M. "Mathematical algorithms for the supervised classification based on fuzzy partial precedence". *Mathematical and Computer Modelling*, Vol. 29, pp. 111-119, 1999.

[Spitt01] Spitters, M. and Kraaij, W. "TNO at TDT2001: Language Model-Based Topic Detection". *Topic Detection and Tracking (TDT) Workshop*, Gaithersburg, USA, November 2001.

[Suss93] Sussna, M. "Word Sense Disambiguation for Free-text indexing using a massive semantic network". In *Proceedings of the Second International Conference on Information and Knowledge Management*, Arlington, Virginia USA, 1993.

[Take94] Takenobu, Tokunaga and Makoto, Iwayama. "Text Categorization based on weighted inverse document frequency". *Technical Report 94-TR0001*, Department of Computer Sciences, Tokyo Institute of Technology, March 1994.

[TDT298] National Institute of Standards and Technology. "*The Topic Detection and Tracking Phase 2 (TDT2)*", evaluation plan, version 3.7, 1998.

[TDT03] National Institute of Standards and Technology. "*The 2003 Topic Detection and Tracking (TDT2003). Task definition and Evaluation Plan*", version 1.0, 2003.

[Vale91] Valev, V. and Zhuravlev, Y. J. "Integer-valued problems of transforming the training tables in k-valued code in pattern recognition problems". *Pattern Recognition 24*, pp. 283-288,1991.

[Voor86] Voorhees, E. M. "Implementing agglomerative hierarchical clustering

algorithms for use in document retrieval". *Information Processing and Management*, 22:465-476, 1986.

[Voor93] Voorhees, E. M. "Using WordNet to disambiguate word sense for text retrieval". In *Proceeding of ACM SIGIR Conference* 16, pp. 171-180, 1993.

[Voss98] Vossen, P. "The Restructured Core WordNets in EuroWordNet: Subset 1". In *Deliverable D014, D015, WPS, WP4, EuroWordNet LE2-4003*, 1998.

[Wall99] 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.

[Wayn00] Wayne, C. L. "Multilingual Topic Detection, and Tracking: Successful Research Enabled by Corpora and Evaluation". *Language Resources and Evaluation Conference LREC*, pp. 1487-1494, 2000.

[Will88] Willet, P. "Recent trends in hierarchical document clustering: a critical review". *Information Processing & Management* 24(5):577-597, 1988.

[Yamr00] Yamron, J. "Dragon´s Tracking and Detection Systems for TDT2000 Evaluation". *Proceedings of Topic Detection and Tracking Workshop*, pp. 75-80, 2000.

[Yang97] Yang, Y. "An evaluation of statistical approaches to text categorization", *Technical report, Carnegie Mellon University*, 1997.

[Yang98] 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.

[Yang99] 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.

[Yang00] 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.