Generación Automática de Resúmenes de Múltiples Documentos

por

Esaú Villatoro Tello
Lic., BUAP

Tesis sometida como requisito parcial para obtener el grado de

MAESTRO EN CIENCIAS EN LA ESPECIALIDAD DE CIENCIAS COMPUTACIONALES

en el

Instituto Nacional de Astrofísica, Óptica y Electrónica
Febrero 2007
Tonantzintla, Puebla

Supervisada por:

Dr. Luis Villaseñor Pineda
Investigador Titular del INAOE

Dr. Manuel Montes y Gómez
Investigador Titular del INAOE

Resumen

En la era actual en la que vivimos, donde la información en forma textual disponible en medios electrónicos crece de manera exponencial y donde el tiempo es un recurso crítico, se ha vuelto virtualmente imposible para cualquier persona, el navegar y leer toda esta información disponible. Es por esta razón, que surge la importancia de desarrollar métodos que permitan a los usuarios buscar y localizar de una manera rápida, información contenida dentro de grandes colecciones de documentos. La generación automática de resúmenes de múltiples documentos, cumple con estos objetivos al proporcionar a los usuarios un método que permite observar la información importante y/o porciones de información relevante, contenida dentro de una colección de documentos. Actualmente los sistemas de generación de resúmenes de múltiples documentos se encuentran muy poco desarrollados, sin embargo, a la tarea de generar el resumen de un documento se le ha puesto gran interés en los últimos años.

La tarea de generar resúmenes de múltiples documentos se diferencia de la tarea de generar el resumen de un documento en: los niveles de compresión que deben ser manejados, la aparición de información redundante y la forma de seleccionar las porciones de información relevantes, juega un papel crítico al momento de crear un resumen de calidad. Si se desea que el sistema de generación de resúmenes sea útil en diferentes dominios temáticos e incluso diferentes idiomas, es necesario contar con técnicas que no hagan uso de costosos recursos lingüísticos.

La arquitectura que se propone para solucionar el problema de generar el resumen de múltiples documentos, se compone de dos grandes módulos. El primero, basado en técnicas de aprendizaje automático, que tiene por objetivo hacer la adecuada selección de la información relevante. La característica principal de este módulo es el uso de secuencias de palabras para representar las oraciones de los documentos. El segundo

módulo, se compone de un algoritmo de agrupamiento, el cual tiene como objetivo principal organizar la información por sub-temas, eliminar redundancias y controlar los niveles de compresión.

El propósito de este enfoque es eliminar los problemas de portabilidad que actualmente presentan este tipo de sistemas. Finalmente, para mostrar lo útil de la arquitectura propuesta, se compara el desempeño de ésta contra el obtenido por otros dos sistemas. Los resultados de estas evaluaciones demuestran que la propuesta es útil en la creación de resúmenes muy similares en contenido a los creados por humanos.

Abstract

In this era, where electronic text information is exponentially growing and where time is a critical resource, it has become virtually impossible for any user to browse or read large numbers of individual documents. It is therefore important to explore methods of allowing users to locate and browse information quickly within collections of documents. Automatic text summarization of multiple documents fulfills such information seeking goals by providing a method for the user to quickly view highlights and/or relevant portions of document collections. Now days, there has been little work with multi-document summarization, although single document summarization has been subject of focus in the last few years.

Multi-document summarization differs from single document summarization in that the issues of compression levels, management of redundant information and the method used for the sentence selection are critical in the formation of useful summaries. If multi-document summarization needs to be useful across subject areas and languages, it must be relatively independent of natural language understanding (i.e., scarce use of linguistic resources).

The proposed approach to solve the task of multi-document summarization contains two main modules. The first one, a module based on machine learning techniques has as a main goal to identify an extract relevant sentences. The main characteristic of the proposed classifier is that uses word sequences as features to represent sentences. The second module consists of a clustering process, the main goal of this is to organize the information extracted by the classifier and find the main sub-themes contained in the collection, this module also deals with the problem of redundant information and the compression levels.

The main goal of the proposed approach is to reduce the portability problems of current multi-document summarization systems. Finally, in order to show the usefulness of the proposed scheme, a comparison between our proposal and two other

systems was made. The evaluations showed that the proposal is useful for the creation of multi-document summaries of high quality and allows the creation of summaries that are very similar to those created by humans.

Agradecimientos

Se agradece al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo otorgado a través de la beca para estudios de maestría no. 189943.

En general, se agradece al Instituto Nacional de Astrofísica Óptica y Electrónica (INAOE) por las facilidades prestadas tanto en aspectos de investigación como administrativos, en especial a la coordinación de Ciencias Computacionales, cuyos investigadores siempre mostraron su apoyo.

Finalmente, un agradecimiento muy especial a todos los integrantes del Laboratorio de Tecnologías del Lenguaje del INAOE, principalmente a las personas que dirigieron esta tesis, Luis Villaseñor Pineda y Manuel Montes y Gómez, quienes con su conocimiento, experiencia y buen humor lograron llevar a buen término el trabajo.

Dedicatoria

Para mis padres y hermanos.
Por su apoyo y comprensión.

Para Daymirey, una vez más.
Más que nunca.

Índice General

Lista de Figuras

Lista de Tablas

Capítulo 1
Introducción

Actualmente, el crecimiento exponencial que ha tenido la información textual disponible dentro de la Internet, en bases de datos ó bibliotecas digitales, ha provocado que se incremente la importancia de mejorar el desempeño de mecanismos que buscan y presentan esta información a los usuarios. De manera convencional, los sistemas de recuperación de información (IR) buscan y ordenan los documentos en base a la relevancia que tienen con respecto a la petición de un usuario [12, 52]. Recientemente, sistemas que generan resúmenes de un solo documento, entregan un *resumen* genérico construido automáticamente como resultado de una petición [7, 15, 29, 42, 53]. Este resumen, generalmente denominado "resumen indicativo", contiene la información mínima necesaria para que un usuario pueda tomar la decisión de leer o no el documento de donde dicho resumen fue obtenido. De manera "ideal", un resumen debería de contener explícitamente la información que el usuario está buscando. Sin embargo, actualmente los grandes sistemas de generación de resúmenes no han alcanzado este nivel.

Considere la situación en la que un usuario se encuentra realizando una búsqueda en Internet , por ejemplo una noticia, el sistema de recuperación de información encontrará cientos de documentos altamente relacionados. Seguramente muchos de estos documentos coincidirán en mucha de la información y al mismo tiempo diferirán en ciertas partes. Los resúmenes individuales de estos documentos ayudarían al usuario a discernir entre los documentos que le interesan y los que no, pero seguramente estos resúmenes serán muy parecidos entre si, a menos que el sistema de generación de resúmenes, tome en cuenta la información contenida en los resúmenes que ya se han ido generando. En el estilo de vida de la sociedad actual, el tiempo es un recurso muy importante, los sistemas de generación de resúmenes de múltiples documentos –capaces de resumir ya sea colecciones completas de documentos, o do-

cumentos simples– son esenciales para este tipo de situaciones. El trabajo realizado en esta tesis se enfoca principalmente en la creación de estos resúmenes de múltiples documentos, para así ayudar a resolver las necesidades de información de los usuarios. Idealmente, los resúmenes de múltiples documentos, deberían contener toda la información relevante compartida entre los documentos de la colección (una sola vez), y además toda la información única y relevante al tópico de la colección que se encuentre presente en los documentos de manera individual.

La generación de resúmenes basada en extractos (*extractive summarization*), i.e., construir resúmenes reutilizando porciones del documento(s) original (palabras, oraciones, párrafos, etc.), es una de las técnicas que hasta nuestros días sigue siendo ampliamente utilizada por muchos grupos de investigación. En un principio, los sistemas de generación de resúmenes de un solo documento se enfocaban en el cálculo de atributos relativamente sencillos (atributos superficiales); por ejemplo, la posición de las oraciones dentro del texto [9, 18], frecuencia de palabras [34], uso palabras clave (e.g., "es importante notar", "en conclusión", etc.) [18].

Una vez que se han calculado estos atributos, se realiza una suma normalizada de estos con el fin de asignar un valor numérico (peso) a cada oración. Finalmente las oraciones que obtengan un valor más alto, son ordenadas y entregadas al usuario como resumen final.

Técnicas más recientes, utilizan conjuntos de atributos más sofisticados (e.g., similaridad y relación entre oraciones y/o palabras, análisis sintácticos, análisis semánticos, etc.), para tomar la decisión de que oraciones *extraer*. Una de las desventajas más claras al hacer uso de este tipo de técnicas, es que el sistema de generación de resúmenes se vuelve altamente dependiente del dominio temático y también del idioma de los documentos.

Una de las características importantes de estas técnicas recientes, es la incorporación de métodos de *aprendizaje automático*, para hacer la identificación de atributos importantes. Uno de los pioneros en mostrar las ventajas del aprendizaje automático fue Kupiec [29], quien utiliza un clasificador Bayesiano para generar resúmenes de un documento.

Sin embargo, la tarea de generar resúmenes de múltiples documentos involucra nuevos problemas. En primer lugar, el sistema debe ser capaz de identificar la información importante dentro de cada documento, evitar redundancias y además debe de ser capaz de entregar la información final en forma coherente. Uno de los primeros

trabajos realizados para la resolución de esta tarea, se basaba en el uso de un módulo de *extracción de información*(IE). Es decir, el sistema extrae la información requerida por las plantillas para cada uno de los documentos de la colección, posteriormente se hace una comparación entre ellas, y de esta forma se logra la identificación de similitudes y diferencias entre los múltiples documentos, finalmente la información similar junto con la diferente se da al usuario como el resumen de los múltiples documentos [40].

Trabajos más recientes siguen utilizando técnicas de IE en combinación con un módulo especial para la identificación de diferencias entre documentos, el cuál hace uso de reglas del discurso, además de un módulo encargado de comprimir y/o fusionar oraciones, con el objetivo de producir resúmenes más coherentes [46]. El problema con este tipo de sistemas es, que debido al uso de técnicas de extracción de información, sólo buscan información predeterminada dentro de los documentos, volviéndose sistemas dependientes del dominio; y debido al uso de sofisticados recursos lingüísticoos, se vuelven dependientes del lenguaje.

El propósito en ambos casos es la creación de sistemas con una mayor portabilidad, no sólo entre distintos dominios temáticos, sino también entre idiomas, además de alcanzar una mayor eficiencia al no utilizar un costoso análisis lingüístico. De lo anterior, las preguntas generales y que buscamos responder a lo largo del documento de tesis son: *¿cómo construir un sistema de generación de resúmenes de múltiples documentos que haga uso de las ventajas que aportan los métodos de aprendizaje automático, sin la necesidad de utilizar costosos recursos lingüísticos?* y *¿qué tan eficaz puede ser un sistema con estas características?*.

A continuación, se introduce de manera formal la descripción del problema, además de continuar exponiendo la motivación del trabajo. Posteriormente , en la sección 1.3 se presenta la organización de la tesis.

1.1. Descripción del Problema

Un resumen puede ser definido como un texto que es producido a partir de uno o más documentos. Las características principales de este texto son: contiene sólo la información importante del documento(s) original y son generalmente textos cortos.

El principal objetivo de un sistema generador de resúmenes es, presentar a un usuario las ideas principales de uno o varios documentos, en un documento pequeño.

Si todas las oraciones dentro de un documento tuvieran la misma importancia, la tarea de generar un resumen no sería muy efectiva, pues cualquier reducción en tamaño del documento significaría la pérdida de información importante. Afortunadamente, la información relevante de un documento tiende a aparecer sólo en determinadas secciones, de esta forma un algoritmo adecuado será capaz de diferenciar entre oraciones que contengan más o menos información relevante. El reto de un sistema generador de resúmenes es entonces identificar oraciones relevantes.

Los resúmenes pueden ser caracterizados por su contenido en resúmenes indicativos y resúmenes informativos [25], donde los primeros sólo aportan una idea general sobre el contenido real del documento origen, y los segundos son los que dan al usuario una version corta del contenido real del documento origen. Sin embargo, los resúmenes pueden ser clasificados también por la forma en que son construidos, estas son: (i) *resúmenes basados en extractos* los cuales se crean re-utilizando porciones del documento original (i.e., palabras, oraciones, párrafos, etc.); y (ii)*resumen ó abstracto*, los cuales son creados utilizando herramientas que permiten comprimir y/o fusionar oraciones, con el objetivo de construir resúmenes más coherentes.

La gran mayoría de los sistemas actuales utilizan en un primer paso la identificación y *extracción* de unidades de información importantes dentro de un documento para la creación del resumen final. El paso posterior consiste en crear nuevas oraciones a partir de la información extraída en el primer paso. En este trabajo de tesis nos enfocaremos en el paso de la *extracción* de la información importante para la creación de un resumen basado en extractos[1].

Como se menciona en la pregunta de investigación nos interesa crear un sistema generador de resúmenes que sea portable a diferentes dominios temáticos y además a distintos idiomas. La hipótesis de la cual se parte para la realización de este trabajo y que se pretende probar en el contenido de este documento es, la posibilidad de aplicar técnicas aprendizaje automático dentro de un sistema de generación de resúmenes de múltiples documentos, sin la necesidad de aplicar sofisticados recursos lingüísticos.

Como se mencionó antes, actualmente los sistemas de generación de resúmenes de múltiples documentos son métodos tan especializados que se vuelven sistemas altamente dependientes del dominio y del lenguaje [8, 28, 46]. Por otro lado existen también métodos que hacen uso de atributos superficiales [45], por ejemplo conside-

[1] A partir de este punto el uso de la palabra *"resumen ó resúmenes"* la utilizaremos para referirnos a un *"resumen basado en extractos"*

ran la longitud de las oraciones, su posición dentro del documento, etc., los cuales muestran cómo la tarea de generar resúmenes de múltiples documentos puede ser realizada sin la necesidad de complejos recursos lingüísticos. El problema principal con el trabajo propuesto en [45] es que no se hace ningún tipo de estudio que asegure o demuestre que los atributos que se están utilizando, son los mejores para representar y ponderar las oraciones.

A partir de esto surgen varias preguntas, ¿tiene algún sentido utilizar técnicas de aprendizaje automático?, ¿es posible identificar información discriminante (atributos) que ayuden a resolver el problema de clasificación de oraciones relevantes?, ¿cuáles son los mejores atributos?, Una vez que las oraciones importantes de cada documento han sido extraídas ¿cómo mezclar toda esta información? ,¿es posible identificar información característica que ayude a la eliminación de redundancias entre oraciones sin la ayuda de complejos recursos lingüísticos?, ¿qué método nos ayudaría a identificar la información común y la información única?, ¿un sistema con estas características será realmente un sistema portable?, ¿cuál será la calidad de los resúmenes creados?.

1.2. Objetivos

Objetivo general:

A partir de estas preguntas surge el objetivo principal de este trabajo de tesis:

- Proponer un método para generar resúmenes de múltiples documentos, que aproveche las ventajas de utilizar técnicas de aprendizaje automático en combinación con métodos que no hagan uso de grandes y/o complejos recursos lingüísticos para la selección de oraciones relevantes.

Objetivos Particulares:

- Identificar información discriminante (atributos) para la resolución del problema de clasificación de oraciones relevantes.
- Identificar información característica que ayude a la eliminación de redundancias entre oraciones.

En el resto del contenido de este documento, se tratará de dar respuesta a todas las preguntas planteadas hasta este momento. Además de estas, se tratará de responder

otra pregunta: *¿cuál es la calidad de los resúmenes generados con esta propuesta?*, i.e., *¿cuáles son los alcances y las limitaciones de este trabajo?*.

1.3. Organización de la Tesis

El documento está organizado de la siguiente forma: en el siguiente capítulo se presentan conceptos básicos para entender el contenido de la tesis, los cuales incluyen principalmente conocimientos de aprendizaje automático relacionados a nuestro problema de investigación y la descripción del algoritmo de aprendizaje utilizado en este trabajo. Además de esto, se incluye una sección donde se describe brevemente la tarea de clasificación automática de textos, en la que se abordan conceptos clave que sirvieron a la arquitectura propuesta para la identificación de las oraciones relevantes dentro de un documento. Por último, se expone la tarea de agrupamiento de textos, tarea que forma parte fundamental de la arquitectura propuesta en este trabajo.

En el capítulo 3 se presenta una revisión del trabajo más relevante y reciente en el área de generación de resúmenes, tanto de un documento como de múltiples documentos. Dentro de este capítulo se describen las arquitecturas tradicionalmente utilizadas por diferentes grupos de investigación. Entre los objetivos de esta revisión esta el identificar los aspectos positivos de diseño, así como los negativos con el propósito de evitar tales obstáculos.

Los capítulos 4 y 5 son la parte más importante de este trabajo de tesis, debido a que en éstos se resume nuestra contribución al conocimiento. En el primero se exhibe tanto la idea como la arquitectura propuestas para resolver el problema de generar el resumen de un documento. La principal aportación de este capítulo se resume a la propuesta de una nueva forma de representación de las oraciones y además se discuten las ventajas y las desventajas de la misma. En el capítulo 5 se describe la idea y la arquitectura propuesta para resolver el problema de la generación de resúmenes de múltiples documentos. Entre las principales aportaciones expuestas en este capítulo se encuentran la metodología seguida para la identificación tanto de la información común como la información única dentro de la colección de documentos.

Finalmente, en el capítulo 6 resumimos las principales aportaciones de la investigación así como las conclusiones, y discutimos posibles direcciones para trabajo futuro.

Capítulo 2
Conceptos Básicos

El objetivo de este capítulo es describir e introducir al lector de manera rápida a la teoría que fundamenta el trabajo realizado en esta tesis. Se presentan las definiciones formales utilizadas a lo largo del documento. En primer lugar se pretende familiarizar al lector con la tarea de aprendizaje automático (sección 2.1) y el algoritmo de aprendizaje utilizado en la tesis (sección 2.1.3). Posteriormente la sección 2.2 describe la tarea de clasificación automática de textos, uno de los puntos clave de la idea que fundamentan el trabajo de tesis y que es expuesto en el capítulo 4. Finalmente se introduce al lector al área de agrupamiento de textos, segundo punto clave dentro de este trabajo de tesis, que es expuesto en el capítulo 5.

2.1. Aprendizaje Automático

El aprendizaje automático es la disciplina que estudia cómo construir sistemas computacionales que mejoren automáticamente mediante la experiencia. En otras palabras, se dice que un programa "aprendió" a desarrollar la tarea T, si después de proporcionarle la experiencia E, el sistema es capaz de desempeñarse razonablemente bien cuando nuevas situaciones de la tarea se presenten. Aquí, E es generalmente un conjunto de ejemplos de T, y el desempeño es medido usando una métrica de calidad P. Por lo tanto, un problema de aprendizaje bien definido requiere que T, E y P estén bien especificados [41].

A pesar de que se desconoce cómo lograr que las computadoras aprendan tan bien como las personas, algunos algoritmos propuestos dentro del área de aprendizaje automático han resultado ser bastante efectivos en varias tareas de aprendizaje. La clasificación automática es un claro ejemplo de este tipo de tareas. Debido a la im-

portancia que tiene el concepto de clasificación automática para este trabajo de tesis, en las siguientes secciones se expone la definición formal de éste, así como la forma habitual de medir su desempeño.

2.1.1. Clasificación

La *clasificación* puede ser formalizada como la tarea de aproximar una *función objetivo* desconocida $\Phi : I \times C \rightarrow \{T, F\}$ (que describe cómo las instancias del problema deben ser clasificadas de acuerdo a un experto en el dominio) por medio de una función $\Theta : I \times C \rightarrow \{T, F\}$ llamada el *clasificador*, donde $C = \{c_1, \ldots, c_{|c|}\}$ es un conjunto de categorías predefinido, e I es un conjunto de instancias del problema. Comúnmente cada instancia $i_j \in I$ es representada como una lista $A = \langle a_1, a_2, \ldots, a_{|A|} \rangle$ de valores característicos, conocidos como *atributos*, i.e. $i_j = \langle a_{1j}, a_{2j}, \ldots, a_{|A|j} \rangle$. Si $\Phi : i_j \times c_i \rightarrow T$, entonces i_j es llamado un *ejemplo positivo* de c_i, por otro lado si $\Phi : i_j \times c_i \rightarrow F$ éste es llamado un *ejemplo negativo* de c_i.

Para generar automáticamente el clasificador de c_i es necesario un proceso inductivo, llamado el *aprendiz*, el cual por observar los atributos de un conjunto de instancias pre-clasificadas bajo c_i o $\bar{c_i}$, adquiere los atributos que una instancia no vista debe tener para pertenecer a esa categoría. Por tal motivo, en la construcción del clasificador se requiere de la disponibilidad inicial de una colección Ω de ejemplos tales que el valor de $\Phi(i_j, c_i)$ es conocido para cada $\langle i_j, c_i \rangle \in \Omega \times C$. A la colección usualmente se le llama *conjunto de entrenamiento* (Tr). En resumen, al proceso anterior se le identifica como *aprendizaje supervisado* debido a la dependencia en Tr.

2.1.2. Evaluación de la Clasificación

En la investigación realizada, la efectividad es considerada usualmente como uno de los criterios de evaluación gracias a su confiabilidad para comparar diferentes metodologías. La forma usual de medir la efectividad de un clasificador es por medio de su exactitud (α), i.e. el porcentaje de decisiones correctas. Un cálculo para la exactitud sobre la clase c_i puede darse en términos de su tabla de contingencia de la siguiente manera:

$$\alpha_i = \frac{TP_i + TN_i}{TP_i + FP_i + TN_i + FN_i} \tag{2.1.1}$$

donde FP_i *(falsos positivos)* es el número de instancias de prueba incorrectamente clasificadas bajo c_i, TN_i *(verdaderos negativos)* es el número de instancias de prueba correctamente clasificadas bajo $\bar{c_i}$, TP_i *(verdaderos positivos)* y FN_i *(falsos negativos)* son definidos consecuentemente (Ver tabla 2.1).

Tabla 2.1: Tabla de contingencia para la clase c_i

		Juicio del Experto	
Categoría C_i		SI	NO
Juicio del	SI	TP_i	FP_i
Clasificador	NO	FN_i	TN_i

Durante la experimentación es usual dividir Ω en los tres conjuntos disjuntos Tr *(el conjunto de entrenamiento)*, Va *(el conjunto de validación)* y Te *(el conjunto de prueba)*. El conjunto de entrenamiento es el conjunto de instancias observadas cuando el aprendiz construye el clasificador. El conjunto de validación es el conjunto de instancias utilizadas para optimizar parámetros en los clasificadores, o para seleccionar uno en particular. Entonces, el conjunto de prueba es el conjunto de instancias sobre las cuales se evalúa finalmente la efectividad del clasificador. Sin embargo, la partición anterior no es adecuada cuando la cantidad de datos en Ω es limitada; en tal caso, el camino estándar para calcular la exactitud de una técnica de aprendizaje es usar una *validación cruzada con 10 pliegues* (*10FCV*, por su siglas en inglés) [54]. Esta técnica consiste en dividir aleatoriamente Ω en diez partes, conservando en cada partición la proporción original de las clases, posteriormente cada parte es mantenida una vez y el esquema de aprendizaje entrena sobre las nueve partes restantes, entonces la exactitud es calculada sobre la parte conservada fuera del proceso de entrenamiento. Así, el proceso es ejecutado un total de diez veces sobre diferentes conjuntos de entrenamiento. Finalmente, los diez estimados de exactitud son promediados para producir una completa estimación de la misma.

2.1.3. Algoritmos de Aprendizaje

Diferentes tipos de aprendices, i.e. el proceso inductivo necesario para generar automáticamente el clasificador, han sido utilizados en la literatura de aprendizaje supervisado. Dado que el objetivo de la investigación no involucra hacer un estudio riguroso sobre cual podría ser el mejor clasificador o aprendiz para la tarea de gene-

ración automática de resúmenes, sólo nos enfocamos a uno de estos (Naïve Bayes), un clasificador que aunque sencillo ha demostrado ser competente con los esquemas más complejos dentro de tareas que involucra trabajar con valores que caracterizan texto.

Naïve Bayes

El clasificador Naïve Bayes (NB) se considera como parte de los clasificadores probabilísticos, los cuales se basan en la suposición que las cantidades de interés se rigen por distribuciones de probabilidad, y que la decisión óptima puede tomarse por medio de razonar acerca de esas probabilidades junto con los datos observados [41]. Este algoritmo ha sido ampliamente usado dentro de la tarea de generación automática de resúmenes [29]. Para nuestro trabajo empleamos el Naïve Bayes tradicional, el cual se describe a continuación.

En este esquema el clasificador es construido usando Tr para estimar la probabilidad de cada clase. Entonces, cuando una nueva instancia i_j es presentada, el clasificador le asigna la categoría $c \in C$ más probable por aplicar la regla:

$$c = \arg \max_{c_i \in C} P(c_i|i_j) \tag{2.1.2}$$

utilizando el teorema de Bayes para estimar la probabilidad tenemos:

$$c = \arg \max_{c_i \in C} \frac{P(i_j|c_i)P(c_i)}{P(i_j)} \tag{2.1.3}$$

dado que el denominador en la ecuación anterior no difiere entre categorías puede omitirse quedando de la siguiente forma:

$$c = \arg \max_{c_i \in C} P(i_j|c_i)P(c_i) \tag{2.1.4}$$

tomando en cuenta que el esquema es llamado "naïve" debido al supuesto de independencia entre atributos, i.e. se asume que las características son condicionalmente independientes dadas las clases. Esto simplifica los cálculos produciendo:

$$c = \arg \max_{c_i \in C} P(c_i) \prod_{k=1}^{n} P(a_{kj}|c_i) \tag{2.1.5}$$

donde $P(c_i)$ es la fracción de ejemplos en Tr que pertenecen a la clase c_i, es decir:

$$P(c_i) = \frac{|Tr_{c_i}|}{|Tr|} \tag{2.1.6}$$

y $P(a_{kj}|c_i)$ se calcula de acuerdo a:

$$P(a_{kj}|c_i) = \frac{1 + |Tr_{ki}|}{|A| + \sum_{l=1}^{|A|} |Tr_{li}|} \tag{2.1.7}$$

donde $|Tr_{ki}|$ es el número de elementos dentro del conjunto de entrenamiento que poseen el atributo k y además pertenecen a la clase c_i, $|A|$ es el número total de atributos, de esta forma $|Tr_{li}|$ es el número de ejemplos de entrenamiento que poseen el atributo l y pertenecen a la clase c_i.

En resumen, la tarea de aprendizaje en el clasificador Naïve Bayes consiste en construir una hipótesis por medio de estimar las diferentes probabilidades $P(c_i)$ y $P(a_{kj}|c_i)$ en términos de sus frecuencias sobre Tr.

2.2. Clasificación Automática de Textos

La *clasificación de textos* (también conocida como categorización de textos o ubicación de temas), es la tarea de clasificar automáticamente un conjunto de documentos en categorías (o temas) dentro de un conjunto predefinido [48]. Actualmente la exactitud de muchos de los sistemas de clasificación de textos compite con la de profesionales humanos especializados. Tal avance se debe principalmente a la combinación de tecnologías como son la recuperación de información y el aprendizaje automático, lo cual, desde principios de los 90's ha ganado popularidad y eventualmente se ha convertido en el enfoque dominante para construir sistemas de clasificación de textos. La idea básica de este enfoque es, que un proceso inductivo automáticamente construya un clasificador al observar las características de un conjunto de documentos previamente clasificados (Figura 2.1). De tal forma que el problema de clasificación de textos se convierte en una actividad de aprendizaje supervisado como se describe en la sección 2.1.

Sin embargo, un inconveniente inicial es que los textos no pueden ser directamente interpretados en primer lugar por el aprendiz, y posteriormente por el clasificador una vez que éste ha sido construido. Por lo tanto, antes de aplicar el método inductivo

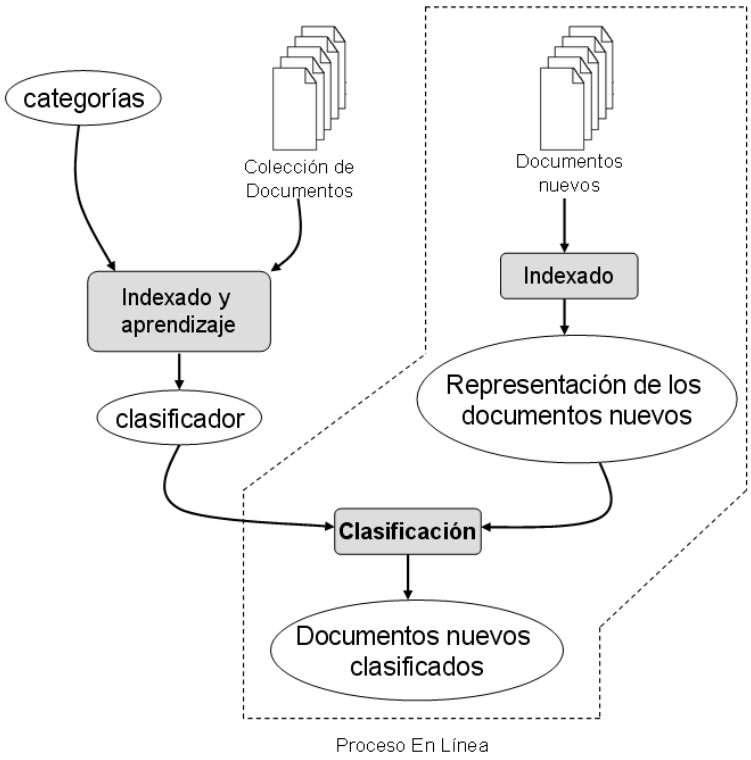

Figura 2.1: Visión general de los procesos en la clasificación de textos.

un proceso conocido como indexado se aplica para representar los textos, el cual necesita ser aplicado uniformemente al conjunto de entrenamiento, así como a los nuevos documentos a ser catalogados.

En la evaluación de efectividad para la clasificación de textos, las medidas generalmente usadas son la precisión y recuerdo [30], que miden lo correcto y completo del método, respectivamente. Además de estas dos se tiene la medida F, que es una combinación lineal de la precisión y del recuerdo.

2.2.1. Forma de Representación de los Documentos

El *indexado* denota la actividad de hacer el mapeo de un documento d_j en una forma compacta de su contenido[1]. La representación más comúnmente utilizada para

[1]A pesar de que este tipo de indexado difiere del utilizado en recuperación de información, en la clasificación de textos se decidió utilizar el mismo nombre para referirse a la representación de los textos, esto por que hay una similitud en los procesos realizados.

representar cada documento es un vector con términos ponderados como entradas, concepto tomado del modelo de espacio vectorial usado en recuperación de información [4]. Es decir, un texto d_j es representado como el vector $\overrightarrow{d}_j = \langle w_{ij}, \ldots, w_{|\tau|j} \rangle$, donde τ es el *diccionario*, i.e., el conjunto de términos que ocurren al menos una vez en algún documento de Tr, mientras que w_{kj} representa la importancia del término t_k dentro del contenido del documento d_j. En ocasiones τ es el resultado de filtrar las palabras del vocabulario con respecto a una lista de *palabras vacías*[2] y al uso de un *lematizador*[3]. Una vez que hemos hecho los filtrados necesarios, el diccionario τ puede definirse de acuerdo a diferentes criterios:

- **Bolsa de Palabras (BOW)[4]:** Dentro del área de TC es la forma tradicionalmente utilizada para representar los documentos [48]. Este método de representación utiliza a las palabras simples como los elementos del vector de términos.

- **n-Gramas:** Este tipo de representación, utilizado también dentro del área de TC [10, 19, 14] ha demostrado ser una buena forma de representación, pues compite con la representación por medio de BOW. Este método de representación utiliza n palabras consecutivas como los elementos del vector de términos.

- **Secuencias Frecuentes Maximales (SFM) [1, 16, 23]:** Asumamos un conjunto de textos D (siendo D un documento entero o incluso una sola oración) y cada texto consiste de una secuencia de palabras. De aquí tenemos las siguientes definiciones:

  1. Una secuencia $p = a_1 \ldots a_k$ es una *subsecuencia* de una secuencia q si todos los elementos $a_i, 1 \leq i \leq k$, ocurren en q y además ocurren en el mismo orden que en p. Si una secuencia p es una subsecuencia de una secuencia q, entonces decimos que p ocurre en q.

  2. Una secuencia p es *frecuente* en D si p es una subsecuencia de al menos σ textos de D, donde σ es un umbral de frecuencia dado.

[2]Son palabras frecuentes que no transmiten información, e.g. artículos y preposiciones.

[3]El objetivo es obtener las raíces morfológicas de las palabras tras aplicar un algoritmo de eliminación de afijos de tal modo que aparezca sólo la raíz léxica de las palabras, e.g. el algoritmo de Porter (una versión para el idioma español de este algoritmo se encuentra en http://snowball.tartarus.org/) que elimina sufijos de términos en inglés.

[4]Siglas en inglés para el término Bag Of Words.

3. Una secuencia p es una *secuencia frecuente maximal* en D si no existe ninguna otra secuencia p' en D tal que p sea una subsecuencia de p' y p' sea frecuente en D.

Note que el valor de n, para el caso de los n-gramas y el de k, para el caso de las SFM, ambos pueden ser igual a 1, lo cual nos lleva a la representación inicial, i.e. la bolsa de palabras o palabras simples.

Con respecto al peso w_{kj}, se tienen diferentes formas de calcularlo, entre las más usadas en la comunidad científica se tienen el ponderado booleano y ponderado por frecuencia de término. Una breve descripción es dada a continuación:

- *Ponderado Booleano:* Consiste en asignar el peso de 1 si la palabra ocurre en el documento y 0 en otro caso.

$$W_{kj} = \begin{cases} 1, & \text{si } t_k \in d_j \\ 0, & \text{en otro caso} \end{cases} \tag{2.2.1}$$

- *Ponderado por frecuencia de termino:* En este caso el valor asignado es el número de veces que el término t_k ocurre en el documento d_j.

$$W_{kj} = f_{kj} \tag{2.2.2}$$

- *Ponderado por frecuencia relativa:* Este tipo de ponderado es una variación del tipo anterior y se calcula de la siguiente forma:

$$W_{kj} = TF(t_k) \times IDF(t_k) \tag{2.2.3}$$

  donde $TF(t_k) = f_{kj}$, es decir, la frecuencia del termino t_k en el documento d_j. IDF es conocido como la "frecuencia inversa" del termino t_k dentro del documento d_j. El valor de IDF es una manera de medir la "rareza" del termino t_k. Para calcular el valor de IDF se utiliza la siguiente formula:

$$IDF(t_k) = \log \frac{|d_j|}{\#oraciones\ en\ d_j\ que\ contienen\ t_k} \tag{2.2.4}$$

Este tipo de técnicas trae un costo agregado, que es el producir un espacio de términos (atributos) τ de alta dimensionalidad (i.e.,$|\tau| \to \infty$). A este problema se le conoce

comúnmente como un problema de alta dimensionalidad de los datos, el cual puede ocasionar problemas de *sobre-ajuste* en el proceso de aprendizaje, i.e. ocurre el fenómeno por medio del cuál un clasificador se adapta a las características contingentes de Tr, en lugar de únicamente a las características constitutivas de las categorías, provocando problemas de efectividad debido a que el clasificador tiende a comportarse mejor sobre los datos con los que ha sido entrenado y sin conservar la tendencia en aquellos no vistos.

Además de esto, la alta dimensionalidad también se refleja en la eficiencia, haciendo el problema menos tratable para el método de aprendizaje. Para evitar el problema un subconjunto de τ es a menudo seleccionado (i.e. una *selección de características*), este proceso se conoce como *reducción de dimensionalidad*. Su efecto es producir un vector de características que cumple con $|\tau'| \ll |\tau|$; el conjunto τ' es llamado el *conjunto de términos reducido*

Dentro del área de aprendizaje automático, lidiar con este tipo de problemas es algo común. Existen diferentes algoritmos diseñados para ayudar a reducir la dimensionalidad de los datos, uno de estos es conocido como Ganancia de Información, el cual por medio de una función de calidad asigna una calificación a cada uno de los términos de τ, quedándose al final con los de mayor calificación.

Ganancia de Información

El algoritmo de Ganancia de Información, utiliza la entropía como una medida de impureza de los ejemplos de entrenamiento dentro de una colección. A partir de aquí es posible definir una medida de efectividad para cada atributo al momento de clasificar un elemento de los ejemplos de entrenamiento. Esta medida es llamada *ganancia de información*, y es definida simplemente como la reducción esperada de la entropía causada por el particionamiento de los ejemplos de entrenamiento de acuerdo a un atributo [41]. De manera más precisa, la ganancia de información, $Ganancia(Tr, A)$ de un atributo A relativo a una colección de ejemplos Tr, se define como:

$$Ganancia(Tr, A) \equiv Entropia(Tr) - \sum_{v \in Valores(A)} \frac{|Tr_v|}{|Tr|} Entropia(Tr_v) \qquad (2.2.5)$$

donde $Valores(A)$ es el conjunto de todos los posibles valores para el atributo A, y Tr_v es el subconjunto de Tr para el cuál el atributo A tiene el valor v (i.e. $Tr_v =$

$\{tr \in Tr | A(tr) = v\}$). Para hacer el cálculo de la entropía utilizamos la siguiente formula:

$$Entropia(Tr) = \sum_{i=1}^{|C|} -P(c_i) \log_2 P(c_i) \tag{2.2.6}$$

Nótese que el primer término de la ecuación (2.2.5) se refiere a la entropía de la colección original Tr, y que el segundo termino es el valor esperado de la entropía después de que Tr es particionado usando el atributo A. La entropía esperada que describe este segundo término es simplemente la suma de las entropías de cada subconjunto Tr_v, ponderado por la fracción de ejemplos $\frac{|Tr_v|}{|Tr|}$ que pertenecen a Tr_v. Así entonces, $Ganancia(Tr, A)$ es la reducción esperada de la entropía causada por el valor conocido de un atributo A.

2.2.2. Evaluación de la Clasificación de Textos

La *precisión* (π), *recuerdo* (ρ) y medida F (F_β) son nociones clásicas de recuperación de información adaptadas a la clasificación de textos. Donde π es la probabilidad de que si un documento aleatorio d_x es clasificado bajo c_i, esta decisión sea correcta. Mientras que ρ es la probabilidad de que si un documento aleatorio d_x debe ser clasificado bajo c_i, esta decisión sea tomada. Estas probabilidades pueden ser estimadas como se muestra en la tabla 2.2, donde el micro-promedio es para dar a las categorías una importancia proporcional al número de ejemplos positivos que le corresponden, mientras que en el macro-promedio todas las categorías importan lo mismo (las ecuaciones están en términos de la tabla de contingencia para la categoría c_i, ver tabla 2.1).

Tabla 2.2: Medidas de evaluación para un sistema de clasificación de textos

Medida	Micro-Promedio	Macro-Promedio
Precisión(π)	$\pi = \frac{\sum_{i=1}^{\|C\|} TP_i}{\sum_{i=1}^{\|C\|} TP_i + FP_i}$	$\pi = \frac{\sum_{i=1}^{\|C\|} \frac{TP_i}{TP_i + FP_i}}{\|C\|}$
Recuerdo (ρ)	$\rho = \frac{\sum_{i=1}^{\|C\|} TP_i}{\sum_{i=1}^{\|C\|} TP_i + FN_i}$	$\rho = \frac{\sum_{i=1}^{\|C\|} \frac{TP_i}{TP_i + FN_i}}{\|C\|}$

Además se tiene una combinación lineal de π y ρ calculada por la función F_β, ésta

es:

$$F_\beta = \frac{(\beta^2 + 1) \times \pi \times \rho}{\beta^2 \times \pi + \rho} \tag{2.2.7}$$

aquí β puede ser vista como el grado de importancia atribuido a π y ρ. Si $\beta = 0$ entonces F_β coincide con π, mientras que si $\beta = +\infty$ entonces F_β coincide con ρ. Usualmente, un valor de $\beta = 1$ es usado, el cual atribuye igual importancia a π y ρ.

2.3. Agrupamiento

El análisis y organización de información es la base de muchas aplicaciones computacionales, ya sea en la fase de diseño o como parte del proceso en línea de algún sistema. Un elemento clave en el proceso del análisis de información es el agrupamiento ó *clustering*. El agrupamiento consiste en la organización de una colección de patrones[5], en grupos basándose en su similaridad.

Intuitivamente, las características de un grupo válido son más similares entre ellas que aquellas que pertenecen a elementos de otros grupos. Un ejemplo de la tarea de agrupamiento se muestra en la figura 2.2. La entrada al algoritmo de agrupamiento se muestra en la figura 2.2(a), mientras que la salida deseada se muestra en la figura 2.2(b). En la figura, los puntos que pertenecen al mismo grupo tienen la misma etiqueta numérica. La gran variedad de técnicas existentes para la representación de los datos junto con las diferentes formas de medir la proximidad (i.e., medidas de similaridad) entre los elementos han permitido la existencia de gran cantidad de métodos de agrupamiento.

Es importante remarcar que el agrupamiento (*clasificación no supervisada*) y el análisis discriminante(*clasificación supervisada*) son tareas muy diferentes. En la clasificación supervisada, se cuenta con una colección de elementos *etiquetados* (preclasificados), aquí el problema consiste en etiquetar un nuevo elemento encontrado aún sin etiquetar. Típicamente, el conjunto de elementos etiquetados (*conjunto de entrenamiento*) es utilizado para aprender las características de las clases o grupos entre los cuales se quiere ubicar a el nuevo elemento. En el caso del agrupamiento como tarea no supervisada, el problema es encontrar y formar grupos significativos a partir de una colección de elementos no etiquetados. En cierta forma, las etiquetas

[5]Estos patrones son usualmente representados como un vector de características ponderadas, o como un punto en un espacio multidimensional

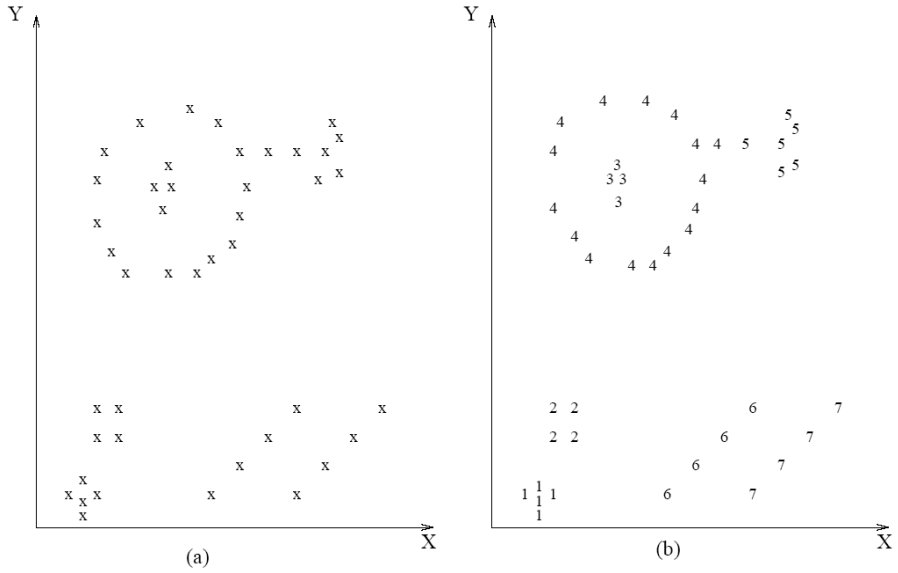

Figura 2.2: Proceso de agrupamiento.

son asociadas con el número de grupos, la característica de estas etiquetas es que son deducidas únicamente a partir de los elementos de entrada [27].

Típicamente, la tarea de agrupamiento involucra los siguientes pasos [26]:

1. Representación de los documentos (extracción y/o selección de atributos): este paso se refiere al proceso de indexado necesario para poder ejecutar el algoritmo de agrupamiento seleccionado. En la sección 2.2.1 se describe con mayor detalle en que consiste el indexado de los documentos.

2. Definición de una medida de proximidad apropiada al conjunto de datos: Muchos de los algoritmos de agrupamiento existentes hacen el cálculo de medidas de proximidad entre los elementos de entrada como paso previo a la etapa de agrupamiento. Las medidas de similaridad utilizadas en este trabajo de tesis son explicadas en mayor detalle en la sección 5.1.2.

3. Aplicar el algoritmo de agrupamiento.

4. Extracción de datos (si es necesario): Consiste en entregar una representación simple y compacta de cada grupo encontrado por el algoritmo de agrupamiento, e.g., el centroide de cada grupo, similaridad entre elementos, etc.

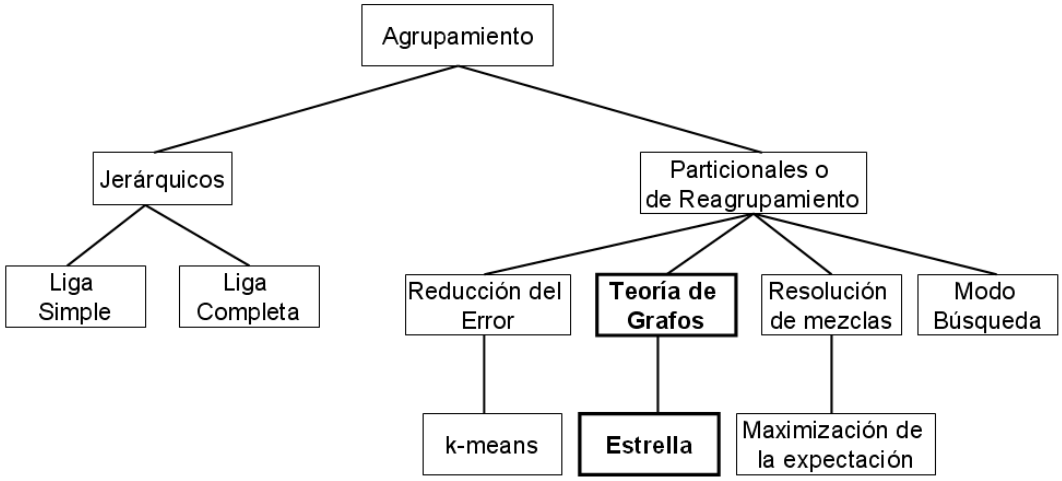

Figura 2.3: Técnicas de Agrupamiento.

5. Establecer la salida del agrupamiento: Mostrar el conjunto de grupos encontrados junto con la evaluación de calidad de los grupos formados.

2.3.1. Técnicas de Agrupamiento

En la figura 2.3 se puede observar una descripción de las diferentes técnicas de agrupamiento. En el nivel más alto se distingue una división entre técnicas jerárquicas y particionales. La principal distinción entre estas dos técnicas es, que los algoritmos jerárquicos producen una serie de particiones anidadas, mientras que los particionales producen solamente una.

Además de la clasificación mostrada en la figura 2.3, las técnicas de agrupamiento se caracterizan también por las siguientes propiedades:

- Algomerativos *vs.* Divisivos. Los algoritmos aglomerativos comienzan con cada elemento en un grupo (grupos simples), y sucesivamente mezclan los grupos hasta que un criterio de paro es alcanzado. Los algoritmos divisivos comienzan con todos los elementos de entrada dentro de un solo grupo, el cual se va dividiendo hasta alcanzar un criterio de paro.

- Estrictos *vs.* Difusos: Los algoritmos estrictos colocan a cada elemento dentro de un grupo y no lo mueven durante el resto del proceso de agrupamiento.

Los algoritmos difusos asignan a cada elemento de la colección un grado de pertenencia a cada uno de los diferentes grupos.

- Determinísticos *vs.* Estocásticos: Este tipo de algoritmos están diseñados para optimizar (minimizar) una función de error. Esta optimización puede ser alcanzada usando técnicas tradicionales o a través de búsquedas aleatorias en el espacio de estados.

- Incrementales *vs.* No-incrementales: Estos algoritmos son útiles cuando el conjunto de elementos de entrada es demasiado grande y además se tienen restricciones en tiempo y memoria.

Para más detalles sobre las diferentes técnicas de agrupamiento refiérase a [27, 26].

2.3.2. Evaluación del Agrupamiento

Todo algoritmo de agrupamiento, al darle una colección de entrada, producirá siempre algún conjunto de grupos sin importar si estos grupos existen o no realmente en la colección de entrada. Si en la colección de entrada existe algún conjunto de grupos, entonces algún algoritmo generará una "mejor" salida que otros. De esta forma, la valoración de la salida de un proceso de agrupamiento se divide entonces en varias facetas. La primera de estas debería ser un análisis sobre los datos de entrada con el objetivo de determinar si es conveniente aplicar un algoritmo de agrupamiento —una colección de entrada que no contiene grupos no debería ser procesada por ningún algoritmo de agrupamiento–, a este estudio se le denomina análisis de la *tendencia de grupos*, pero debido al costo que implica este análisis no es comúnmente realizado.

El análisis de la *validez de los grupos*, en contraste, consiste en la valoración de la salida de un proceso de agrupamiento. A menudo este análisis utiliza un criterio de optimización específico; sin embargo la definición de este criterio es a menudo una tarea subjetiva. De aquí, la poca existencia de estándares bien definidos para esta valoración, excepto en algunos dominios bien definidos. Generalmente existen tres tipos de validación de la salida de un algoritmo de agrupamiento, la primera una *validación externa* se logra tras comparar la estructura de los grupos formados por el algoritmo contra una estructura definida a priori. La *validación interna* consiste en tratar de determinar si la estructura de los grupos formados es intrínsecamente

apropiada para el conjunto de datos. Por último, la *valoración relativa* compara las estructuras de los grupos formados midiendo el mérito relativo de cada una. Para más detalles de sobre la forma de validar la salida de un algoritmo de agrupamiento refiérase a [26, 17].

2.3.3. Agrupamiento Estrella

Como se mencionó antes, los algoritmos de agrupamiento particionales generan como salida una única estructura de grupos a partir de una colección de entrada. La principal ventaja de este tipo de algoritmos es que pueden trabajar con grandes conjuntos de datos.

El algoritmo de *agrupamiento estrella* es un algoritmo que ha mostrado evidencia positiva sobre la verdad de la hipótesis del grupo[6] [3]. Este es un algoritmo que induce de manera natural el número de grupos y la estructura de los temas dentro del espacio de textos. Es un algoritmo de tipo particional, y que dentro de la figura 2.3 cae en la clasificación denominada como "Teoría de grafos".

En las siguientes secciones se describe de manera más detallada, el funcionamiento del algoritmo de agrupamiento estrella en su versión "fuera de linea", el cual fue el que se utilizó para nuestro sistema generador de resúmenes de múltiples documentos.

Agrupando Datos Estáticos con Sub-grafos en forma de Estrella

Formulemos el problema de la siguiente manera, la representación de nuestro sistema de información será por medio de un *grafo de similaridad*. Un grafo de similaridad es un grafo no dirigido, ponderado $G = (V, E, w)$ donde los vertices en el grafo corresponden a los textos y cada arista ponderada corresponde a la similaridad que existe entre dos textos. Nosotros medimos la similaridad entre textos utilizando métricas estándar utilizadas dentro de la comunidad de recuperación de información, por ejemplo, la medida cosenoidal (Para conocer las diferentes métricas refiérase a [21]).

Para poder aplicar cualquiera de estas medidas de similitud entre documentos, es necesario trasladar nuestros textos a una forma de representación donde estas puedan ser aplicadas. Para esto utilizamos el modelo vectorial (descrito en la sección 2.2.1), el cual agrega estadísticas sobre la ocurrencia de palabras dentro de los textos. La

[6]El uso del agrupamiento dentro del área de Recuperación de Información surge debido a una hipótesis *(hipótesis del grupo)*, que dice que los documentos relevantes a una petición tienden a ser más cercanos entre ellos que aquellos que no son relevantes a una petición en particular.

premisa del modelo vectorial es que dos textos son similares si usan palabras similares. Como se mencionó antes, este vector se crea a partir de la colección de textos (corpus), asociando cada palabra importante a una dimensión del espacio vectorial. De esta forma los textos son mapeados a vectores n dimensionales de acuerdo a sus frecuencias de palabras. Así, textos similares son mapeados por medio de vectores similares. Continuando con el caso de la medida cosenoidal, en el modelo vectorial la similaridad de dos textos es medida por el ángulo entre los dos correspondientes vectores. La idea básica de la medida cosenoidal es medir el ángulo entre el vector del documento Q y el del documento D, para hacerlo calculamos:

$$\angle(Q, D) = \frac{\sum_{j=1}^{t} w_{qj} d_j}{\sqrt{\sum_{j=1}^{t} (d_j)^2 \sum_{j=1}^{t} (w_{qj})^2}} \tag{2.3.1}$$

Donde j va de 1 a el número total de términos del vocabulario t, w_{qj} indica la frecuencia del termino j en el documento Q y d_j la frecuencia del termino j en el documento D.

G es un grafo completo con aristas de peso variable. Una organización del grafo, que da como resultado grupos confiables de similitud σ (i.e., grupos donde los textos tienen pares de similitud al menos σ), puede ser obtenido de las siguientes formas:

1. Umbralizando el grafo a σ.

2. Aplicando una cobertura de *grupos mínimos* con grupos máximos en el grafo resultante G_σ.

El grafo umbralizado G_σ, es un grafo no dirigido obtenido de G al ir eliminando todas las aristas cuyos pesos fueran menores a σ. La cobertura de grupos mínimos tiene dos características. Primero, al usar grupos para cubrir el grafo de similaridad, estamos asegurando que todos los textos en un grupo tienen el grado de similitud deseado. Segundo, la cobertura de grupos mínimos con grupos máximos permite que los vertices pertenezcan a más de un sólo grupo. Desafortunadamente esta técnica es un problema intratable.

Sin embargo, el problema de la cobertura de grupos mínimos, puede ser aproximado por medio de hacer una cobertura en el grafo de similaridad umbralizado utilizando subgrafos en forma de estrella. Un grafo en forma de estrella de $m+1$ vertices consiste de un *centro de estrella* y de m vertices *satélite*, donde existen aristas entre el centro

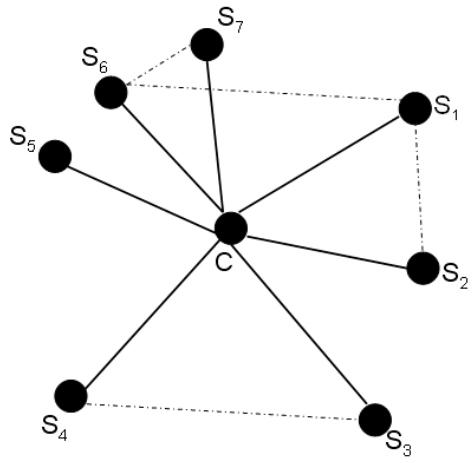

Figura 2.4: La figura muestra un ejemplo de un subgrafo en forma de estrella.

de la estrella y cada uno de los satélites (Ver Figura 2.4).

El Algoritmo Estrella

El algoritmo de agrupamiento estrella puede ser utilizado para organizar textos dentro de un sistema de información. El algoritmo de agrupamiento estrella está basado en una cobertura ávida[7] de el grafo de similaridad umbralizado por medio de subgrafos en forma de estrella. El algoritmo aparece en la tabla 2.3.

Este algoritmo tiene dos características interesantes. La primera es que la cobertura de la estrella no es única. Un grafo de similaridad puede tener diferentes estrellas debido a que cuando hay diferentes vértices del mismo grado, el algoritmo escoge arbitrariamente uno de ellos como el centro de la estrella. La segunda característica es que este algoritmo muestra un codificado simple de la cobertura estrella al ir asignando etiquetas de "centro" y "satélite" a los vértices. Para ver un estudio más detallado sobre las características del algoritmo de agrupamiento estrella refiérase a [3, 20].

[7]El nombre de algoritmos ávidos, también conocidos como voraces (del término inglés "greedy") se debe a su comportamiento: en cada etapa el algoritmo "toma lo que puede o la mejor solución en ese instante" sin analizar consecuencias, es decir, son glotones por naturaleza.

Tabla 2.3: Algoritmo Estrella

Para cualquier umbral σ:

1. Calcular $G_\sigma = (V, E_\sigma)$ donde $E_\sigma = \{e \in E : w(e) \geq \sigma\}$.
2. Poner cada vértice en G_σ inicialmente marcado como *no-visitado*
3. Calcular el grado de cada vértice $v \in V$
4. Tomar el vértice de mayor grado que tenga la etiqueta ‘‘no-visitado’’ como centro de la estrella, y construir un grupo con éste como centro de la estrella y sus satélites con sus vértices asociados. Marcar cada nodo de la estrella recién construida como ‘‘visitados’’.
5. Repetir el paso 4 hasta que todos los vértices estén visitados.
6. Representar cada grupo por medio del texto correspondiente al centro de cada estrella.

Capítulo 3
Estado del Arte

El presente capítulo tiene como objetivo ubicar al lector en el área donde se enfoca el trabajo de tesis, i.e. en el desarrollo de sistemas para la generación automática de resúmenes. Durante la revisión se darán a conocer las principales ramas existentes actualmente, que son los sistemas de generación de resúmenes de un documento y los que generan resúmenes de múltiples documentos; se describirán las arquitecturas generales de ambas tareas con el objetivo de destacar los puntos positivos y negativos de cada una. Se mencionarán las tendencias que han ido adquiriendo cada una de estas ramas de investigación y al mismo tiempo se mencionará como ha ido evolucionando el proceso de evaluación de este tipo de sistemas hasta nuestros días.

3.1. Generación Automática de Resúmenes

Estudios hechos a finales de la década de 1950 y principios de los años 60's sugerían que la tarea de generar de manera automática resúmenes era posible, sin embargo no de una manera directa [24]. Fue necesario un tiempo para que el progreso en el área de Procesamiento del Lenguaje Natural junto con el incremento de poder computacional *(memoria, velocidad)* y además el aumento de textos en línea *(corpus en la Web)* despertara de nuevo el interés en distintos investigadores en la tarea de generar resúmenes.

A pesar de los resultados alentadores, algunas dudas siguen sin ser totalmente resueltas. Un ejemplo claro es, *determinar que tipo de información debe o no de estar presente en un resumen.* Un resumen generado automáticamente, comúnmente denominado "sumario", lo podemos definir de la siguiente manera:

- **Definición:** Un resumen o sumario es un texto producido a partir de uno o más

documentos, que contiene una porción significante de la información contenida en los documentos originales, y además no es más grande que la mitad del documento(s) original.

En la definición anterior, la palabra "documento" involucra diferentes tipos de fuentes, por ejemplo documentos multimedia, documentos Web, hypertextos, y actualmente también es posible considerar secuencias de voz [47]. Varios trabajos de investigación han logrado definir dos tipos de resúmenes de acuerdo a su contenido [24, 25], los **resúmenes indicativos** son aquellos que dan al lector una idea simple de cual es el contenido del documento y los **resúmenes informativos** son los que proveen al lector del contenido esencial del documento.

En particular, dentro del área de generación automática de resúmenes existen dos formas básicas en las cuales los resúmenes son generados. Los **resúmenes basados en Extractos** son los resúmenes que son creados al reutilizar porciones (palabras, oraciones, párrafos, etc.) del documento(s) de origen; el otro tipo de resúmenes denominados simplemente como **resúmenes** ó **Abstractos** los cuales *generan*[1] el resumen final a partir de los extractos obtenidos en un primer paso.[2]

3.1.1. Etapas de un sistema generador de resúmenes

Diferentes trabajos de investigación [24, 25] en el área de generación automática de resúmenes han identificado tres diferentes etapas que son necesarias para la realización de la tarea. Sin embargo, la mayoría de los sistemas actuales llevan acabo simplemente la primer etapa.

- **Identificación del tópico:** como su nombre lo dice consiste principalmente identificar de alguna manera el tema principal del documento(s) origen. Sin importar el criterio de importancia utilizado, una vez que el sistema ha identificado las unidades más importantes (palabras, oraciones, párrafos, etc.), éste puede simplemente listarlas creando de esta forma un resumen basado en extractos

[1] El proceso de generación involucra crear nuevas oraciones a partir de las que han sido extraídas en un primer paso por el sistema de generación de resúmenes. Para poder crear estas nuevas oraciones es necesario contar con sofisticados recursos lingüísticos que interpreten adecuadamente contenido y significado de las oraciones extraídas. Una vez hecha esta interpretación el sistema puede mezclar y/o comprimir oraciones con el objetivo de entregar al usuario un resumen más coherente.

[2] Con el objeto de facilitar la lectura a lo largo del documento de tesis se hará uso de la palabra *"resumen ó resúmenes"* para referirnos a un *"resumen basado en extractos"* a no ser que explícitamente se exprese lo contrario.

o desplegarlas a manera de diagramas produciendo así un resumen esquemático. Típicamente, la identificación del tema se lleva acabo a través del uso de herramientas y/o técnicas complementarias.

- **La interpretación:** El proceso de la interpretación requiere del uso de herramientas o recursos lingüísticos externos (bases de conocimiento externo). El resultado de esta etapa generalmente es algo nuevo(algo que no existe de manera explícita en el documento original), producto de fusionar y/o comprimir porciones de información.

- **La generación:** El resultado de la etapa de *interpretación* generalmente es ilegible para un usuario común. Incluso los resúmenes basados en extractos son raramente coherentes, pues muchas veces omiten o repiten algo del material contenido en el documento de original. El objetivo de esta etapa es crear un texto más coherente fácil de leer para los usuarios. En el caso de los resúmenes basados en extractos el proceso de generación involucraría algo tan simple como ordenar las oraciones extraídas de acuerdo a como aparecieron en el documento original.

3.2. Métodos para la Generación de Resúmenes de un solo Documento

3.2.1. Etapa 1: Identificación del tópico

Para realizar esta etapa la mayoría de los sistemas hace uso de distintos módulos independientes. Cada módulo asigna un puntaje (o peso) a cada unidad (palabras, oraciones, párrafos, etc.); posteriormente un módulo en particular se encarga de hacer la combinación de los diferentes pesos asignados a cada unidad obteniendo un solo valor para cada unidad; finalmente el sistema regresará las n primeras unidades con el peso más alto, tomando en cuenta el tamaño del resumen solicitado por el usuario.

Un problema es la forma de elegir el tamaño de la "unidad" que se utilizará para el proceso de extracción. La mayoría de los sistemas utilizan oraciones completas. El desempeño de esta etapa es generalmente medido por medio de Precisión y Recuerdo (ver capítulo 2). Dado un documento de entrada, un resumen basado en extractos producido por un humano, y el resumen basado en extractos producido por el sistema,

estas medidas dicen que tan cerca estuvo el sistema de producir una salida igual a la producida manualmente. Para cada unidad, se define como *correctas*= a el número de oraciones extraídas por el sistema y por el humano; *incorrectas*= es el número de oraciones extraídas por el sistema pero no por el humano; y *olvidadas*= el número de oraciones extraídas por el humano pero no por el sistema. Así entonces tenemos:

$$P = \frac{correctas}{(correctas + incorrectas)} \tag{3.2.1}$$

$$R = \frac{correctas}{(correctas + olvidadas)} \tag{3.2.2}$$

de esta forma decimos que la Precisión refleja cuantas de las oraciones extraídas por es sistema fueron buenas, y el Recuerdo refleja cuantas de las oraciones buenas olvido el sistema.

A continuación se describen las técnicas más utilizadas para la realización de la identificación del tópico.

- *Posición.* Gracias a las regularidades en la estructuras de los textos de diferentes dominios, ciertas secciones de los documentos (encabezados, títulos, párrafos iniciales, etc.) tienden a contener información relevante. Un método muy simple que compite con los mejores sistemas, consiste en tomar los primeros párrafos de un documento para generar el resumen. Esta técnica funciona muy bien en artículos de noticias [11]. Algunas variaciones sobre el uso de este tipo de criterio para hacer la selección de oraciones importantes aparece en [18, 29, 42, 45].

- *Palabras clave.* En algunos documentos es común el uso de palabras o incluso frases para introducir conceptos importantes (e.g., "en este artículo se muestra", "en conclusión", etc.), de esta forma las oraciones que contiene dichas frases o palabras deberían ser extraídas por el sistema. En [51] se reportan valores del 54 % en precisión y recuerdo utilizando una lista de 1,423 palabras/frases clave para la identificación de oraciones importantes dentro de artículos científicos.

- *Frecuencia de palabras.* El trabajo presentado por Luhn en [34] utiliza la ley de distribución de las palabras de Zipf[3] para establecer el siguiente criterio

[3]La ley de Zipf establece que en uno o varios documentos se cumple que: un conjunto pequeño de palabras aparecerán de manera muy frecuente dentro del documento, un conjunto de palabras de mayor tamaño aparecerán con una frecuencia menor y por último un conjunto grande de palabras aparecerán con muy poca frecuencia.

de extracción: Si una oración contiene palabras que sobrepasan algún umbral de repetición , entonces esta oración contiene muy probablemente información relevante.

Los sistemas propuestos por Luhn [34], Edmundson [18], Kupiec *et. al.* [29], Hovy y Lin [25], junto con otros más [15, 42] utilizan diferentes medidas para determinar la frecuencia de las palabras y en sus trabajos se reporta entre el 15 % y 35 % en precisión y recuerdo (utilizando únicamente este atributo como criterio de extracción). En algunos trabajos se muestra como la combinación de este atributo junto con otros más resulta en un sistema con mayores niveles de precisión y recuerdo [29, 51].

- *Similitud con el título o con una petición.* Un atributo muy simple pero útil para determinar la relevancia de las oraciones es por medio de calcular la similitud de las oraciones con el título del documento o contra un conjunto de palabras claves proporcionadas por el usuario [15, 29, 25, 42, 51].

- *Conexión léxica.* Dentro de un documento, las palabras o incluso las oraciones se encuentran conectadas de alguna forma, por medio de repeticiones, coreferencias, sinónimos y por asociaciones semánticas que pueden ser expresadas en un tesauro. De esta forma, el nivel de relevancia de las oraciones puede ser determinado por medio de calcular el grado de conectividad de las palabras que la conforman contra el resto del documento. Algunos resultados reportados utilizando este tipo de técnicas van desde un 30 % [13] (utilizando un criterio de conectividad muy estrictos) hasta un 60 % [7].

- *Estructura del discurso.* Este criterio es una variación sofisticada del punto anterior. Este criterio consiste en determinar la estructura fundamental del texto y de esta forma ponderar las oraciones de acuerdo a la cercanía que tengan con la idea central. Este tipo de técnicas involucra el uso de recursos lingüísticos muy sofisticados y costosos, para ver un sistema que hace uso de este tipo de métodos refiérase a [35, 36].

- *Combinación de varios módulos.* En todos los casos, los resultados han permitido concluir que ningún método por si solo es capaz de desempeñarse tan bien como lo haría un humano en la tarea de crear resúmenes basados en extractos. Sin embargo, cada método muestra evidencia de que la tarea es realizable, razón

por la cual la combinación de estos mejora el desempeño del sistema. El uso de técnicas de aprendizaje automático han adquirido mucha popularidad con el objetivo de tratar de encontrar la mejor combinación de estos criterios.

Algunos trabajos que hacen uso del aprendizaje automático para la resolución de la tarea son:

1. Kupiec [29] entrena un clasificador Bayesiano con los atributos de *posición, palabras/frases clave, frecuencia de palabras, uso de palabras en mayúsculas y la longitud de la oración.* El sistema determina la probabilidad que tiene cada oración de pertenecer al resumen final. En sus resultados reportados, Kupiec muestra que el atributo *posición* por si solo permite al sistema alcanzar niveles de 33 % de precisión, el uso de únicamente las *palabras clave* permite al sistema desempeñarse a un 29 % (pero cuando estos dos son combinados se logra obtener hasta un 42 %). Uno de los problemas de dos de estos atributos es que son dependientes del dominio, i.e. no son fácilmente trasladables a una estructura de documentos diferente. Estos son el uso de *palabras clave* para identificar cuando una oración podría contener o no información importante. El usar este tipo de criterio obliga a trabajar con sólo un determinado tipo de documentos, en particular Kupiec trabaja únicamente con artículos científicos. Más sin embargo utiliza tres atributos que denominamos independientes del dominio, que son la posición y la longitud de las oraciones así como la presencia de nombres propios. Algo que es interesante del trabajo de Kupiec es que aún con tan pocos atributos para representar los textos logra obtener buenos resultados.

2. Chuang *et. al.* [15] evalúa el desempeño de distintos algoritmo de aprendizaje, pero a diferencia de Kupiec, él representa a las oraciones por medio de 23 atributos, a los cuales divide en tres grupos. Al primer grupo formado de cinco atributos los llama atributos no estructurales del texto, por ejemplo frecuencias de términos, similaridad con el titulo del documento, etc. los cuales son los únicos que podemos decir son independientes del dominio y del lenguaje. El segundo grupo de 15 atributos los denomina relaciones retóricas. Este conjunto de atributos que podríamos también considerarlos independientes del dominio y del lenguaje tienen la desventaja de necesitar

gran cantidad de conocimiento externo para poder ser calculados, pues involucra resolver problemas del lenguaje, por ejemplo encontrar la antítesis, la causa, circunstanciales, etc. de cada oración. Y el último grupo de tres atributos tiene que ver con estadísticas tomadas del grupo de atributos anterior.

3. Neto *et. al.* [42] utiliza 13 atributos en su sistema. De los cuales sólo cuatro de ellos son considerados independientes del dominio, los cuales son el valor centroide de las oraciones, la longitud y posición de las oraciones, al igual que la similaridad con el título y la presencia de nombres propios. El resto de los atributos, al igual que en el caso de Chuang, son atributos que requieren de conocimiento externo para poder ser calculados.

3.2.2. Etapa 2: Interpretación

La etapa de interpretación es la que diferencia a los sistemas que sólo crean *resúmenes basados en extractos* de aquellos que crean *resúmenes o abstractos*. Durante la interpretación, las unidades identificadas previamente como importantes son fusionadas, representadas en nuevos términos, y expresadas usando una nueva formulación, para lo cual es necesario utilizar conceptos o palabras que no se encuentran en el documento original.

Ningún sistema es capaz de realizar una interpretación sin contar con una base de conocimientos especializada en el dominio de el documento(s) original(es). Actualmente la etapa de interpretación se encuentra todavía muy limitada, pues la adquisición y/o creación de conocimiento es una tarea muy complicada, la cal es una de las razones por las que esta etapa no se realiza en nuestro sistema propuesto. Para un estudio más extenso sobre algunas técnicas utilizadas para la realización de esta etapa, refiérase a [8, 25].

3.2.3. Etapa 3: Generación

El objetivo de esta etapa es la de proporcionar al usuario final un resumen coherente y de fácil lectura. Para hacer eso es necesario contar con técnicas sofisticadas para la generación de lenguaje natural. Sin embargo, como se mencionó anteriormente, el sistema propuesto en este trabajo de tesis no alcanzará la etapa de interpretación

y consecuentemente tampoco la etapa de generación. Para un estudio más extenso sobre algunas técnicas utilizadas refiérase a [5, 28].

3.3. Métodos para la Generación de Resúmenes de Múltiples Documentos

La tarea de realizar resúmenes de un solo documento ya es muy compleja por si sola. Pero la tarea de realizar el resumen de una colección de documentos relacionados temáticamente posee varios retos adicionales. El proceso de generar un resumen de múltiples documentos consiste en la creación de un resumen simple de un conjunto de documentos relacionados temáticamente. Existen tres grandes problemas que surgen al momento de manejar múltiples documentos: (i) reconocer y resolver redundancias, (ii) identificar diferencias importantes entre los documentos, y (iii) asegurar la coherencia del resumen, tomando en cuenta que diferentes porciones de información provienen de diferentes fuentes.

Uno de los primeros métodos propuestos para resolver el problema de generar resúmenes de múltiples documentos propone el uso de técnicas de extracción de información para facilitar el proceso de identificación de similaridades y diferencias entre documentos [40]. Al igual que como sucedería en el caso de un solo documento, los resúmenes generados con esta técnica tienden a contener únicamente ciertos tipos de información predefinida, i.e., contendrían únicamente la información que le interesa al sistema de IE. Trabajos más recientes mezclan las técnicas de extracción de información con métodos de regeneración de texto para aumentar la calidad de la información contenida en los resúmenes creados [46].

Comúnmente se utilizan medidas de similaridad para lograr la identificación de redundancias en los documentos. Una técnica muy común consiste en medir la similaridad entre cada par de oraciones y posteriormente un proceso de agrupamiento es aplicado con la finalidad de encontrar los temas comunes dentro de la colección de documentos [37, 39, 44]. Una vez hecha la identificación de la información similar, esta debe de ser incluida en el resumen. Más allá de simplemente listar estas porciones de información (como se hace al trabajar con un documento), es necesario seleccionar las oraciones más representativas asegurando así que la mayor cantidad de información esta siendo considerada para el resumen [44].

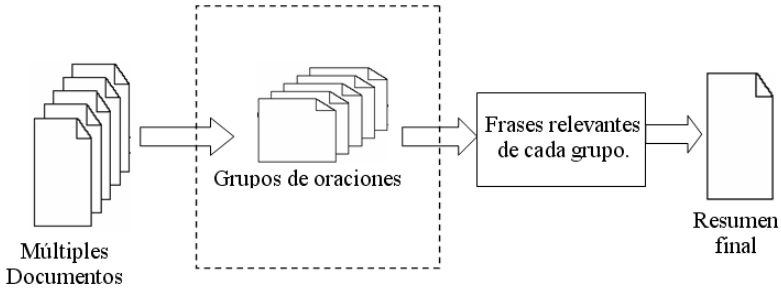

Figura 3.1: Arquitectura general de un sistema generador de resúmenes de múltiples documentos

Asegurar la coherencia en el resumen final es una tarea difícil, pues en principio se requiere de un cierto nivel de entendimiento del contenido de cada oración además de conocimiento sobre reglas del discurso. Debido a la complejidad que esto involucra, actualmente muchos de los sistemas simplemente siguen la línea del tiempo para ordenar el contenido del resumen y en algunos casos una etiqueta indicando la fecha del evento acompaña a cada oración [32]. La figura 3.1 muestra de manera gráfica la arquitectura general de un sistema generador de resúmenes de múltiples documentos. Dentro de esta arquitectura general se puede observar los siguientes pasos generales a seguir:

- **Agrupación.** El proceso de agrupación tiene como principal objetivo permitir al sistema dividir la colección inicial en sus diferentes sub-temas. Esto permite identificar las similaridades entre documentos además de la información única dentro de cada un de ellos.

- **Identificación de información representativa.** Una vez que los diferentes sub-temas han sido clasificados, es necesario contar con un método de selección de los elementos más representativos de cada grupo. El objetivo es abarcar la mayor cantidad de información posible con el menor número de oraciones.

- **Generación.** Al igual que en la tarea de crear el resumen de un solo documento, el proceso de generación consiste en crear un documento coherente. Como se menciono antes, en muchos casos donde la etapa de generación no es alcanzada, las oraciones son ordenadas de acuerdo a la línea del tiempo.

3.4. Evaluando la calidad de los resúmenes

La tarea de evaluar la calidad de los resúmenes producidos de manera automática siempre ha sido un problema para los grupos de investigación dentro de esta área. La evaluación de un resumen se vuelve una tarea muy subjetiva debido a que no existe un resumen ideal contra el cual se pueda comparar la salida de un sistema. En las próximas secciones se describen algunas de las técnicas de evaluación utilizadas comúnmente por diferentes grupos de investigación, cabe mencionar que dichas técnicas han ido evolucionando junto con los sistemas de generación de resúmenes.

3.4.1. Estudios previos

Dentro del área de generación automática de resúmenes existen dos tipos de evaluación. Evaluación **intrínseca** que mide la calidad de la salida y **extrínseca** que miden la ayuda o asistencia que la salida proporciona al usuario en el desempeño de una tarea particular.

La mayoría de los sistemas de evaluación son intrínsecos. Típicamente los evaluadores crean un conjunto de resúmenes ideales, uno por cada documento de prueba, después comparan la salida del sistema con éste, midiendo el traslape del contenido. Este tipo de evaluación se puede hacer por medio de Precisión y Recuerdo [15, 29, 42]. Dado el hecho de que no existe un "resumen ideal", algunos evaluadores utilizan más de un resumen generado por humanos por cada documento de prueba, y promedian el puntaje obtenido por el sistema a través del conjunto de resúmenes ideales[4].

Un segundo método de evaluación intrínseca es el de contar con una escala de evaluación que miden la calidad de los resúmenes basándose en su legibilidad; cantidad de información contenida, fluidez y cobertura [45].

Por otro lado, las medidas de evaluación extrínsecas son fáciles de aplicar. Sin embargo, el problema es saber que tan bien éstas correlacionan la calidad del resumen generado con el desempeño de alguna tarea en particular. Un ejemplo de una medida de evaluación extrínseca es dentro de la categorización de textos (Para ver en detalle el objetivo de la tarea vea la sección 2.2). Ésta consiste en un primer paso en clasificar los documentos completos, y en un segundo paso estos documentos son

[4]Este tipo de evaluación es el más utilizado, en particular por aquellos sistemas que se basan en técnicas de aprendizaje automático.

resumidos automáticamente y nuevamente vuelven a ser clasificados. Al final se revisa el acuerdo que tuvo el sistema al momento de clasificar los documentos completos con la clasificación asignada a los documentos resumidos. Entre mayor sea el acuerdo, mejor se considera el resumen generado.

3.4.2. Dos medidas básicas

Mucha de la complejidad en la tarea de evaluar los resúmenes generados automáticamente surge debido a la dificultad de especificar qué es lo que realmente se quiere medir, y porqué, i.e. no existe una formulación clara de lo que debe contener la salida de un sistema generador de resúmenes. Aquí mencionamos algunos puntos que deben ser considerados.

En general, para considerarse un resumen, éste debe de cumplir dos requerimientos:

- Debe ser un texto más corto que el texto de entrada.
- Debe de contener la información importante existente dentro del texto original (donde la importancia es definida por un experto).

A partir de esto, se pueden definir dos medidas que capturan lo que un resumen S debe de ser con respecto a un texto T:

- Razón de compresión
$$CR = \frac{longitud\ S}{longitud\ T} \tag{3.4.1}$$
- Razón de Retención
$$RR = \frac{info\ en\ S}{info\ en\ T} \tag{3.4.2}$$

Independientemente de la forma en que se mida la longitud y la cantidad de información contenida en los textos, un buen resumen es el cual logra tener una CR pequeña (i.e., tiende a cero) mientras que su RR es grande (i.e., tiende a la unidad). La figura 3.2 muestra los comportamientos típicos que un sistemas de generación de resúmenes puede tener.

Midiendo la longitud. Medir la longitud de un texto es un proceso directo; se puede medir al contar el número de palabras contenidas, el número de letras, el número de oraciones, etc.

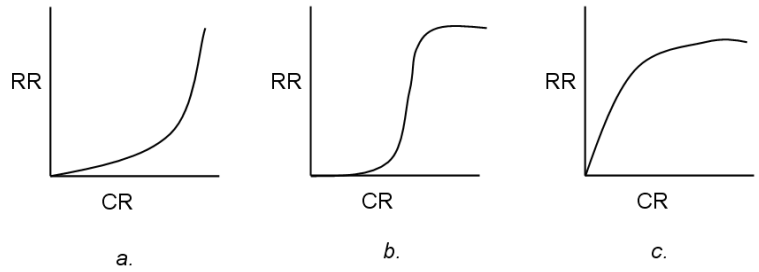

Figura 3.2: Razón de compresión VS Razón de Retención.

Midiendo la cantidad de información. Idealmente, quisiéramos medir no sólo la información contenida en el resumen, sino la cantidad de información interesante dentro de ese resumen. Esto resulta ser una tarea complicada, para la cual se han propuesto diferentes métodos:

1. *El juego del experto:* Un conjunto de personas expertas en la temática marcan la información más importante dentro de los textos. De esta forma es posible medir la precisión y el recuerdo. Esta técnica es de las más populares dentro del área de generación de resúmenes.

2. *El juego de la clasificación:* Refiérase a la sección 3.4.1 para entender como funciona este tipo de evaluación. Este tipo de evaluación extrínseca es de los más populares dentro del área de generación de resúmenes.

3. *El juego Shannon:* En teoría de la información [49], la cantidad de información contenida en un mensaje es medida por medio de $-p \log p$, donde p es la probabilidad de que el lector adivine el mensaje (o cada pieza de él de manera individual). Así, para medir la cantidad de información contenida en S relativa a su correspondiente texto T, se realizan tres tipos de evaluadores. Cada sistema de evaluación debe crear T adivinando su contenido letra por letra. El primero lee T antes de empezar, el segundo lee S antes de empezar, y el tercero no lee nada. Para cada evaluador, se almacena el número de adivinaciones incorrectas $g_{equivocadas}$ y el total de adivinadas g_{total}, de forma que se calcula $R = g_{equivocadas}/g_{total}$. La calidad de S puede ser calculada al comparar los resultados de cada uno de los evaluadores. R_S mide que tanto un evaluador pudo adivinar a partir de S, mientras que R_T mide que tanto el evaluador tiene

todavía que adivinar, incluso con "perfecto" conocimiento previo. Entre más cercanos estén R_S y R_T, de mejor calidad se considera el resumen.

4. *El juego de la pregunta:* Este tipo de evaluación determina la calidad de la información contenida en S al contar la cantidad de preguntas (previamente formuladas a través de T) que es posible responder con S.

Actualmente la evaluación automática de resúmenes es un concepto que esta provocando el interés de muchos grupos de investigación. Es claro que cuando un resumen "ideal" basado en extractos sea creado por un humano, los sistemas que generan resúmenes basados en extractos serán fáciles de evaluar. Recientemente el uso de una variante del método BLEU [43] (Bilingual Evaluation Understudy) ha sido utilizado para evaluar de manera automática sistemas de generación de resúmenes. BLEU es un método de evaluación que determina la calidad de la traducción hecha automáticamente por medio de una combinación lineal entre los n-gramas contenidos en el texto producido por el sistema y los n-gramas encontrados en el texto producido por un humano. Sus creadores han dado el nombre de ROUGE [6, 33] a esta técnica de evaluación, y aunque es una técnica muy prometedora y ampliamente utilizada en nuestros días, todavía no es considerada como suficiente.

Capítulo 4

Generando Automáticamente el Resumen de un Documento

En este capítulo se describe con mayor detalle el sistema propuesto para la generación de resúmenes de un sólo documento. La principal característica del método es que funciona bajo un esquema supervisado. Y aunque es un método supervisado, nos interesa que sea un método que utilice un conjunto suficiente de atributos, los cuales sean independientes del dominio y además independientes del lenguaje. Estos atributos deben ser capaces de aportar al clasificador toda la evidencia necesaria para que éste pueda determinar la existencia de información relevante.

Para poder cumplir estos objetivos se hizo un extenso estudio sobre diferentes sistemas (Ver capítulo 3). Esta revisión nos permitió identificar un conjunto de atributos que son considerados independientes del dominio y del lenguaje, con los cuales se construyó un primer clasificador. Para poder evaluar el desempeño del sistema fue necesario trabajar con conjunto de datos etiquetado, de esta forma el medir precisión y recuerdo se vuelve tarea fácil.

Los resultados de esta primer configuración no fueron los esperados, pues por medio de un tipo de evaluación extrínseca se pudo observar que los resúmenes que se estaban generando eran de calidad menor en comparación con aquellos que estaban siendo generados a través del método base líder en el área. A partir de estos resultados se concluyó que la forma de representación utilizada por el sistema de aprendizaje no era la adecuada, se optó entonces por utilizar las palabras como la nueva forma de representación [53]. Esta representación a diferencia de la primera, considera la información del contexto para determinar si las oraciones de un documento pueden o no ser importantes, de aquí que en experimentos posteriores se utilizaron secuencias de palabras (n-gramas) para la representación de las oraciones. La misma técnica

de evaluación fue aplicada a los resúmenes generados con estas nuevas formas de representación de las oraciones, dando como resultado resúmenes de mayor calidad.

En la primer sección de este capítulo se describe con mayor detalle las características de la arquitectura propuesta. En esta misma sección se describen los atributos utilizados para cada una de las diferentes representaciones utilizadas por el modelo de aprendizaje. En la segunda sección se describen los conjuntos de datos utilizados para la etapa de experimentación. Posteriormente un tipo de evaluación extrínseca es aplicado a los resúmenes obtenidos con el objetivo de determinar la calidad de su contenido. Por último se presenta una sección donde se discute sobre los resultados obtenidos.

4.1. Arquitectura Propuesta

La parte fundamental de la arquitectura propuesta se compone de un clasificador, i.e. el uso de herramientas de aprendizaje automático. En la figura 4.1 se muestra un diagrama a bloques de los componentes de nuestra arquitectura.

La idea básica de este esquema es que un proceso inductivo automáticamente construya un clasificador por medio de observar las características de un conjunto de documentos previamente resumidos, i.e. lo que se le da al algoritmo de aprendizaje son pares $(documento, resumen)$. De tal forma que el problema de generación de resúmenes se convierte en una actividad de aprendizaje supervisado(Sección 2.1).

El proceso de indexado es como ya se ha mencionado antes, un proceso en el cual la idea es mapear los documentos a una representación que sea fácil de interpretar para el algoritmo de aprendizaje. Dentro de este proceso de indexado podemos tener diferentes subprocesos, eliminación de palabras vacías, un lematizador, etiquetador de partes de la oración, etc.

El proceso de aprendizaje consiste en entrenar a un clasificador, en este caso utilizando Naïve Bayes, tomando como entrada los pares $(documento, resumen)$. Idealmente el clasificador será capaz de aprender las características que diferencian las oraciones que pertenecen a un resumen de las que no.

El proceso de clasificación consiste en generar los resúmenes de los documentos nuevos tomando como modelo lo aprendido en el proceso de aprendizaje. La salida serán entonces documentos cortos que equivalen al resumen de cada uno de los documentos de entrada.

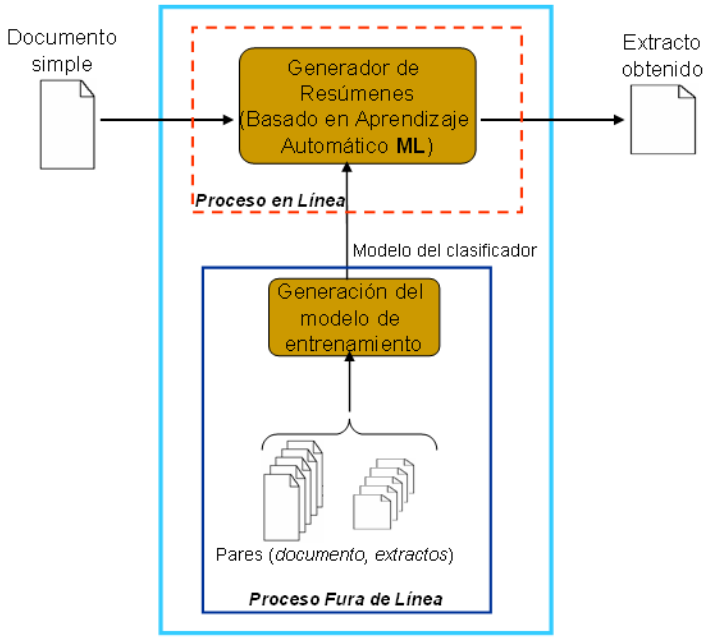

Figura 4.1: Arquitectura general del sistema.

4.1.1. Atributos Estadísticos-Heurísticos

Con base en el análisis hecho a los diferentes trabajos en el capítulo 3, se logró recolectar un conjunto de atributos clasificados como atributos independientes del dominio y del lenguaje. Recordemos que dado que se quiere contar con un sistema que funcione en cualquier dominio temático y que además se fácilmente trasladable a otro idioma, se decidió trabajar únicamente con este conjunto de atributos. Este conjunto de atributos calculados a base de datos estadísticos o simples heurísticas nos permitieron definir una primera configuración para el modelo de aprendizaje. El total de atributos estadísticos/heurísticos que se utilizaron en esta primer configuración fueron seis:

1. **Centroide:** El valor centroide es una medida que indica que tan frecuentes son las palabras de un documento. Se le atribuye importancia a este atributo pues utilizándolo es posible saber que tan cercanas son las oraciones al tema principal de un documento. Pensemos en el centroide como un vector muy grande, cada componente del vector es una de las palabras que aparecen en el documento. Sea D el documento, y $|D|$ el número de oraciones en el documento. Este vector,

definido como el centroide del documento, se describe por medio de:

$$Centroide(D) = (v_{w_0}, v_{w_1}, v_{w_2}, \ldots, v_{w_n}) \tag{4.1.1}$$

Donde:

$$v_{w_i} = \frac{TF(w_i)IDF(w_i)}{|D|} \tag{4.1.2}$$

$TF(w_i) = f(w_i, D)$ es conocido como "Frecuencia de Términos", y se define como el número de ocurrencias de w_i en el documento D. $IDF(w_i)$ es conocida como "Frecuencia Inversa", y se calcula de la siguiente forma:

$$IDF(w_i) = \log \frac{|D|}{\#oraciones\ en\ D\ que\ contienen\ w_i} \tag{4.1.3}$$

IDF es una medida que indica que tan "rara" es la palabra w_i dentro del documento D. Entre más "rara" sea la palabra se le considerará más indicativa o informativa. Sea entonces S_i donde i denota la i-esima oración en el documento D, Radev *et. al.* [45] define al valor centroide de S_i como la suma normalizada de los componentes del centroide:

$$C(S_i) = \sum_{w \in S_i} v_w \times f(w, S_i) \tag{4.1.4}$$

2. **Posición:** Este atributo es considerado importante debido al hecho de que en diferentes tipos de documentos, la información más importante se da en las primeras oraciones del mismo, i.e. se da la idea principal en los primeros párrafos y en el resto del documento simplemente esta idea es desarrollada. Para cada oración S_i en el documento D, el valor de la posición de esta oración se calcula por medio de:

$$P_i = \frac{|D| - i + 1}{|D|} \times C_{max} \tag{4.1.5}$$

Donde C_{max} es el valor centroide máximo obtenido en el paso anterior. De esta forma, la primera oración en un documento obtiene el mismo valor que C_{max}.

3. **Similaridad con el título:** Varios sistemas hacen uso de palabras clave para poder identificar oraciones relevantes [29]. Pero como sabemos, es imposible definir un conjunto de palabras clave que apliquen a todo tipo de documentos,

pues el hacerlo obligaría al sistema a depender del dominio temático y aún más del lenguaje. Con lo que si contamos es generalmente con un título. A menudo el título de un documento da gran información sobre el contenido del mismo. Esta es la razón por la cual se decide utilizar éste atributo. Cuando no existe título, la primera oración del documento es tomada como el título y así de esta forma las oraciones con mayor similaridad al título son consideradas más importantes. Este valor es calculado por medio del producto punto del vector de la oración actual S_i y el vector de la primera oración del documento (S_1). Los vectores de las oraciones son representaciones n-dimensionales de las palabras en cada oración, i.e. el vector en la posición j tiene un valor que equivale al número de veces que ese término ocurre en la oración[1]. Sea:

$$\langle \overrightarrow{S}_i, \overrightarrow{S}_j \rangle = \sum_{w \in S_i \cap S_j} f(w, S_i) \cdot f(w, S_j) \tag{4.1.6}$$

El cual es el producto punto entre dos vectores; donde $f(w, S)$ es la frecuencia del término w dentro de la oración S. Entonces el traslape o la similaridad con el título se calcula de la siguiente forma:

$$F_i = \frac{\langle \overrightarrow{S}_i, \overrightarrow{S}_1 \rangle}{\langle \overrightarrow{S}_1, \overrightarrow{S}_1 \rangle} \tag{4.1.7}$$

El cual es producto de los vectores normalizado, así entonces al evaluar la primera oración (i.e., $i = 1$) el valor de F_1 será igual a 1.

4. **Presencia de Cantidades Numéricas:** Este es un atributo de tipo booleano que fue considerado debido a la existencia de muchos datos numéricos dentro de los documentos con los cuales se trabajó. Generalmente los autores indican por medio de números información considerada importante. Por ejemplo, en un documento que hable de algún desastre natural, los daños materiales, pérdidas humanas, personas afectadas, etc. son datos que se dan por medio de un número. La forma en que se define el valor de este atributo es:

$$Q_i = \begin{cases} 1, & \text{si } S_i \text{ contiene cantidades numéricas} \\ 0, & \text{en otro caso} \end{cases} \tag{4.1.8}$$

[1]En la sección 2.2.1 se explica con mayor detalle la forma de representación vectorial

5. **Nombres Propios:** Igual que el anterior este es un atributo de tipo booleano. La razón por la que se utiliza es debido a que la presencia de nombres refiriéndose a personas y/o lugares son pistas que indican que la oración podría contener datos importantes. Siguiendo el ejemplo de la noticia de un desastre natural, la oración que contenga el nombre del lugar donde ocurrió el desastre podría ser importante, lo cual le da la posibilidad de pertenecer al resumen final. La forma en que se calcula este atributo es:

$$P_i = \begin{cases} 1, & \text{si } S_i \text{ contiene nombres propios} \\ 0, & \text{en otro caso} \end{cases} \tag{4.1.9}$$

6. **Longitud de la oración:** Este atributo es empleado para penalizar aquellas oraciones que sean demasiado cortas, dado que este tipo de oraciones no se espera que pertenezcan al resumen [29, 42]. Se utiliza como atributo la longitud de la oración normalizada, esto es:

$$L_i = \frac{\#num\ de\ palabras\ en\ S_i}{\#num\ de\ palabras\ de\ S_i\ más\ larga} \tag{4.1.10}$$

4.1.2. Secuencias de Palabras como Atributos

El uso de atributos *estadísticos-heurísticos* permite en efecto que el sistema de generación de resúmenes sea portable; i.e. se pueden aplicar a otro dominio temático e incluso a otro idioma sin mucho esfuerzo, permitiendo un buen desempeño. En los experimentos realizados se demuestra la validez de esta afirmación de manera empírica. Al estudiar los resultados obtenidos con esta primer configuración se pudo notar que la exactitud del clasificador se mantiene por niveles debajo del 75 %, lo cual, aunque el sistema es muy preciso en la mayoría de los casos, nos dice que sigue faltando información importante por encontrar, i.e. todas las oraciones relevantes no están siendo correctamente recuperadas[2].

A continuación se listan tres razones que pueden estar causando el mal comportamiento del clasificador:

- *Distribución.* Como es posible observar en la Tabla 4.1 en ambos conjuntos de datos estamos hablando de que sólo aproximadamente el 30 % de las instancias

[2]Ver tablas 4.2 y 4.3 en la Sección 4.2.2.

pertenecen a la clase que nos interesa (i.e. oraciones relevantes), si la distribución de los datos fuera uniforme, podríamos aspirar a tener un mejor desempeño del sistema. Desafortunadamente, la misma naturaleza de la tarea obliga a este tipo de distribución de las clases.

- *Etiquetado.* Otra posible causa es que los conjunto de datos estén mal etiquetados, permitiendo la presencia de inconsistencias en los datos. Lamentablemente no tenemos una forma correcta de erradicar este problema, dado que un conjunto de expertos fueron los encargados de identificar oraciones relevantes y las no-relevantes, permitiendo esto que la tarea de etiquetado se vuelva una tarea muy subjetiva.

- *Representación.* Una última razón por la cual el sistema pude estar comportándose tan deficientemente es debido a una mala representación de las oraciones. Aunque los atributos que se están utilizando son independientes del dominio y del lenguaje, dados los resultados obtenidos concluimos que son muy pocos atributos y que además no es posible capturar con estos la evidencia necesaria para que el clasificador pueda determinar cuando una oración contiene o no información relevante. Debido a que las dos razones expuestas anteriormente son problemas muy difíciles de erradicar surge la necesidad de buscar una forma de representación de las oraciones diferente que siga cumpliendo con nuestros objetivos (atributos independientes del dominio y del lenguaje) y que además ayuden al clasificador a tener un mejor desempeño.

Nuestra propuesta es entonces utilizar las palabras como modo de representación de forma que sea posible aumentar la flexibilidad del sistema generador de resúmenes y al mismo tiempo ir disminuyendo la dependencia del dominio como del lenguaje. En particular proponemos el uso de n-gramas (secuencias de n palabras consecutivas, ver sección 2.2.1) como los atributos de las oraciones. Así, en nuestro modelo cada oración será representada por un vector que contendrá un atributo booleano por cada n-grama que ocurra en el conjunto de entrenamiento.

Este tipo de representación ha sido utilizada ampliamente dentro de diferentes tareas de procesamiento de textos. En particular dentro de clasificación de textos (Sección 2.2) el uso de la bolsa de palabras (BOW) o 1-gramas como forma de representación corresponde a la técnica que mejores resultados ha proporcionado [48].

Sin embargo, existen diferentes estudios que se han realizado con el objetivo de determinar el efecto de la generalización de esta técnica a través del uso de secuencias de palabras como atributos para representar los textos [19, 10, 14]. Estos estudios indican que el uso de n-gramas no aumenta de manera considerable el desempeño de la tarea de clasificación de textos.

A pesar de los resultados poco favorables del uso de n-gramas dentro de la tarea de clasificación de textos, nosotros creemos que éstos pueden ser útiles dentro de la tarea de generación automática de resúmenes. Esta hipótesis se basa en que:

- Por un lado tenemos que las oraciones son mucho más pequeñas que un documento, y consecuentemente el clasificador requerirá de información más detallada para poder distinguir entre oraciones relevantes y no relevantes. Por ejemplo, dentro de clasificación de textos, la simple presencia de la palabra "temblor" dentro de un documento indicaría que éste habla sobre un evento de este tipo. No obstante, no podría ser suficiente información como para seleccionarla como una oración relevante. En una oración que contenga n-gramas como "el-temblor-afecto" o "temblor-de-magnitud" podrían indicar la presencia de información más importante.

- Por otro lado, recientemente grandes avances se han dado en la forma de evaluación de los sistemas de generación automática de resúmenes [6, 33, 31]. Estos trabajos han mostrado que las correspondencias de n-gramas entre los resúmenes producidos automáticamente y los hechos por humanos son un buen indicador de la calidad del resumen y del buen desempeño del sistema. Nuestra propuesta difiere de este tipo de sistemas en que se emplean directamente los n-gramas para la construcción de los resúmenes, i.e., utiliza los n-gramas para seleccionar las oraciones relevantes.

4.2. Experimentos y Resultados

4.2.1. Conjunto de datos

Para los diferentes experimentos se utilizaron dos diferentes conjuntos de datos, uno de ellos en Español mexicano y el otro en Inglés. Ambos conjuntos consisten de artículos de noticias, pero el primero sólo contiene noticias que hablan sobre desastres

naturales, mientras que el segundo considera además de este tipo de noticias también de otros tipos adicionales, por ejemplo política, deportes, economía, etc. La tabla 4.1 resume algunas estadísticas sobre estos conjuntos de datos.

Tabla 4.1: Estadísticas de los conjuntos de datos

Conjunto	Idioma	Dominio	Número de Oraciones	Oraciones Relevantes
DESASTRES	Español	Desastres Naturales	2833	863 (30 %)
CAST	Inglés	General	4873	1316 (27 %)

El conjunto de datos de *Desastres* consiste de 300 noticias recolectadas de diferentes periódicos publicados en México. Cada una de las oraciones fue etiquetada utilizando dos etiquetas básicas: Relevante y No-Relevante. De forma que fuera posible evitar la subjetividad en el proceso de etiquetado, los expertos fueron instruidos para marcar como oraciones "Relevantes" sólo aquellas que contengan al menos un hecho concreto sobre el evento sucedido. Por ejemplo, la fecha o el lugar en el que el desastre natural ocurrió, o el número de personas o casas afectadas, daños económicos, magnitud o escala del evento.

Por otro lado, el conjunto de datos *CAST* (Computed-Aided Summarization,el significado de su siglas en inglés) consiste de 164 artículos de noticias. Al contrario del conjunto de *Desastres*, éste incluye reportes sobre diferentes temas tales como política, economía, deportes, etc. Las oraciones de este conjunto de datos fueron etiquetadas a mano por expertos como *Relevantes* y *No-Relevantes*. Como es posible ver en la tabla 4.1, ambos conjuntos de datos mantienen una distribución similar de oraciones relevantes. Más detalles sobre la forma en que el conjunto de datos CAST fue construido puede revisarse en [22]. El trabajar con datos etiquetados y bajo un esquema supervisado nos permite medir el desempeño del sistema por medio de precisión y recuerdo. La evaluación para todos los experimentos se hizo por medio de la técnica de validación cruzada con 10 pliegues.

4.2.2. Experimentos con atributos estadísticos - heurísticos

El primer bloque de experimentos se realizó bajo la siguiente configuración:

- En un primer modo se utilizó el conjunto de datos "Desastres". Cada instancia (i.e. cada oración de cada texto) fue representada por medio de sus atributos

estadísticos-heurísticos. El clasificador se construyó utilizando el algoritmo de Naïve Bayes.

- En un segundo modo, se utilizó el conjunto de datos "CAST". La configuración del experimento fue la misma que la del punto anterior.

En la tabla 4.2, se muestran los resultados de precisión (p), recuerdo (r) y exactitud (e) del clasificador para los experimentos realizados con el conjunto de datos de DESASTRES. En la segunda columna de la tabla 4.2 se muestra el desempeño obtenido por el sistema (p, r, e) utilizando los atributos de manera individual. La tercera columna de la tabla 4.2 muestra la variación del desempeño del sistema al ir usando sucesivamente los diferentes atributos.

Tabla 4.2: Resultados Experimentos con atributos estadísticos heurísticos para el conjunto de Desastres

Atributos	**Desempeño Individual**			**Desempeño Acumulativo**		
	p	r	e	p	r	e
Centroide	100	69.51	69.51	100	69.51	69.51
Posición	94.56	75.42	74.80	92.27	75.43	73.95
Similaridad	94.61	75.25	74.62	94.06	75.54	74.70
Cantidades numéricas	76.14	78.32	68.77	88.73	78.10	74.87
Nombres Propios	100	69.51	69.51	88.78	78.22	75.01
Longitud	100	69.51	69.51	87.30	78.89	74.94

Al observar los resultados de la tabla 4.2 podemos observar que el atributo que de manera individual aporta mayor información al clasificador; permitiendo esto tener un balance entre la precisión y el recuerdo; es el que mide la *similaridad con el título* y también la *posición*. Por otro lado, la combinación de los seis atributos no resulta en un buen desempeño del sistema, pues a partir de que se agregan el atributo *cantidades numéricas, nombres propios y longitud* la precisión del sistema empieza a decaer. En este punto concluimos que el mejor resultado se alcanza con la combinación de los tres primeros atributos.

La siguiente tabla(Tabla 4.3) muestra los resultados obtenidos bajo el mismo esquema de evaluación pero en esta ocasión se realizaron las pruebas en el conjunto de datos CAST.

Recordemos que el conjunto de datos CAST es un conjunto de noticias en inglés y además las notas son sobre gran variedad de temas, lo cual no sucede con el conjunto

Desastres. El objetivo de este experimento es mostrar la portabilidad de los atributos estadísticos heurísticos entre diferentes dominios y lenguajes.

Tabla 4.3: Resultados Experimentos con atributos estadísticos heurísticos para el conjunto de datos CAST

Atributos	Desempeño Individual			Desempeño Acumulativo		
	p	*r*	*e*	*p*	*r*	*e*
Centroide	100	72.99	72.99	100	72.99	72.99
Posición	100	73	73	100	72.99	72.99
Similaridad	100	73	73	86.50	76.73	71
Cantidades numéricas	100	73	73	86.95	76.76	71.27
Nombres Propios	100	73	73	86.33	76.64	70.81
Longitud	98.98	73.12	72.69	74.36	80.44	68.08

Lo importante a resaltar de estos experimentos es que el sistema se comporta de manera similar. En este caso el atributo que permite tener un mejor balance entre la precisión y el recuerdo es el de *longitud de la oración*. Este, aunque por muy poco ayuda al sistema a identificar mejor aquellas oraciones que pudieran ser más importantes. Y en este caso, la combinación de los cuatro primeros atributos es la que permite al clasificador encontrar aquellas oraciones relevantes.

4.2.3. Experimentos con secuencias de palabras

Tres diferentes experimentos fueron realizados en esta sección. Primero evaluamos el desempeño del sistema utilizando una representación básica, esto es palabras simples como atributos. Un segundo experimento consistió en utilizar secuencias de palabras como forma de representación (i.e. *n*-gramas). Un tercer experimento fue realizado empleando como atributos secuencias frecuentes maximales(SFM).

Primer Experimento: Palabras simples como atributos

En este experimento se utilizó palabras simples como atributos para representar las oraciones. Dado que el espacio original de atributos tiene una alta dimensionalidad, fue necesario aplicar una técnica de reducción de dimensionalidad, en particular se utilizó el muy conocido algoritmo de ganancia de información para seleccionar un subconjunto de atributos con mayor aporte de información (Ver sección 2.2.1). La

tabla 4.4 muestra el número de atributos considerados en este experimento para cada conjunto de datos.

Tabla 4.4: Número de atributos (palabras simples)

	Originales	**Seleccionados**
DESASTRES	8958	530
CAST	10410	612

La siguiente tabla (Tabla 4.5) muestra los resultados obtenidos en este experimento. Es importante resaltar dos cosas para estos resultados (i) la representación propuesta muestra un desempeño similar para ambos conjuntos de datos, indicando esto que la representación cumple con su independencia de dominio y de lenguaje, y (ii) la representación propuesta sobrepasa al desempeño obtenido con el sistema que utiliza atributos estadísticos-heurísticos tanto en precisión, recuerdo y exactitud por aproximadamente 4 %, 9 % y 10 % respectivamente para ambos conjunto de datos.

Tabla 4.5: Evaluación del sistema con la representación palabras simples

	Estadísticos-heurísticos			**Palabras simples**		
	p	r	e	p	r	e
DESASTRES	87.89	78.89	74.94	91.72	87.12	84.82
CAST	74.36	80.44	68.08	88.67	84.39	79.76

Segundo Experimento: Secuencias de palabras como atributos

En este experimento se representó a las oraciones por medio de secuencias de palabras (n-gramas). Específicamente, se consideraron secuencias de hasta tres palabras, i.e., de *unigramas* hasta *trigramas*. Al igual que en el experimento previo, se utilizó el algoritmo de ganancia de información para reducir la alta dimensionalidad del espacio de atributos y así seleccionar un pequeño subconjunto representativo. No se consideraron secuencias mayores de n-gramas, debido a que éstas resultaban ser eliminadas, en su mayoría, por el algoritmo de reducción de dimensionalidad dado que su aporte de información era nulo. La tabla 4.6 muestra el número de atributos considerados para este experimento en ambos conjuntos de datos.

La tabla 4.7 muestra los resultados obtenidos en este experimento. Es posible observar que el uso de n-gramas incrementa la precisión de la clasificación y al mis-

Tabla 4.6: Número de atributos (n-gramas)

	Originales				Seleccionados
	1-gramas	*2-gramas*	*3-gramas*	*Total*	*Total*
DESASTRES	8958	34340	53356	96654	2284
CAST	10410	52745	72953	136108	2316

mo tiempo se mantiene los valores de recuerdo, i.e., no estamos perdiendo oraciones correctamente recuperadas . Este comportamiento es una causa directa de utilizar atributos que son más detallados. Este tipo de atributos permite tener una mejor distinción entre las oraciones relevantes y no relevantes. En particular nos permiten tratar los casos difíciles.

Tabla 4.7: Evaluación del sistema con la representación n-gramas

	Palabras simples			n-gramas		
	p	c	e	p	c	e
DESASTRES	91.72	87.12	84.82	95.53	86.09	86.16
CAST	88.67	84.39	79.76	96.48	84.53	84.54

Tercer Experimento: Secuencias Frecuentes Maximales como atributos

Las secuencias frecuentes maximales son el siguiente paso después de haber utilizado n-gramas en muchas de las tareas de procesamiento de texto. Al observar que el uso de n-gramas en efecto hizo que el sistema se desempeñara de una mejor manera, se optó por representar las oraciones por medio de sus secuencias frecuentes maximales (Ver sección 2.2.1). Uno de los parámetros necesarios antes de calcular las secuencias frecuentes maximales del conjunto de entrenamiento, es el umbral de repetición. Tras haber realizado varias pruebas y observar la tendencia de los resultados obtenidos se decidió que el mejor valor para este umbral sería igual a tres. Dada la alta dimensionalidad de los datos fue necesario aplicar un método de reducción de dimensionalidad, que al igual que en los experimentos anteriores fue Ganancia de Información. La tabla 4.8 muestra el número de atributos seleccionados para la realización de este experimento en ambos conjuntos de datos.

La tabla 4.9 muestra los resultados obtenidos en este experimento. Es posible distinguir que el uso de las secuencias frecuentes maximales no ayuda al sistema pa-

Tabla 4.8: Número de atributos (Secuencias frecuentes maximales)

	Originales	Seleccionados
DESASTRES	5469	1277
CAST	7178	542

ra hacer la adecuada identificación de las oraciones relevantes. En la tabla es posible distinguir que el desempeño aunque mejor al uso de palabras simples no mejora de manera considerable los resultados obtenidos con la representación hecha con n-gramas (En particular para el conjunto de DESASTRES).

Tabla 4.9: Evaluación del sistema con la representación SFM

	n-Gramas			SFM		
	p	c	e	p	c	e
DESASTRES	95.53	86.09	86.16	96.23	87.09	87.48
CAST	96.48	84.53	84.54	96.25	82.17	82.02

Como se puede observar en la tabla 4.9 sólo para el caso del conjunto de datos Desastres los resultados son un poco mayores (aprox 1 % mayores), pero dada la complejidad que implica el cálculo de las secuencias frecuentes maximales y la poca ganancia que estas otorgan al desempeño del sistema, hemos decidido no utilizar este tipo de representación para futuras pruebas.

4.2.4. Evaluación Extrínseca

Método Base

Para tener resultados contra los cuales comparar el desempeño del sistema de generación automática de resúmenes utilizando una representación de los textos por medio de atributos estadísticos-heurísticos, es necesario definir un método base o *baseline* (Término en inglés).

El método base establecido por la comunidad científica que trabaja dentro del área de generación de resúmenes, consiste en tomar las n primer líneas del texto y posteriormente éstas son entregadas como resumen al usuario. Esto se hace debido a la hipótesis que asegura que la información más importante de cualquier documento se encuentra en las primeras secciones de éste.

Para nuestros experimentos tomamos $n = 4$. Este número se eligió pues resultó ser el tamaño promedio de los resúmenes que nuestro sistema estaba generando. El objetivo de calcular este método base es el de comprobar que efectivamente los resúmenes que se están generando con nuestro sistema son de mayor calidad que el obtenido por el método base. Para evaluar esta calidad lo haremos por medio de evaluar el desempeño de otra aplicación (evaluación extrínseca, ver sección 3.4.1). En la sección 4.2.4 se describe con mayor detalle como se llevo acabo esta comparación.

Evaluación con TOPO, un sistema de IE

Dado que estamos trabajando bajo un esquema de aprendizaje supervisado, nos es posible evaluar el desempeño del sistema por medio de precisión, recuerdo y exactitud del clasificador. Sin embargo, estas medidas no nos dicen mucho acerca de la calidad del resumen generado, lo único que indica en que tan bien el clasificador encontró las instancias positivas. En secciones anteriores (Sección 3.4.1), se dijo que, una de las formas de evaluar la calidad de los resúmenes producidos por este tipo de sistemas, es por medio de evaluar el desempeño de otro sistema, en una tarea totalmente diferente. A este tipo de evaluación se le conoce como evaluación extrínseca.

La propuesta es entonces evaluar el desempeño de un sistema de extracción de información al utilizar los resúmenes generados por nuestro sistema contra el desempeño que obtendría si utilizará las noticias completas. Generalmente los sistemas de extracción de información son especializados en algún dominio, por esta razón este tipo de evaluación solo fue posible con el conjunto DESASTRES, el cual es especializado en desastres naturales.

Un sistema de *extracción de información* se ocupa de estructurar información contenida en textos que son relevantes para el estudio de un dominio (o escenario) particular, llamado *dominio de extracción*. En otras palabras, el objetivo de un sistema de extracción de información (IE, por sus siglas en inglés) es encontrar y enlazar la información relevante mientras ignora la extraña e irrelevante.

TOPO es un sistema de IE especializado en el dominio de desastres naturales [50]. Un sistema de extracción de información como TOPO tiene diferentes medidas para poder evaluar su desempeño. Una de estas medidas que es muy conocida en muchas tareas es la medida F (Ecuación 2.2.7).

El procedimiento para realizar la evaluación fue el siguiente, se tomó el conjunto de datos de DESASTRES. En un primer modo las noticias completas fueron dadas

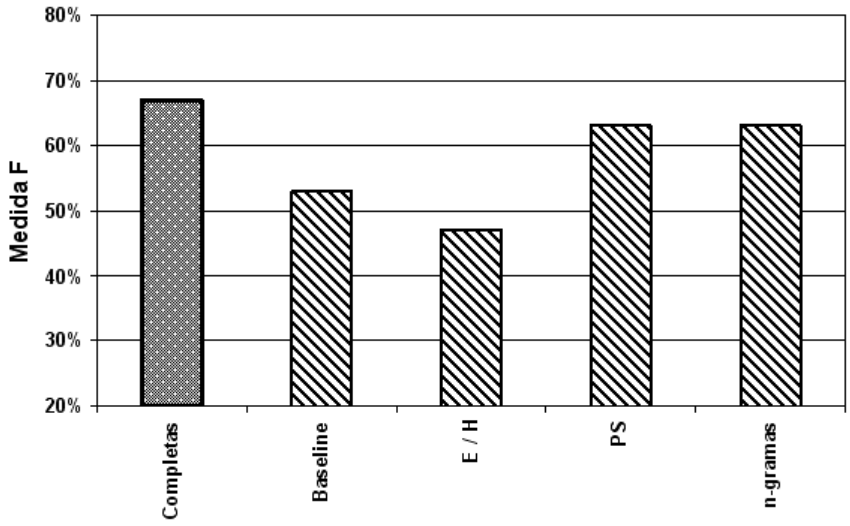

Figura 4.2: Evaluación extrínseca del sistema de generación de resúmenes

como entrada a TOPO sin ningún tipo de procesamiento. El desempeño de TOPO con estos datos en la gráfica de la figura 4.2 equivalen a la barra titulada *completa*. En un segundo modo, se dió como entrada a TOPO las noticias calculadas con el método base definido en la sección 4.2.4 (*Baseline*). En un tercer modo se dió como entrada al sistema TOPO los resúmenes generados utilizando la técnica de representación por medio de atributos estadísticos-heurísticos (*E/H*). En un cuarto modo se dió como entrada a TOPO los resúmenes generados utilizando la técnica de representación por medio de palabras simples (*PS*). Y por último la entrada a TOPO fueron los resúmenes generados utilizando n-gramas como modo de representación (*n-gramas*).

El desempeño del sistema TOPO evaluado por medio de la medida F con estas cinco configuraciones diferentes se muestran en la gráfica de la figura 4.2.

Utilizar las noticias completas apenas le permite a TOPO alcanzar un desempeño del 67 %, mientras que con el método base definido se logra alcanzar un 53 %. Utilizando los atributos estadísticos-heurísticos se logra alcanzar un desempeño del 47 %, con lo cual podemos reafirmar lo dicho anteriormente, que por medio de este tipo de representación **no** es posible para el clasificador poder diferenciar correctamente las oraciones relevantes de aquéllas que no lo son.

Por otro lado, al utilizar una representación por medio de palabras simples y por medio de n-gramas se logra tener un empate en el desempeño de TOPO, alcanzando

un 63 %. Estos resultados nos permiten concluir que efectivamente el representar a las oraciones por medio de n-gramas hace que para el sistema de clasificación sea más sencilla la tarea de identificar las oraciones relevantes.

4.3. Discusión

En este capítulo se abordó la problemática que implica la generación de resúmenes de un solo documento. Nuestra propuesta consiste en crear un clasificador, el cual por medio de observar pares de *(documento, resumen)* hechos por un conjunto de expertos, éste fuera capaz de aprender las características que componen a las oraciones importantes dentro de un documento.

El uso de un método de aprendizaje automático se hizo con el fin de construir un sistema portable a diferentes dominios e incluso a distintos idiomas. Para lograr esto fue necesario hacer un estudio exhaustivo sobre las diferentes técnicas que han sido utilizadas en el pasado para resolver la tarea de generar el resumen de un documento. Algo que se pudo observar en este estudio fue que todos los sistemas revisados hacen uso de atributos que son dependientes del dominio temático e incluso del lenguaje de los documentos. Sin embargo se logró conjuntar diferentes atributos que se consideraron independientes del dominio y del lenguaje a los cuales denominamos atributos *estadísticos-heurísticos*.

Con el objetivo de mostrar la portabilidad de nuestro método se utilizaron dos conjuntos de datos. La principal diferencia entre estos conjuntos es el idioma, el primero escrito en español y el segundo en inglés, y aunque ambos eran artículos de noticias, el primero contiene únicamente noticias correspondientes a desastres naturales mientras que el segundo contiene noticias de deportes, política, economía, etc.

Se realizó un primer conjunto de experimentos utilizando como forma de representación los atributos estadísticos-heurísticos. Al observar el comportamiento del clasificador en ambos conjuntos de datos, se confirmó la portabilidad del método, pues en ambos conjuntos el desempeño del sistema es muy similar. Aunque el sistema obtuvo altos niveles de precisión, fue posible observar que la tarea de identificar las oraciones relevantes seguía siendo difícil para el clasificador, debido a que los niveles de recuerdo obtenidos eran todavía muy deficientes.

Se hizo un análisis para determinar cuales podrían ser las verdaderas causas de este comportamiento. Tres posibles razones fueron identificadas: debido a la distribu-

ción de las clases, al etiquetado de los datos y a la representación de los datos. La distribución tan desbalanceada de las clases es algo natural dentro del área de generación de resúmenes, por lo cual el tratar de balancear las clases no era una opción. Por otro lado, el etiquetado de los datos para ambos conjuntos de datos fue realizado por un conjunto de expertos, el problema aquí es la subjetividad de la tarea, pues siempre existirán oraciones que puedan ser consideradas relevantes por un experto mientras que para otro no lo serán. La conclusión fue entonces que los datos estaban siendo mal representados, es decir, el conjunto de atributos que se utilizaron no eran suficientes para capturar la evidencia necesaria que serviría para identificar las oraciones relevantes.

La solución propuesta fue utilizar una forma de representación diferente. Nuestra propuesta es entonces utilizar las palabras como modo de representación de forma que sea posible aumentar la flexibilidad del sistema generador de resúmenes y al mismo tiempo ir disminuyendo la dependencia del dominio como del lenguaje. El uso de las palabras como forma de representación ha sido utilizado ampliamente en diferentes tareas de procesamiento de textos (e.g., clasificación de textos), sin embargo nunca utilizado en la tarea de generación de resúmenes.

En un primer bloque de experimentos se utilizó a las palabras simples (BOW) como forma de representación. Los resultados obtenidos demostraron cómo el utilizar las palabras como atributos, ayudan al clasificador a identificar las oraciones importantes. Además se observó que el comportamiento del sistema en ambos conjuntos de datos era muy similar, lo cual demuestra la pertinencia del método.

En un segundo bloque de experimentos se utilizó secuencias de palabras (n-gramas) como atributos. El uso de los n-gramas permite agregar al clasificador información del contexto (son atributos más detallados), la cual debido a que se trata de una tarea de clasificación "fina" representó un incremento en el desempeño del sistema. Cabe mencionar que con estos experimentos se demostró que el uso de n-gramas son adecuados para tareas donde la clasificación que se desea hacer es más fina (en particular a nosotros nos interesa identificar oraciones relevantes dentro de un documento completo), que no es el caso de la clasificación de textos temática, razón por la que los n-gramas no han funcionado en la resolución de esta tarea.

Evaluación tradicional fue aplicada al sistema al evaluar precisión, recuerdo y exactitud del clasificador. Pero además de esto, un tipo de evaluación extrínseca fue propuesto. El objetivo tras aplicar este tipo de evaluación a los resúmenes creados era

el determinar la calidad del resumen, es decir, queríamos medir que tanta información realmente importante estaba contenida en los resúmenes. Para lograr esto, se comparó el desempeño de un sistema de extracción de información al recibir como entrada en un primer modo los documentos completos y en un segundo modo los resúmenes generados por nuestro sistema. Por medio de esta evaluación comprobamos que la representación que utilizó atributos estadísticos-heurísticos que aunque si encuentra parte de la información relevante no lo hace tan bien como cuando se utiliza una representación por medio de secuencias de palabras (n-gramas).

Capítulo 5

Generando Automáticamente Resúmenes de Múltiples Documentos

En este capítulo se describe el trabajo realizado para resolver la tarea de generar resúmenes a partir de múltiples documentos. Como es de suponerse, el generar resúmenes de múltiples documentos implica necesariamente que se resuelvan los problemas a los que se enfrenta la tarea de generar resúmenes de un solo documento además de algunos otros como la eliminación de redundancias, resolver problemas temporales, etc.

En la primera sección de este capítulo se describe la arquitectura propuesta para la resolución de la tarea de generación de resúmenes de múltiples documentos. Esta arquitectura, a diferencia de la presentada en el capítulo anterior (Figura 4.1), combina dos grandes módulos. Un primer módulo que funciona bajo un esquema supervisado el cual es el encargado de identificar y extraer los elementos más importantes de cada documento dentro de la colección inicial, dando como salida un resumen por cada documento existente en la colección. Estos resúmenes sirven como colección de entrada al segundo módulo, el cual funciona bajo un esquema no supervisado. Este módulo se compone principalmente de un método de agrupamiento (*clustering*) que tiene como objetivo principal encontrar los "sub-temas" tratados en los resúmenes de entrada, además de la identificación y el manejo de la información redundante.

En la segunda sección se describe brevemente en que consiste ROUGE, que es el método de evaluación utilizado. ROUGE es una herramienta estándar de reciente creación que tiene por objetivo principal evaluar la calidad de los resúmenes gene-

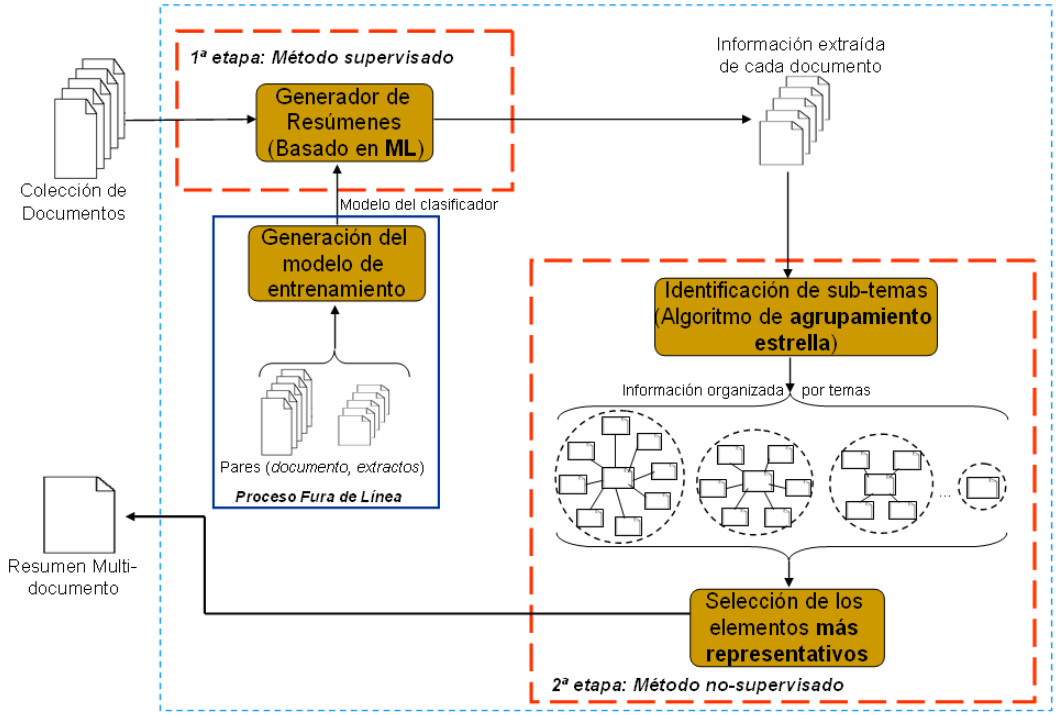

Figura 5.1: Arquitectura general del sistema multi-documento

rados automáticamente. Esta evaluación mide el "parecido" del resumen obtenido a través de un sistema de computo contra resúmenes "ideales" generados por humanos expertos en esta tarea.

La tercera sección muestra los resultados obtenidos con la arquitectura propuesta. Cabe mencionar que además de esta configuración , dos más son evaluadas con el objetivo de comparar y demostrar la pertinencia del método propuesto en este trabajo. Estas dos configuraciones agregadas simulan el comportamiento de técnicas utilizadas tradicionalmente dentro de esta área permitiéndonos así tener un punto de comparación. Finalmente son presentadas algunas conclusiones sobre los resultados obtenidos en este punto.

5.1. Arquitectura Propuesta

La figura 5.1 muestra el diagrama a bloques de la arquitectura propuesta para resolver la tarea de generar resúmenes de múltiples documentos.

Como es posible observar en la figura, la arquitectura esta dividida en dos grandes módulos o etapas. La primera, llamada etapa supervisada consiste principalmente de un clasificador. La idea básica del enfoque utilizado en esta etapa es que por medio de un proceso inductivo se construya de manera automática un clasificador al observar las características de un conjunto de documentos previamente resumidos, en particular estamos hablando de pares *(documentos, extractos)*. La salida de esta primera etapa consiste en un nuevo conjunto, esta vez de documentos cortos que corresponden al resumen de cada uno de los documentos de la colección inicial.

Los documentos cortos generados hasta este punto, servirán de entrada al segundo gran módulo, o etapa no supervisada. El objetivo principal de esta segunda etapa es identificar la información común como aquella no común (i.e., única) entre los documentos de la colección, en otras palabras, lo que se quiere encontrar son los diferentes sub-temas tratados dentro de la colección. En particular se utiliza un algoritmo de agrupamiento (*agrupamiento estrella*) para lograr estos objetivos. Dada la naturaleza de este algoritmo, la salida resulta en grupos ordenados de acuerdo a su tamaño y además dentro de cada grupo existe una cierta jerarquía entre sus elementos que nos permite identificar al elemento más representativo (i.e., el que contiene más información) resolviendo así los problemas de redundancia de información. Finalmente, un documento muy corto es generado a partir de estos grupos encontrados hasta alcanzar un tamaño especificado por el usuario.

5.1.1. Primera Etapa: El clasificador

Como hemos mencionado previamente, esta etapa se compone principalmente por un clasificador. Este clasificador es construido por medio de observar las características de un conjunto de documentos previamente resumidos i.e., pares *(documentos, resúmenes)*, convirtiéndose esto en una tarea de aprendizaje supervisado.

La configuración de este primer módulo es similar a la arquitectura propuesta en el capítulo 4. La característica principal de este clasificador es que utiliza la información del contexto para determinar cuando una oración es o no relevante. Si se observa de manera aislada, este primer módulo funcionaría por si solo como un sistema de generación de resúmenes de un documento. Sin embargo, una de las restricciones de la arquitectura mostrada el la figura 5.1 es que los documentos de entrada deberán estar relacionados temáticamente, e.g., si los documentos de entrada son noticias,

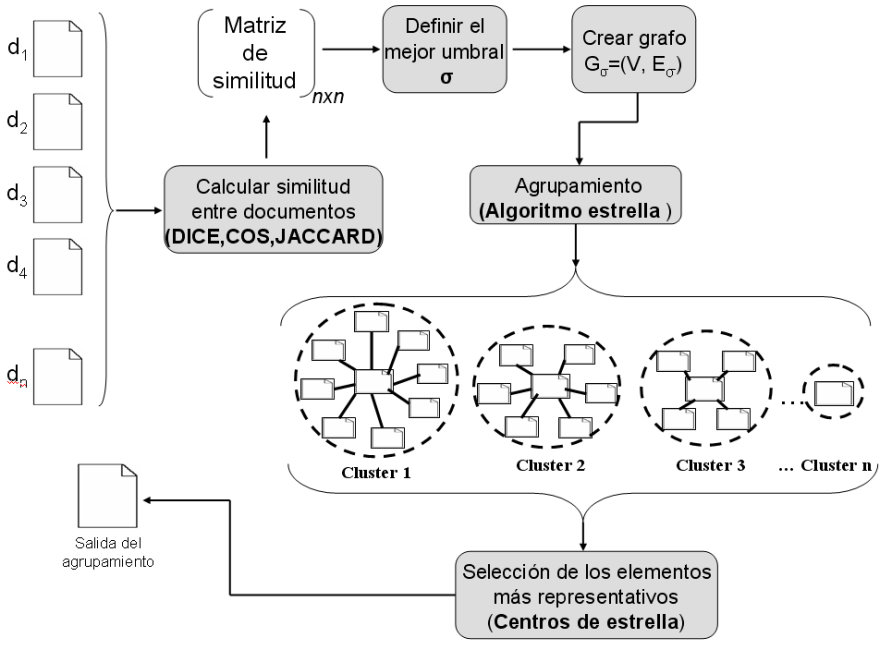

Figura 5.2: Descripción de la etapa no supervisada

éstas deberán hablar del mismo evento.

Para la tarea de generar resúmenes de múltiples documentos la configuración que se le dió a este primer módulo fue la siguiente: (i) la forma de representación de las instancias se hizo por medio de n-gramas, en particular se tomó hasta trigramas, (ii) el algoritmo de clasificación utilizado fue Naïve Bayes. En la sección 5.3 se muestran los resultados obtenidos de esta primer etapa. Esta evaluación consiste únicamente en medir la precisión y el recuerdo obtenido por el algoritmo de clasificación. Recordemos que esto es posible pues estamos trabajando con conjuntos de datos etiquetados.

5.1.2. Segunda Etapa: Algoritmo de Agrupamiento

Ya hemos descrito en la sección 2.3 el algoritmo de agrupamiento que utilizamos en esta parte de la arquitectura propuesta, el algoritmo estrella. Este algoritmo fue seleccionado debido a varias ventajas que tiene, una de las principales es el hecho de que es un algoritmo que deduce el número de grupos de manera natural, i.e., no es necesario especificar el número de grupos que se quieren formar.

La salida de este algoritmo son grupos de documentos en forma de "estrella"

(Figura 5.2). Otra de las ventajas de este algoritmo es que dada la forma de la salida (forma estrella), el algoritmo garantiza que el elemento central de cada una de las estrellas formadas es el elemento más representativo del grupo, pues el algoritmo induce de manera natural la estructura de los textos dentro del espacio de los textos, lo cual se logra asegurando que los demás elementos al menos son semejantes a el centro por un factor σ.

Este algoritmo, aunque con muchas ventajas, tiene algunas restricciones, el algoritmo descrito en la tabla 2.3 en el primer paso menciona el cálculo de un grafo umbralizado G_σ (Ver Figura 5.2). Para calcular este grafo es necesario aplicar técnicas de medición de similaridad entre documentos por medio de las cuales es posible definir el umbral σ. Para este trabajo se utilizaron tres técnicas estándares dentro de la comunidad de recuperación de información, la siguiente sección describe estas técnicas.

Calculando similaridad entre documentos

El objetivo de las medidas de similaridad dentro de la comunidad de recuperación de información es comparar que tanto un vector representando (Modelo vectorial ver sección 2.2.1) un documento petición se parece a otro dentro de una colección. Existen varias métricas de similaridad entre documentos bien documentadas pero nosotros sólo mencionaremos tres de las más utilizadas.

El objetivo es contar con un valor numérico al cual llamaremos coeficiente de similaridad SC, el cual nos dirá cuan parecidos son los documentos D_i y D_j. Es conveniente mencionar que para nuestro sistema ambos documentos son manejados como *documentos cortos*, en particular nuestro sistema coloca en cada documento sólo una oración. Entonces, dadas estas restricciones tenemos:

- **Medida Cosenoidal.** La idea básica de ésta es medir el ángulo entre el vector de D_i y de D_j, para hacerlo, calculamos:

$$SC(D_i, D_j) = \frac{\sum_{k=1}^{t} w_{ik} w_{jk}}{\sqrt{\sum_{k=1}^{t} (w_{jk})^2 \sum_{k=1}^{t} (w_{ik})^2}} \tag{5.1.1}$$

- **Medida DICE.** El coeficiente de DICE es obtenido por medio de:

$$SC(D_i, D_j) = \frac{2\sum_{k=1}^{t} w_{ik}w_{jk}}{\sum_{k=1}^{t}(w_{jk})^2 + \sum_{k=1}^{t}(w_{ik})^2} \tag{5.1.2}$$

- **Medida de Jaccard.** El coeficiente de Jaccard es calculado por medio de:

$$SC(D_i, D_j) = \frac{\sum_{k=1}^{t} w_{ik}w_{jk}}{\sum_{k=1}^{t}(w_{jk})^2 + \sum_{k=1}^{t}(w_{ik})^2 - \sum_{k=1}^{t} w_{ik}w_{jk}} \tag{5.1.3}$$

En todos los casos k va de 1 a el número total de términos del vocabulario t, w_{ik} indica la frecuencia del término k en el documento D_i y w_{jk} la frecuencia del término k en el documento D_j.

Para los fines de nuestra investigación, sólo estas tres diferentes formas de medir la similitud entre documentos fueron consideradas, para ver más detalles sobre estas refiérase a [21].

Construcción del agrupamiento

Como hemos mencionado antes, el primer paso antes de pasar a la identificación y formación de los grupos es calcular un grafo umbralizado. Para lograr esto, es necesario calcular una matriz de similitudes entre los diferentes documentos de la colección. Esta matriz se forma al utilizar alguna de las medidas mencionadas en la sección anterior.

Una vez que la matriz está completa, se puede proceder a formar el grafo, lo cual se logra por medio de definir el umbral σ. Una de las restricciones del algoritmo de agrupamiento estrella es este, pues dependiendo del umbral σ el resultado será diferente. Por ejemplo, si σ tiene un valor alto ($\sigma \to \infty$) esto para el algoritmo quiere decir que la formación de los grupos se deberá ser bastante estricta, i.e. el número de grupos resultantes serán más de los que se hubieran logrado obtener si σ hubiese sido más pequeño ($\sigma \to 0$), pero se garantiza que los grupos formados estarán fuertemente relacionados.

Esto en términos de los documentos quiere decir que, si seleccionamos un valor alto para σ muy probablemente obtendremos un gran número de grupos, pero los documentos contenidos en cada grupo tendrán una similitud bastante marcada. Por el contrario, si σ es pequeño, el número de grupos que se formarían sería mucho

menor, y además la similitud entre los elementos de cada grupo no será tan marcada que como lo serían con σ grande.

El problema es entonces, antes de obtener por completo el grafo umbralizado y poder aplicar el algoritmo descrito en la tabla 2.3, como definir el mejor umbral σ.

Para hacer esta selección, se hizo uso de la información estadística de la matriz de similitud. Lo que se hizo fue calcular la media ($\overline{x}$) y la desviación estándar (δ) de los datos. Como es conocido, la media es una medida que nos describe cuál es la tendencia de los datos, y la desviación estándar es una medida que nos dice que tanto se acercan o se alejan los datos entre si. Para nuestros experimentos utilizamos tres diferentes formas de seleccionar σ que son: En un primer modo sólo tomamos el valor de la media $\overline{x}$ para definir el umbral. En un segundo modo tomamos la media más la desviación estándar ($\overline{x} + \delta$) como valor del umbral, siendo este un umbral alto o fuerte. Por ultimo, a forma de un umbral bajo o laxo se seleccionó como valor de σ el valor de la media menos la desviación estándar ($\overline{x} - \delta$).

En nuestra etapa de experimentación se consideraron las tres diferentes medidas de similitud para formar la matriz de similitudes y además se consideraron también las tres formas para definir el valor de σ.

Construcción del resumen

Otro paso importante dentro del algoritmo de agrupamiento es la selección de los elementos más representativos. Como se ha mencionado antes, el algoritmo estrella garantiza que el elemento central de cada estrella será el elemento más representativo de cada grupo. Dado esto, la solución más sencilla consistiría en tomar el centro de cada unas de las estrellas y a partir de estas construir el resumen final. Sin embargo hay un problema, el resumen final debe cumplir con una restricción en tamaño.[1] Con esta restricción el problema de seleccionar los elementos más representativos ya no es tarea sencilla, pues la forma de selección dependerá del número de estrellas y además del tamaño especificado para el resumen final.

Como se pudo observar en la figura 5.2 el tamaño de las estrellas van de mayor a menor, el cual es un patrón que siempre se repetirá sin importar la medida de similitud utilizada para el cálculo de la matriz de similitud o el valor del umbral σ utilizado para la creación del grafo umbralizado. Esto sucede debido a que siempre

[1]Generalmente el usuario proporcionada el número de palabras u oraciones que desea en el resumen final. Para nuestros experimentos este tamaño se fijó a 200 palabras.

habrá palabras o secuencias de palabras que se repetirán con mayor frecuencia que otras dentro de la colección de documentos (Ley de Zipf, ver sección 3.2.1). A partir de aquí se define el primer criterio para la selección de los elementos más representativos: tomar los elementos centrales de las estrellas empezando con las de mayor tamaño hasta llegar a las más pequeñas.

Al seguir este criterio de selección nos enfrentamos a dos escenarios diferentes:

- *El número de estrellas es insuficiente para completar el tamaño del resumen requerido únicamente con elementos centrales.* En este caso el criterio de selección fue el siguiente: Una vez que se han tomado todos los elementos centrales de cada estrella, volvemos a empezar con la estrella de mayor tamaño y seleccionamos en esta ocasión un elemento "satélite". Esto se hace con las estrellas siguientes hasta completar el tamaño del resumen requerido por el usuario.

- *El número de estrellas es mucho mayor al número de elementos centrales necesarios para completar el resumen.* Este es el caso más sencillo, pues aquí el criterio de paro será establecido por el tamaño del resumen requerido. Es decir, se irán tomando elementos centrales de cada estrella, empezando por las de mayor tamaño, hasta completar el tamaño del resumen requerido por el usuario.

5.2. Evaluación ROUGE

Tradicionalmente la evaluación de los resúmenes generados de manera automática involucra el juicio de algún conjunto de expertos basándose en algunas métricas, por ejemplo la coherencia y consistencia gramatical, legibilidad y contenido. Sin embargo una evaluación manual tan simple como esta, tomaría el esfuerzo considerable de muchas personas lo cual resultaría en un método de evaluación costoso y al mismo tiempo difícil de mantener bajo control evitando la subjetividad.

ROUGE es un sistema automático para la evaluación de resúmenes propuesto por Lin y Hovy [33, 31]. Este sistema está basado en el método propuesto para la evaluación de traducciones automáticas BLEU [43], i.e. en la co-ocurrencia de los n-gramas. Lin y Hovy demuestran en [33] cómo este tipo de métricas pueden ser aplicados para evaluar la calidad de los resúmenes generados automáticamente.

ROUGE ha demostrado ser una herramienta capaz de medir la correlación entre resúmenes generados por humanos y los resúmenes generados automáticamente,

razón por la cual ha sido utilizado en los dos últimos años del DUC[2] (2004 y 2005). ROUGE incluye diferentes métricas para evaluar ésta correlación. En esta sección sólo describimos dos de estas métricas, las cuales fueron utilizadas en nuestra investigación y además son las que normalmente se utilizan en el foro DUC.

- **ROUGE-N: Co-ocurrencia de N-gramas.** ROUGE es un método de evaluación basado en el recuerdo entre un resumen "candidato" (resumen generado automáticamente) y un resumen "referencia" (generado por un experto humano). ROUGE-N es calculado por medio de:

$$ROUGE-N = \frac{\sum_{S_i \in \{ResumenReferencia\}} \sum_{gram_n \in S_i} Count_{match}(gram_n)}{\sum_{S_i \in \{ResumenReferencia\}} \sum_{gram_n \in S_i} Count(gram_n)} \quad (5.2.1)$$

  donde S_i se refiere a la oración i dentro del resumen de referencia, n es la longitud del n-grama, $gram_n$ y $Count_{match}(gram_n)$ es el máximo número de n-gramas que co-ocurren en el resumen candidato y el conjunto de resúmenes de referencia.

  Es claro que ROUGE-N es una medida basada en recuerdo, pues el denominador de la ecuación es la suma total de n-gramas que ocurren en el o los resúmenes de referencia.

  Nótese que el número de n-gramas en el denominador de ROUGE-N se incrementa al mismo tiempo que incrementamos el número de resúmenes de referencia. Esto es algo intuitivo y razonable debido a que puede existir más de un "buen" resumen, de esta forma nosotros preferimos un resumen candidato que tiene mayor similitud entre varios resúmenes de referencia.

- **ROUGE-L: Sub-secuencia Común más Larga.** Una secuencia $Z = [z_1, z_2, \ldots, z_n]$ es una subsecuencia de otra secuencia $X = [x_1, x_2, \ldots, x_m]$, si existe una secuencia incremental de índices $[i_1, i_2, \ldots, i_k]$ de X de tal forma que para todo $j = 1, 2, \ldots, k$, se tiene $x_{ij} = z_j$. Dadas dos secuencias X y Y, la

[2]DUC (*Document Understanding Conference*) es un foro especializado en la evaluación de los sistemas de generación de resúmenes. Uno de los objetivos principales de este foro es lograr la estandarización sobre el modo de evaluación de este tipo de sistemas. Desde sus inicios en el año 2001 los encargados de este foro se han ocupado de la recolección y evaluación de diferentes sistemas de generación de resúmenes, lo cual ha permitido la creación de múltiples corpus con los cuales los participantes de este foro pueden entrenar y evaluar sus sistemas (http://duc.nist.gov/).

subsecuencia común más larga (LCS) de X y Y es la subsecuencia común con longitud máxima.

Para aplicar esta medida, pensemos en una oración de un resumen como una secuencia de palabras. La intuición nos dice que entre mayor el valor de LCS entre dos oraciones, entonces existe mayor similitud entre los resúmenes. A diferencia de ROUGE-N, ROUGE-L es una medida basada en la medida F, la cual ayuda a estimar la similitud entro dos resúmenes, X de longitud m y Y de longitud n, supongamos que X es el resumen de referencia y que Y es el resumen candidato, entonces ROUGE-L se calcula:

$$R_{lcs} = \frac{LCS(X,Y)}{m} \tag{5.2.2}$$

$$C_{lcs} = \frac{LCS(X,Y)}{n} \tag{5.2.3}$$

$$F_{lcs} = \frac{(1+\beta^2)R_{lcs}C_{lcs}}{R_{lcs} + \beta^2 C_{lcs}} \tag{5.2.4}$$

donde $LCS(X,Y)$ es el valor de la subsecuencia más larga entre X y Y, $\beta = C_{lcs}/R_{lcs}$. De esta forma ROUGE-L equivale a el valor obtenido de calcular la ecuación 5.2.4.

5.3. Experimentos y Resultados

Dado que la arquitectura propuesta consta de dos etapas (*supervisada y No-supervisada*) nuestros experimentos son divididos en dos etapas. Primero evaluamos la etapa supervisada por medio de la estrategia de validación cruzada (*cross fold validation*) y además también se muestran los resultados del clasificador al tratar con un conjunto de datos no vistos. Posteriormente, dado que la salida de la segunda etapa es el resumen multi-documento, ésta es evaluada por medio de la herramienta ROUGE.

5.3.1. Conjunto de Datos

Para la realización de los experimentos se trabajo con el conjunto de datos del DUC 2003, en particular con el conjunto de datos correspondiente a la tarea 4 del DUC. La razón por la que se seleccionó este conjunto de datos, es debido a que es

un conjunto de datos etiquetado (cada oración tiene la etiqueta "Relevante" ó "No-relevante"), y dado que nuestro sistema hace uso de una etapa supervisada, este corpus es fácilmente adaptable a nuestro esquema, además que permite evaluar el clasificador por medio de precisión y recuerdo.

La tabla 5.1 muestra las características de este conjunto de datos.

Tabla 5.1: Estadísticas del conjunto de datos DUC-task-4

	Idioma	**Dominio**	**Número de Colecciones**	**Número oraciones**	**Oraciones Relevantes**
DUC	Inglés	General	30 (22 doctos × colecc)	33666	1995 (5.92 %)

Como es posible notar en la tabla 5.1 la distribución de las clases es bastante desbalanceada pues apenas el 5,92 % de las instancias son instancias *positivas*. Para hacer los experimentos se seleccionó de manera aleatoria el 80 % de las instancias como conjunto de entrenamiento, siendo el 20 % restante el conjunto de prueba. La distribución de los datos tras hacer esta separación se muestra en la tabla 5.2

Tabla 5.2: Estadísticas del conjunto de datos DUC-task-4

	Número de Colecciones	**Número oraciones**	**Oraciones Relevantes**
DUC-2003-80 %	22	27975	1548(5.53 %)
DUC-2003-20 %	8	5691	447 (7.85 %)

5.3.2. Evaluación de la etapa supervisada

En el capítulo 4 se concluye que la representación por medio de secuencias de palabras (i.e., n-gramas) es la más adecuada para el modelo de aprendizaje utilizado para la obtención de los extractos más importantes dentro de un documento. Así pues, para el sistema de generación de resúmenes de múltiples documentos se optó por utilizar la misma forma de representación, y en esta sección se muestra el desempeño obtenido por el clasificador.

Debido a que el conjunto de datos es mucho mayor en comparación con los utilizados en el capítulo 4, consecuentemente el número de $1, 2$ y 3 gramas aumenta

de manera considerable. La tabla 5.3 muestra con mayor detalles estos datos. Dado este incremento tan considerable, un algoritmo de reducción de dimensionalidad fue aplicado, al igual que en nuestros experimentos anteriores, Ganancia de Información fue el método que se utilizó para hacer la selección de atributos.

Tabla 5.3: Número de atributos (n-gramas) en el corpus DUC-2003

	Originales				**Seleccionados**
	1-gramas	*2-gramas*	*3-gramas*	*Total*	*Total*
DUC-2003-80 %	25937	205052	339696	570685	1291

En la tabla 5.4 se observan los resultados obtenidos por el clasificador al entrenar y evaluar sobre el 80 % del conjunto de datos DUC-2003 con la representación antes mencionada. En la segunda línea de la tabla se observan los resultados obtenidos por el clasificador al entrenar con el 80 % de los datos, pero esta vez se evaluó sobre el 20 % restante de los datos.

Tabla 5.4: Evaluación del sistema con la representación n-gramas

	Palabras simples			**n-gramas**		
	p	r	e	p	r	e
DUC-2003-80 %	93.04	97.35	91.03	93.57	97.77	91.91
DUC-2003-20 %	95.94	92.28	89.60	96.35	92.65	89.62

Es posible observar en los resultados de este experimento que la tendencia se sigue manteniendo en comparación con los resultados obtenidos en el capítulo 4, pues el representar los documentos por medio de n-gramas sigue siendo mejor que el hacerlo con palabras simples. En este caso, la mejora es menos notoria que en los experimentos realizados con los conjunto de datos de Desastres y CAST, pero recordemos también que el conjunto DUC-2003 es mucho más grande que cualquiera de estos dos, lo cual provoca que esta mejora en el desempeño del clasificador sea aparentemente pequeña, pero en realidad las pequeñas variaciones decimales implican la identificación correcta de un número considerable de instancias relevantes. Por ejemplo, al hacer la representación del 80 % de los documentos con palabras simples se logran encontrar 880 instancias positivas, mientras con la representación por medio de n-gramas se logran encontrar 984 instancias positivas, es decir casi 100 instancias más que no se ven correctamente reflejadas en los porcentajes de la tabla 5.4.

5.3.3. Evaluación de la etapa NO-supervisada

Hemos mencionado en secciones anteriores los diferentes parámetros que hasta este momento tiene nuestro sistema. Una vez que se tiene la salida de la etapa supervisada, el primer parámetro que se debe definir es la medida de similitud que se utilizará para el cálculo de la matriz de similitudes (Ver figura 5.2). Posteriormente se debe definir el umbral σ que será utilizado para poder ejecutar el algoritmo de la tabla 2.3.

Método Base

En la tarea de generar resúmenes de múltiples documentos existen dos diferentes métodos base (*baselines*) definidos por el foro DUC, los cuales son:

1. **Método Base 1.** Éste consiente en seleccionar las n primeras líneas ó bytes (hasta completar un determinado tamaño) del documento más reciente dentro de la colección de documentos.

2. **Método Base 2.** Éste consiste en tomar las primeras n líneas ó bytes de cada uno de los documentos dentro de la colección. Generalmente n es un número pequeño (e.g. $n = 1$).

Para propósitos de evaluación hemos calculado ambos baselines. Con el objetivo de tener resúmenes de tamaños comparables hemos fijado el tamaño de éstos a 200 palabras para ambos baselines y además para los generados con nuestro sistema.

Las siguientes tablas muestran los resultados obtenidos de nuestro sistema al evaluar el resumen generado automáticamente contra un conjunto de resúmenes ideales generados por humanos.

Las tablas muestran el puntaje obtenido por nuestros resúmenes al ser evaluados con la herramienta ROUGE. En particular ROUGE-N va desde $N = 1$ hasta 4. Si observamos cuidadosamente las tablas podremos notar que el utilizar un umbral alto o fuerte (i.e., $\overline{x}+\delta$) permite al sistema formar resúmenes de mejor calidad sin importar la medida de similitud empleada. Si a este umbral fuerte lo combinamos con la medida de similitud DICE, notamos que los resúmenes obtenidos son aún de mejor calidad. En las cuatro tablas esta configuración es la que siempre obtiene el mejor puntaje.

El puntaje que obtenemos con ROUGE-N ($N = 1 \ldots 4$) es una medida que nos dice, que tanto de los n-gramas utilizados por los expertos para la creación de su propio resumen, estamos obteniendo con la configuración propuesta. Nótese que la

Tabla 5.5: Evaluación ROUGE contra un resumen de referencia

Configuración	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4	ROUGE-L
SUM-DICE ($\overline{x}$)	0.38792	0.06476	0.01991	0.00879	0.34488
SUM-DICE ($\overline{x} - \delta$)	0.38284	0.05863	0.01506	0.00626	0.33721
SUM-DICE ($\overline{x} + \delta$)	**0.41475**	**0.08697**	**0.03248**	**0.01704**	**0.37291**
SUM-COS ($\overline{x}$)	0.39057	0.06445	0.01966	0.00879	0.34492
SUM-COS ($\overline{x} - \delta$)	0.38004	0.05777	0.01477	0.00626	0.33579
SUM-COS ($\overline{x} + \delta$)	0.40482	0.08131	0.02922	0.01585	0.36435
SUM-JACC ($\overline{x}$)	0.3911	0.06217	0.01674	0.00614	0.34525
SUM-JACC ($\overline{x} - \delta$)	0.38004	0.05777	0.01477	0.00626	0.33579
SUM-JACC ($\overline{x} + \delta$)	0.41413	0.08531	0.03129	0.01646	0.37384
Baseline 1	0.25111	0.04065	0.01594	0.0078	0.22983
Baseline 2	0.25322	0.0265	0.00776	0.00434	0.2372

Tabla 5.6: Evaluación ROUGE contra dos resúmenes de referencia

Configuración	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4	ROUGE-L
SUM-DICE ($\overline{x}$)	0.37399	0.06135	0.01798	0.00804	0.33129
SUM-DICE ($\overline{x} - \delta$)	0.37077	0.06583	0.02132	0.00865	0.32903
SUM-DICE ($\overline{x} + \delta$)	**0.3949**	**0.08969**	**0.03393**	**0.01926**	**0.35092**
SUM-COS ($\overline{x}$)	0.3761	0.06036	0.01695	0.00804	0.32983
SUM-COS ($\overline{x} - \delta$)	0.36671	0.06481	0.02081	0.00865	0.32649
SUM-COS ($\overline{x} + \delta$)	0.38219	0.08457	0.03187	0.01926	0.3443
SUM-JACC ($\overline{x}$)	0.37707	0.05669	0.0127	0.00429	0.32929
SUM-JACC ($\overline{x} - \delta$)	0.36671	0.06481	0.02081	0.00865	0.32649
SUM-JACC ($\overline{x} + \delta$)	0.39282	0.08852	0.03283	0.01872	0.3498
Baseline 1	0.24587	0.04328	0.01904	0.01009	0.22621
Baseline 2	0.25173	0.02492	0.00782	0.00473	0.23874

diferencia entre la mejor configuración en la tabla 5.5 y la mejor configuración en la tabla 5.8 es de apenas un punto porcentual aproximadamente, lo cual quiere decir que estamos creando un resumen que coincide en el 40 % de su contenido (para el caso ROUGE-1) con los resúmenes (4 resúmenes de referencia) ideales creados por humanos.

Por otro lado, ROUGE-L es una medida que nos dice que se están utilizando las palabras correctas (n-gramas) en el orden correcto. Es de esperarse que este valor siempre sea menor al puntaje obtenido por ROUGE-1 pues recordemos que un hu-

Tabla 5.7: Evaluación ROUGE contra tres resúmenes de referencia

Configuración	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4	ROUGE-L
SUM-DICE ($\overline{x}$)	0.37586	0.06008	0.01741	0.00788	0.33292
SUM-DICE ($\overline{x} - \delta$)	0.37234	0.05848	0.01604	0.00613	0.32763
SUM-DICE ($\overline{x} + \delta$)	**0.40045**	**0.08305**	**0.03118**	**0.01711**	**0.35589**
SUM-COS ($\overline{x}$)	0.37828	0.05865	0.01669	0.00788	0.3315
SUM-COS ($\overline{x} - \delta$)	0.3696	0.05779	0.01569	0.00613	0.32591
SUM-COS ($\overline{x} + \delta$)	0.38395	0.07669	0.02833	0.01639	0.3429
SUM-JACC ($\overline{x}$)	0.37934	0.05549	0.01315	0.00466	0.33185
SUM-JACC ($\overline{x} - \delta$)	0.3696	0.05779	0.01569	0.00613	0.32591
SUM-JACC ($\overline{x} + \delta$)	0.39834	0.08232	0.03045	0.01676	0.3541
Baseline 1	0.248	0.04071	0.01614	0.00806	0.22763
Baseline 2	0.25251	0.02457	0.00832	0.0045	0.23727

Tabla 5.8: Evaluación ROUGE contra cuatro resúmenes de referencia

Configuración	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4	ROUGE-L
SUM-DICE ($\overline{x}$)	0.38777	0.0663	0.02174	0.00921	0.34519
SUM-DICE ($\overline{x} - \delta$)	0.37797	0.05564	0.01446	0.00568	0.33411
SUM-DICE ($\overline{x} + \delta$)	**0.40981**	**0.08584**	**0.03256**	**0.01666**	**0.36782**
SUM-COS ($\overline{x}$)	0.39038	0.06521	0.0212	0.00921	0.34464
SUM-COS ($\overline{x} - \delta$)	0.37539	0.05486	0.0142	0.00568	0.33281
SUM-COS ($\overline{x} + \delta$)	0.39223	0.07562	0.02659	0.01447	0.3521
SUM-JACC ($\overline{x}$)	0.39118	0.06309	0.01853	0.00678	0.3449
SUM-JACC ($\overline{x} - \delta$)	0.37539	0.05486	0.0142	0.00568	0.33281
SUM-JACC ($\overline{x} + \delta$)	0.40897	0.08446	0.03148	0.01612	0.36829
Baseline 1	0.25237	0.03879	0.01507	0.0074	0.23095
Baseline 2	0.26084	0.02808	0.00868	0.00418	0.24514

mano siempre hará uso de su conocimiento del "mundo" para la generación de su resumen.

Experimentos Adicionales

Dos experimentos adicionales fueron realizados con el objetivo de probar la funcionalidad de la arquitectura propuesta (Figura 5.1). Como se mencionó en el capítulo 3 los esquemas tradicionales para la generación de resúmenes de múltiples documentos

generalmente utilizan técnicas que funcionan igual en cualquier tipo de dominio, idioma, tipo de documento, etc., lo cual provoca que la particularidad que pueda tener cada documento se pierda generando resúmenes que contienen sólo lo "común" de la colección. Nosotros evitamos este problema al utilizar un esquema supervisado para la extracción de la información importante de cada documento de la colección. Esto sucede debido a que por medio de un modelo de aprendizaje podemos adaptarnos a diferentes dominios y/o lenguajes sin mucho esfuerzo, y de esta forma considerar la particularidad de cada documento. Al hacer esto, los extractos obtenidos contienen tanto información común como información particular de cada documento dentro de la colección, la cual es posteriormente organizada por el algoritmo de agrupamiento.

Con el objeto de probar estas afirmaciones se hicieron pruebas con dos configuraciones diferentes a la propuesta original mostrada en la figura 5.1.

Se propone en un primer experimento adicional generar resúmenes utilizando únicamente la etapa no-supervisada. A este experimento le denominaremos "segunda configuración", donde los documentos de la colección entran a esta etapa sin ser procesados de ninguna forma. La etapa de agrupamiento tratara de encontrar los diferentes sub-temas tratados dentro de la colección. Al final de la etapa contamos con una larga lista de grupos estrella, donde cada grupo contiene un gran número de satélites. Posteriormente se seleccionan los elementos más representativos de los grupos más grandes hasta que se construye el resumen del tamaño deseado (200 palabras). En la figura 5.3 puede observarse cómo funciona esta configuración propuesta.

La tabla 5.9 muestra los resultados obtenidos tras evaluar con ROUGE los resúmenes obtenidos utilizando la configuración de la figura 5.3. Esta tabla muestra los resultados al evaluar contra cuatro resúmenes de referencia.

Es posible notar que en esta configuración se generan resúmenes de mejor calidad cuando el parámetro σ es igual a el valor de la media (i.e., $\overline{x}$). En particular la mejor combinación de parámetros en esta segunda configuración es utilizar la medida cosenoidal junto con $\sigma = \overline{x}$. Aún cuando con esta combinación de parámetros se logra obtener un puntaje de 35.98 % para ROUGE-1, éste es 3.05 puntos porcentuales menor a su equivalente en la tabla 5.8 y comparado con el mejor caso obtenido en el mismo experimento (SUM-DICE ($\overline{x}+\delta$) en la Tabla 5.8) éste es 5 puntos porcentuales menor.

Los resultados obtenidos en este experimento nos ayudan a confirmar que el uso de una etapa supervisada donde los documentos sean tratados de manera individual ayuda a identificar y extraer información tanto común como muy particular de cada

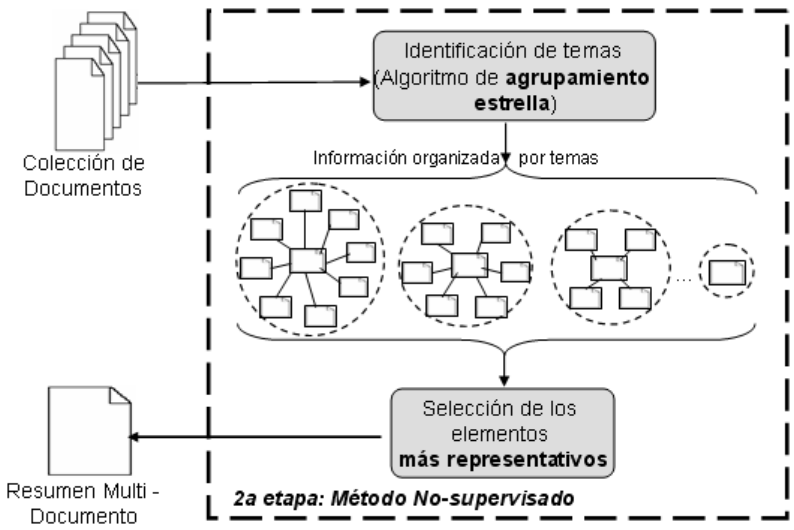

Figura 5.3: Segunda configuración: Solo Agrupamiento

Tabla 5.9: Segunda configuración: Evaluación ROUGE contra cuatro resúmenes de referencia

Configuración	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4	ROUGE-L
SUM-DICE ($\overline{x}$)	0.33264	0.04417	0.01365	0.00431	0.29381
SUM-DICE ($\overline{x} - \delta$)	0.33646	0.04373	0.00862	0.0038	0.2963
SUM-DICE ($\overline{x} + \delta$)	0.26093	0.0433	0.01506	0.00688	0.23304
SUM-COS ($\overline{x}$)	**0.35986**	**0.04979**	**0.01392**	**0.00431**	**0.31933**
SUM-COS ($\overline{x} - \delta$)	0.33646	0.04373	0.00862	0.0038	0.2963
SUM-COS ($\overline{x} + \delta$)	0.24189	0.04181	0.01403	0.00663	0.2169
SUM-JACC ($\overline{x}$)	0.3179	0.0408	0.00911	0.00215	0.2832
SUM-JACC ($\overline{x} - \delta$)	0.33646	0.04373	0.00862	0.0038	0.2963
SUM-JACC ($\overline{x} + \delta$)	0.21825	0.0411	0.01846	0.01059	0.19505
Propuesta	**0.40981**	**0.08584**	**0.03256**	**0.01666**	**0.36782**
Baseline 1	0.25237	0.03879	0.01507	0.0074	0.23095
Baseline 2	0.26084	0.02808	0.00868	0.00418	0.24514

documento, lo que resulta en la generación de un resumen más completo y además más similar al creado por humanos.

El siguiente experimento tiene como finalidad demostrar que, analizar en un primer paso por medio de un modelo de aprendizaje cada documento de la colección ayuda

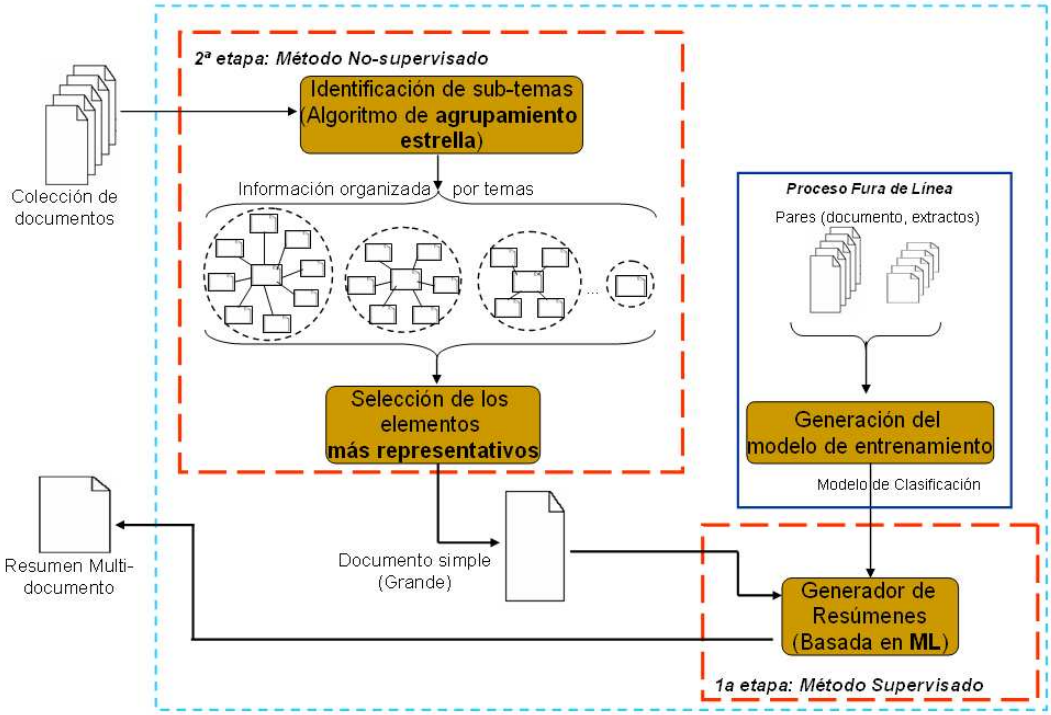

Figura 5.4: Tercera configuración: Etapas invertidas

a aumentar la calidad del resumen generado. Para demostrar esto, la configuración propuesta para este experimento consiste en invertir las etapas, i.e., los documentos de la colección son procesados primero por la etapa de agrupamiento, a partir de los sub-temas encontrados en la colección se genera un documento (grande) sin restricciones de tamaño. Este documento sirve de entrada a la etapa supervisada, convirtiéndose así esto en el problema de generar un resumen de un sólo documento. La figura 5.4 muestra el diagrama a bloques de esta tercera configuración.

La tabla 5.10 muestra los resultados obtenidos tras evaluar los resúmenes obtenidos con la configuración propuesta en la figura 5.4 contra cuatro resúmenes de referencia.

Los resultados obtenidos con esta configuración muestran que resúmenes de mejor calidad son generados cuando se utiliza un umbral alto o fuerte ($\overline{x} + \delta$). Los resultados de la tabla 5.10 muestran la misma tendencia que los que se obtuvieron en la configuración original (Tabla 5.8).

En este experimento la combinación que obtiene el mejor puntaje es DICE junto con un umbral alto. El valor obtenido es de 38.87 % para ROUGE-1 que es 2.11 puntos

Tabla 5.10: Tercera configuración: Evaluación ROUGE contra cuatro resúmenes de referencia

Configuración	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4	ROUGE-L
SUM-DICE ($\overline{x}$)	0.36758	0.05413	0.01901	0.00811	0.32095
SUM-DICE ($\overline{x} - \delta$)	0.33317	0.04076	0.0043	0.00133	0.29699
SUM-DICE ($\overline{x} + \delta$)	**0.38871**	**0.05838**	**0.01549**	**0.00698**	**0.34228**
SUM-COS ($\overline{x}$)	0.35223	0.04113	0.01026	0.00445	0.30649
SUM-COS ($\overline{x} - \delta$)	0.33317	0.04076	0.0043	0.00133	0.29699
SUM-COS ($\overline{x} + \delta$)	0.37265	0.05725	0.01643	0.00542	0.33389
SUM-JACC ($\overline{x}$)	0.35961	0.04512	0.00914	0.00299	0.31647
SUM-JACC ($\overline{x} - \delta$)	0.33317	0.04076	0.0043	0.00433	0.29699
SUM-JACC ($\overline{x} + \delta$)	0.38101	0.05606	0.01468	0.00616	0.33726
Propuesta	**0.40981**	**0.08584**	**0.03256**	**0.01666**	**0.36782**
Tradicional	**0.35986**	**0.04979**	**0.01392**	**0.00431**	**0.31933**
Baseline 1	0.25237	0.03879	0.01507	0.0074	0.23095
Baseline 2	0.26084	0.02808	0.00868	0.00418	0.24514

porcentuales menor a su equivalente en la tabla 5.8. Aunque no es muy grande la diferencia entre estos valores, estos resultados nos ayudan a concluir que el uso de la etapa supervisada ayuda a la generación de mejores resúmenes y además podemos concluir que el tratar en una primera etapa de manera individual a los documentos de la colección, en lugar de tratarlos como un todo, ayuda indudablemente a la generación de un resumen de mayor calidad, esto debido a que al aplicar en un primer paso el algoritmo de agrupamiento provoca que se pierda información importante.

5.4. Discusión

En este capítulo se describió la arquitectura propuesta para el sistema generador de resúmenes de múltiples documentos. La principal característica de ésta es su división en dos grandes módulos, el primero un módulo basado en un esquema de aprendizaje supervisado. La configuración de este módulo se hizo basándonos en los resultados obtenidos en los experimentos realizados en el capítulo 4, donde se concluye que el uso de secuencias de palabras (n-gramas) como modo de representación de las instancias ayuda al clasificador a identificar con mayor eficiencia las piezas de información más importantes dentro de cada documento. El segundo módulo se compone principalmente de un método de agrupamiento (no-supervisado). El uso de

esta etapa permite la identificación de la información común y también de aquella información única dentro de la colección de documentos. Otra ventaja adicional del uso de esta etapa no-supervisada es que permite eliminar las redundancias y además tener un control adecuado sobre el tamaño del resumen a generar.

La principal diferencia del sistema propuesto en este trabajo contra las técnicas utilizadas tradicionalmente, es la adición de un módulo basado totalmente en técnicas de aprendizaje automático que funciona en conjunto técnicas no supervisadas. La principal ventaja que aporta esta configuración es que permite al sistema adaptarse fácilmente a diferentes dominios temáticos, lenguajes y a las diferentes necesidades de los usuarios.

Como se explicó en secciones anteriores, existen varios parámetros que son necesarios para el sistema, la forma en que se calculará la similitud entre las oraciones, la forma en que se elegirá el umbral σ y el tamaño del resumen deseado. Por esta razón se realizaron diferentes experimentos con el objetivo de tratar de identificar la mejor secuencia de parámetros para el sistema de generación de resúmenes de múltiples documentos.

Sin embargo con el objetivo de contar con un punto de comparación más confiable y así poder evaluar la verdadera aportación del sistema propuesto se definieron dos arquitecturas más. La primera de estas a la que denominamos 2a arquitectura, simula el comportamiento del esquema tradicionalmente usado dentro del área de generación de resúmenes de múltiples documentos. Ésta hace uso únicamente de la etapa no supervisada (i.e., el agrupamiento), la principal característica de esta configuración es la facilidad que tiene de encontrar los sub-temas comunes dentro de la colección de documentos (Ver figura 5.3) y al mismo tiempo eliminar las redundancias.

Al comparar los resultados obtenidos con esta segunda arquitectura contra los obtenidos con la arquitectura propuesta se observa que el usar únicamente el módulo no supervisado resulta en resúmenes de baja calidad. En otras palabras, se demostró que el uso del módulo supervisado permite al sistema crear resúmenes de mejor calidad. Esto sucede debido a que el módulo supervisado (i.e., el clasificador) sirve como filtro, dejando pasar únicamente información relevante, y gracias a que este proceso se aplica documento por documento se logra hacer un filtrado más completo y fino de cada uno de estos.

La tercera arquitectura se conforma de ambos módulos, i.e., contiene tanto al módulo supervisado como al no-supervisado. La diferencia de esta tercer arquitectura

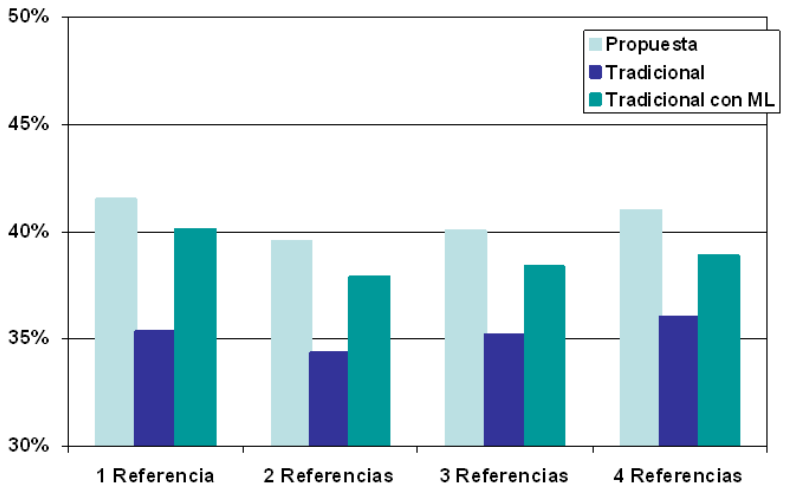

Figura 5.5: Puntaje ROUGE-1 para las diferentes configuraciones propuestas

con la idea original, es que los módulos se encuentran en orden inverso, es decir, los documentos de la colección pasan por el proceso de identificación de sub-temas y eliminación de redundancias en una primera etapa y posteriormente por el clasificador de oraciones relevantes (Ver figura 5.4).

Se hizo una comparación entre los resultados obtenidos con esta tercera configuración y los obtenidos con las dos primeras. Los resúmenes generados con la 3er configuración resultaron ser de mejor calidad en comparación contra los obtenidos con los de la 2a configuración, reafirmando así el hecho de que el uso del módulo supervisado ayuda al sistema a la identificación relevante.

Sin embargo estos resúmenes no resultaron mejores que los ya obtenidos con la arquitectura propuesta (Figura 5.1), lo cual nos permite afirmar que el proceso de identificación de información relevante debe hacerse en una primera etapa, debido a que el proceso de agrupamiento provoca que cierta información se pierda.

La figura 5.5 muestra de manera gráfica el desempeño obtenido por cada una de estas configuraciones. El valor que es mostrado en la gráfica corresponde al valor ROUGE-1 obtenido por cada configuración[3]. Los elementos del eje horizontal indican cuantos resúmenes de referencia fueron utilizados para la evaluación. Como es posible observarse, la configuración que siempre se mantiene por encima de las otras dos es

[3]Para ver con mayor detalle los resultados obtenidos tras evaluar contra 1,2 y 3 resúmenes de referencia los resúmenes generados con la 2a y 3a configuración refiérase al Apéndice A

la propuesta originalmente en este trabajo (figura 5.1).

La 1a y 3a configuración son las que hacen uso del módulo supervisado para la identificación de información relevante, como se puede observar éstas permiten obtener en todos los casos resúmenes de mayor calidad. En general, los resultados aquí presentados nos permiten concluir que el uso de técnicas de aprendizaje automático para la identificación de información relevante permite al sistema adaptarse a diferentes dominios e incluso a diferentes necesidades, y dado que el clasificador utiliza atributos independientes del lenguaje, el sistema se vuele entonces altamente portable.

Capítulo 6
Conclusiones

Se presentó una arquitectura basada completamente en técnicas de aprendizaje automático para desarrollar el sistema de generación de resúmenes de un solo documento. Con el fin de lograr los objetivos plateados en esta tesis, se hizo un estudio exhaustivo de distintos sistemas propuestos por diferentes grupos de investigación, del cual se logró la identificación de un conjunto de atributos considerados independientes del dominio y del lenguaje.

Un primer clasificador fue entrenado utilizando estos atributos como forma de representación de las oraciones. El análisis hecho a los resultados obtenidos nos llevo a concluir que: los atributos que estaban siendo utilizados, no proporcionaban evidencia suficiente al clasificador sobre la existencia o no, de información importante dentro de las oraciones de un documento. La propuesta hecha fue trabajar con atributos que consideraran la información del contexto para representar a las oraciones. Tras hacer los experimentos necesarios se concluyó que efectivamente el uso de secuencias de palabras (n-gramas) permiten al clasificador identificar de manera más eficiente las oraciones relevantes de un documento.

Es importante recalcar un par de cosas que se lograron hasta este punto, en primer lugar, el uso de n-gramas como forma de representación de las oraciones es algo novedoso, pues en la literatura no existe evidencia de sistemas que hagan uso de este tipo de atributos para determinar cuando las oraciones son relevantes. Otro punto importante al que hay que poner atención es al comportamiento que el sistema tuvo al momento de pasar de la representación hecha por medio de *palabras simples* a *secuencias de palabras o n-gramas*.

La representación de documentos por medio de secuencias de palabras es algo muy común en el área de clasificación de textos, mas sin embargo la técnica líder en el área corresponde a la representación por medio de palabras simples o BOW. Numerosos

estudios concluyen que el usar n-gramas no aporta beneficios a los sistemas de TC.

Retomando nuestro problema, lo que nuestro sistema necesitaba era hacer una clasificación a nivel de oraciones, en otras palabras, se necesitaba diferenciar las oraciones relevantes de aquellas que no lo fueran. Esto lo consideramos como un problema de clasificación más fino, lo cual no es el caso de la tarea de clasificación de textos. La aportación hecha en este punto fue el haber mostrado la pertinencia que tiene el hacer uso de n-gramas como forma de representación en problemas de clasificación más finos.

Gracias a las características de uno de los conjuntos de datos utilizados (i.e., DESASTRES), fue posible realizar un tipo de evaluación extrínseca. Para esto se midió el desempeño de un sistema de extracción de información al utilizar tanto los documentos completos, como los diferentes resúmenes generados por nuestro sistema, utilizando las diferentes formas de representación propuestas. Los resultados de esta evaluación demostraron que la calidad del contenido de los resúmenes generados utilizando n-gramas como forma de representación era mucho mayor a la calidad de aquellos generados utilizando algún otro tipo de representación. Con la realización de este tipo de evaluación se afirmó lo adecuado de la configuración del método supervisado, i.e., se concluye que la representación por medio de n-gramas aporta la evidencia necesaria al clasificador sobre la relevancia de las oraciones dentro de un documento.

Para poder generar resúmenes de múltiples documentos fue necesario hacer cambios a la arquitectura inicial. Estos cambios consistieron en agregar un módulo que permitiera al sistema trabajar con múltiples documentos. El módulo agregado se compone principalmente de un algoritmo de agrupamiento. Tradicionalmente los sistemas de generación de resúmenes de múltiples documentos emplean técnicas de agrupamiento, debido a que el objetivo y finalidad de éstos se asemejan mucho a los de un sistema generador de resúmenes de múltiples documentos, i.e., identificar los elementos que compartan características similares, agruparlos y presentarlos al usuario.

Así entonces, nuestra arquitectura se compone de un módulo supervisado el cual esta encargado de identificar las oraciones relevantes de cada documento, seguido de un módulo no-supervisado que consiste de un algoritmo de agrupamiento de datos, cuya principal finalidad es la de encontrar la información única y común a la colección de documentos, además de identificar y eliminar redundancias. El sistema que se propusó es novedoso debido a la adaptación del módulo supervisado, pues tradi-

cionalmente los sistemas de generación de resúmenes de múltiples documentos hacen uso únicamente de técnicas no supervisadas para la resolución de la tarea.

Para evaluar el sistema se hizo uso de una herramienta estándar cuyo principal objetivo es determinar la calidad de los resúmenes generados automáticamente. ROUGE ha demostrado ser una herramienta capaz de medir la correlación entre resúmenes generados por humanos y los resúmenes generados automáticamente. Los resultados obtenidos en la etapa de experimentación muestran que efectivamente nuestra arquitectura propuesta genera resúmenes más parecidos (en contenido) a aquellos creados por humanos.

Las principales conclusiones que se obtuvieron de los experimentos realizados son: en primer lugar, se demostró que el uso de un módulo basado en aprendizaje automático, para la identificación de información importante dentro de cada documento, permite al sistema generar resúmenes de mayor calidad. El no utilizar este módulo provoca que haya pérdida de información y en consecuencia que la calidad de los resúmenes generados disminuya.

En segundo lugar, se mostró la conveniencia que tiene el hacer en un primer paso el proceso de identificación de información importante en cada documento, esto por medio del módulo supervisado, (i.e., primero generar resúmenes individuales), y posteriormente realizar el proceso de identificación de información común y única dentro de los elementos de la colección (i.e., ejecutar el proceso de agrupamiento). El hacerlo de otra forma (i.e., el proceso inverso) provoca pérdida de información, debido a la naturaleza del algoritmo de agrupamiento, provocando que la calidad de los resúmenes disminuya. Sin embargo, con este experimento se comprobó una vez más que el uso del módulo supervisado permite identificar y recuperar las oraciones relevantes.

6.1. Restricciones y Desventajas

Debido a la naturaleza de la arquitectura propuesta, existen algunas restricciones y desventajas que deberán ser tomadas en cuenta para el posible trabajo futuro:

- El uso de un esquema de aprendizaje automático obliga a contar con un conjunto de datos etiquetados necesario para la fase de entrenamiento.
- El algoritmo estrella requiere de dos parámetros importantes para poder funcio-

nar adecuadamente. El primero es la medida de similitud que se utilizará para el cálculo de la matriz de similitudes. El segundo el umbral σ que ayudará a la creación del grafo umbralizado. El problema aquí, surge al momento de definir los valores de estos parámetros, pues no es posible asegurar cual es la mejor combinación.

- La colección de entrada al sistema debe ser una colección relacionada temáticamente, es decir, deben ser documentos que contengan información referente al mismo evento.

- Los resúmenes creados son construidos reutilizando porciones de información de los documentos originales, es decir, la etapa de generación no es realizada.

6.2. Trabajo Futuro

Tomando en cuenta las restricciones y desventajas que presenta la arquitectura, algunas ideas que se pretende explorar en un futuro son:

- Explorar técnicas de aprendizaje no supervisado. Se ha mencionado a lo largo del documento que la arquitectura propuesta es fácilmente portable debido al uso de atributos independientes del dominios e incluso del lenguaje. Sin embargo, la restricción para poder realizar esto es que debe existir un conjunto de datos previamente etiquetado con el cual la fase de entrenamiento se pueda realizar. Una opción a esta restricción es evaluar el desempeño del sistema utilizando técnicas de aprendizaje no supervisado [38].

- Realizar un análisis sobre diferentes atributos que aporten mayor información. Como se estudió en el capítulo 4 el uso de los atributos estadísticos/heurísticos por si solos no permitió obtener un buen desempeño en el sistema. Por otro lado, el uso de n-gramas permitió alcanzar un desempeño bastante bueno, sin embargo, hizo falta un análisis sobre el funcionamiento del sistema al hacer la combinación de estos dos conjuntos de atributos.

- Evaluar nuevos métodos de aprendizaje y diferentes medidas de similitud. El objetivo principal de la tesis fue definir una arquitectura de propósito general para la generación de resúmenes de múltiples documentos, por este motivo algunos

de los aspectos técnicos que dependen del dominio en estudio, como son métodos de aprendizaje supervisado y diferentes formas de medir la similitud entre documentos en la etapa de agrupamiento, no han sido aún evaluados. Ejemplos de métodos que proponemos evaluar son: probar ensambles de clasificadores y utilizar medidas de similitud que tomen en cuenta secuencias de palabras o la estructura de las oraciones.

- Evaluar el desempeño del sistema con otros métodos de agrupamiento. El algoritmo estrella genera estrellas que no son únicas. Un mismo grafo de similaridad puede ser capaz de generar diferentes grupos de estrellas, esto debido a que pueden existir más de un vértice con el mismo grado, es este caso el algoritmo escoge aleatoriamente uno de estos para generar la estrella. Ejemplos de métodos que proponemos evaluar es en primer lugar una versión mejorada del algoritmo estrella que elimina este problema, el *algoritmo estrella extendido* [20]. De manera adicional se propone evaluar el desempeño del sistema al utilizar diferentes métodos de agrupamiento, por ejemplo, k-means.

- Alcanzar la etapa de generación de lenguaje. Como se mencionó en un principio, nuestro sistema genera resúmenes basados en extractos, i.e., porciones de los documentos originales (palabras, oraciones, párrafos, etc.) son reutilizadas para crear el documento que se le entregará al usuario como resumen final. La etapa de generación consiste en tomar estos extractos y con ellos formar nuevas oraciones, generalmente más cortas, con el objetivo de entregar al usuario un resumen más coherente. El agregar la etapa de generación a nuestro sistema involucra que recursos lingüísticos más sofisticados deban ser agregados también al sistema, e.g., analizadores léxicos, sintácticos y/o semánticos, provocando así una dependencia del lenguaje.

Apéndice

Resultados de experimentos adicionales

Tabla A.1: Segunda configuración: Evaluación ROUGE contra un resumen de referencia

Configuración	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4	ROUGE-L
SUM-DICE ($\overline{x}$)	0.33779	0.04641	0.01408	0.00466	0.30043
SUM-DICE ($\overline{x} - \delta$)	0.33598	0.04164	0.00698	0.00265	0.29625
SUM-DICE ($\overline{x} + \delta$)	0.2703	0.04521	0.01607	0.00757	0.24181
SUM-COS ($\overline{x}$)	**0.35291**	**0.04801**	**0.01342**	**0.00468**	**0.031631**
SUM-COS ($\overline{x} - \delta$)	0.33598	0.04164	0.00698	0.00265	0.29625
SUM-COS ($\overline{x} + \delta$)	0.25452	0.04422	0.01519	0.00723	0.22791
SUM-JACC ($\overline{x}$)	0.32635	0.0428	0.00919	0.00235	0.29308
SUM-JACC ($\overline{x} - \delta$)	0.33598	0.04164	0.00698	0.00265	0.29625
SUM-JACC ($\overline{x} + \delta$)	0.22874	0.04346	0.0201	0.0116	0.20435
Propuesta	0.41475	0.08697	0.03248	0.01704	0.37291
Baseline 1	0.25111	0.04065	0.01594	0.0078	0.22983
Baseline 2	0.25322	0.0265	0.00776	0.00434	0.2372

Tabla A.2: Segunda configuración: Evaluación ROUGE contra dos resúmenes de referencia

Configuración	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4	ROUGE-L
SUM-DICE ($\overline{x}$)	0.31873	0.04661	0.01598	0.00643	0.28571
SUM-DICE ($\overline{x} - \delta$)	0.32853	0.04397	0.00582	0.00105	0.28914
SUM-DICE ($\overline{x} + \delta$)	0.25706	0.04802	0.01955	0.00956	0.23429
SUM-COS ($\overline{x}$)	**0.34318**	**0.04966**	**0.01381**	**0.00429**	**0.30172**
SUM-COS ($\overline{x} - \delta$)	0.32853	0.04397	0.00582	0.00105	0.28914
SUM-COS ($\overline{x} + \delta$)	0.23157	0.04459	0.01754	0.00904	0.21138
SUM-JACC ($\overline{x}$)	0.30524	0.03987	0.01014	0.00375	0.27315
SUM-JACC ($\overline{x} - \delta$)	0.32853	0.04397	0.00582	0.00105	0.28914
SUM-JACC ($\overline{x} + \delta$)	0.20923	0.04247	0.02159	0.01321	0.19066
Propuesta	0.3949	0.08969	0.03393	0.01926	0.35092
Baseline 1	0.24587	0.04328	0.01904	0.01009	0.22621
Baseline 2	0.25173	0.02492	0.00782	0.00473	0.23874

Tabla A.3: Segunda configuración: Evaluación ROUGE contra tres resúmenes de referencia

Configuración	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4	ROUGE-L
SUM-DICE ($\overline{x}$)	0.329	0.04426	0.01387	0.00465	0.28946
SUM-DICE ($\overline{x} - \delta$)	0.33014	0.04176	0.00568	0.00141	0.28977
SUM-DICE ($\overline{x} + \delta$)	0.25437	0.04177	0.01509	0.00703	0.22914
SUM-COS ($\overline{x}$)	**0.35182**	**0.04807**	**0.01207**	**0.00322**	**0.31003**
SUM-COS ($\overline{x} - \delta$)	0.33014	0.04176	0.00568	0.00141	0.28977
SUM-COS ($\overline{x} + \delta$)	0.23538	0.0404	0.0137	0.00703	0.2126
SUM-JACC ($\overline{x}$)	0.31435	0.03907	0.00926	0.00251	0.27749
SUM-JACC ($\overline{x} - \delta$)	0.33014	0.04176	0.00568	0.00141	0.28977
SUM-JACC ($\overline{x} + \delta$)	0.2106	0.03809	0.01747	0.01053	0.1902
Propuesta	0.40045	0.08305	0.03118	0.01711	0.35589
Baseline 1	0.248	0.04071	0.01614	0.00806	0.22763
Baseline 2	0.25251	0.02457	0.00832	0.0045	0.23727

Tabla A.4: Tercera configuración: Evaluación ROUGE contra un resumen de referencia

Configuración	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4	ROUGE-L
SUM-DICE ($\overline{x}$)	0.37491	0.0569	0.0199	0.00879	0.32831
SUM-DICE ($\overline{x} - \delta$)	0.33772	0.04175	0.00469	0.00145	0.30091
SUM-DICE ($\overline{x} + \delta$)	**0.40101**	**0.06264**	**0.01697**	**0.00768**	**0.35437**
SUM-COS ($\overline{x}$)	0.35412	0.0426	0.01031	0.00473	0.3113
SUM-COS ($\overline{x} - \delta$)	0.33772	0.04175	0.00469	0.00145	0.30091
SUM-COS ($\overline{x} + \delta$)	0.37468	0.05625	0.01479	0.00472	0.33566
SUM-JACC ($\overline{x}$)	0.36852	0.04753	0.00918	0.00322	0.32478
SUM-JACC ($\overline{x} - \delta$)	0.33772	0.04175	0.00469	0.00145	0.3091
SUM-JACC ($\overline{x} + \delta$)	0.38429	0.05846	0.01566	0.00685	0.34078
Propuesta	0.41475	0.08697	0.03248	0.01704	0.37291
Baseline 1	0.25111	0.04065	0.01594	0.0078	0.22983
Baseline 2	0.25322	0.0265	0.00776	0.00434	0.2372

Tabla A.5: Tercera configuración: Evaluación ROUGE contra dos resúmenes de referencia

Configuración	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4	ROUGE-L
SUM-DICE ($\overline{x}$)	0.35562	0.05726	0.02203	0.00858	0.31505
SUM-DICE ($\overline{x} - \delta$)	0.32794	0.04515	0.00427	0.00105	0.2943
SUM-DICE ($\overline{x} + \delta$)	**0.37887**	**0.05684**	**0.01662**	**0.00918**	**0.33836**
SUM-COS ($\overline{x}$)	0.33879	0.04177	0.01102	0.00469	0.29345
SUM-COS ($\overline{x} - \delta$)	0.32794	0.04515	0.00427	0.00105	0.2943
SUM-COS ($\overline{x} + \delta$)	0.35576	0.05364	0.01392	0.00485	0.31881
SUM-JACC ($\overline{x}$)	0.34659	0.04249	0.00696	0.00161	0.30477
SUM-JACC ($\overline{x} - \delta$)	0.32794	0.04515	0.00427	0.00105	0.2943
SUM-JACC ($\overline{x} + \delta$)	0.3763	0.06069	0.01981	0.00974	0.33812
Propuesta	0.3949	0.08969	0.03393	0.01926	0.35092
Baseline 1	0.24587	0.04328	0.01904	0.01009	0.22621
Baseline 2	0.25173	0.02492	0.00782	0.00473	0.23874

Tabla A.6: Tercera configuración: Evaluación ROUGE contra tres resúmenes de referencia

Configuración	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4	ROUGE-L
SUM-DICE ($\overline{x}$)	0.36176	0.0522	0.01813	0.00787	0.31837
SUM-DICE ($\overline{x} - \delta$)	0.32978	0.04287	0.00499	0.00177	0.29571
SUM-DICE ($\overline{x} + \delta$)	**0.38329**	**0.05499**	**0.01593**	**0.00782**	**0.33705**
SUM-COS ($\overline{x}$)	0.34547	0.04012	0.00999	0.00441	0.29939
SUM-COS ($\overline{x} - \delta$)	0.32978	0.04287	0.00499	0.00177	0.29571
SUM-COS ($\overline{x} + \delta$)	0.3608	0.04979	0.01283	0.00429	0.32206
SUM-JACC ($\overline{x}$)	0.35371	0.03951	0.00643	0.00179	0.30971
SUM-JACC ($\overline{x} - \delta$)	0.32978	0.04287	0.00499	0.00177	0.29571
SUM-JACC ($\overline{x} + \delta$)	0.38087	0.05782	0.0166	0.00745	0.33637
Propuesta	0.40045	0.08305	0.03118	0.01711	0.35589
Baseline 1	0.248	0.04071	0.01614	0.00806	0.22763
Baseline 2	0.25251	0.02457	0.00832	0.0045	0.23727

Bibliografía

[1] H. Ahonen-Myka. Discovery of frequent word sequences in text source. In *Proceedings of the ESF Exploratory Workshop on Pattern Detection and Discovery*, pages 111–121, London, UK, 2002.

[2] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study: Final report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pages 194–218, 1998.

[3] J. Aslam, K. Pelekhov, and D. Rus. A practical clustering algorithm for static and dynamic information organization. In *Proceedings of the 1999 Symposium on Discrete Algorithms*, pages 208–217, 1999.

[4] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrival*. Addison Wesley, 1999.

[5] M. Banko, V. O. Mittal, and M. J. Witbrock. Headline generation based on statistical translation. In *Proceedings of the 38th Annual Conference for Computational Linguistics (ACL 2000)*, pages 111–121, Hong Kong, 2000.

[6] M. Banko and L. Vanderwende. Using n-grams to understand the nature of summaries. In *Proceedings of the Human Technology Conference (HLT-NAACL-2004)*, Boston, MA, 2004.

[7] R. Barzilay, K. R. McKeown, and M. Elhadad. Using lexical chains for text summarization. In *Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*, pages 11–121, Madrid, Spain, 1997.

[8] R. Barzilay, K. R. McKeown, and M. Elhadad. Information fusion in the context of multi-document summarization. In *Proceedings of the 37th Association for Computational Linguistics*, pages 550–557, Maryland, 1999.

[9] P. B. Baxendale. Man-made index for technical literature- an experiment. *IBM Journal of Research and Development*, volume 2(number 4):pages 354–361, 1958.

[10] R. Bekkerman and J. Allan. Using bigrams in text categorization. Technical Report IR-472, 10 pages, Center for Intelligent Information Retrieval (CIIR), University of Massachusetts, 2003.

[11] R. Brandow, K. Mitze, and L. Rau. Automatic condensation of electronic publications by sentence selection. *Information Processing and Managment*, volume 31(number 5):pages 675–685, 1999.

[12] C. Buckley. Implementation of the SMART information retrieval system. Technical Report TR 85-686, 37 pages, Cornell University. Department of Computer Science, 1985.

[13] C. Buckley and C. Cardie. Using EMPIRE and SMART for high-precision IR and summarization. In *Proceedings of the TIPSTER Text Phase III 12-Month Workshop*, pages 107–121, San Diego, CA, 1997.

[14] W. B. Canvar and J. M. Trenkle. N-gram-based text categorization. In *Proceedings of the third Annual Symposium on Document Analysis and Information retrieval*, pages 161–169, Nevada, Las Vegas, 1994.

[15] W. T. Chuang and J. Yang. Text summarization by sentence segment extraction using machine learning algorithms. In *Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Current Issues and New Applications*, pages 454–457, London, UK, 2000.

[16] R. M. Coyotl-Morales, M. Montes-y-Gómez, and L. Villaseñor-Pineda. Authorship attribution using word sequences. In *Proceedings of the 11th Iberoamerican Congress on Pattern Recognition (CIARP 2006)*, pages 844–853, Cancun, Mexico, 2006.

[17] R. C. Dubes. *Handbook of Pattern Recognition and Computer Vision*, chapter Cluster analysis and related issues, pages 3–32. World Scientific Publishing Co., Inc., River Edge, NJ., 1993.

[18] H. P. Edmundson. New methods in automatic extracting. *Journal of the Association for Computing Machinery*, volume 16(number 2):pages 264–285, 1969.

[19] J. Fürnkranz. A study using n-gram features for text categorization. Technical Report OEFAI-TR-98-30, 10 pages, Austrian Institute for Artificial Intelligence, Wien, Austria, 1998.

[20] R. Gil-García, J. M. Badia-Contellens, and A. Pons-Porrata. Parallel algorithm for extended star clustering. In *Proceedings of the 9th Iberoamerican Congress on Pattern Recognition*, pages 402–409, Mexico, 2004.

[21] D. A. Grossman and O. Frieder. *Information Retrieval, Algorithms and Heuristics*. Springer, second edition edition, 2004.

[22] L. Hasler, C. Orasan, and R. Mitkov. Building better corpora for summarization. In *Proceedings of Corpus Linguistics 2003*, pages 309–319, Lancaster, UK, 2003.

[23] E. Hernández-Reyes, J. F. Martinez-Trinidad, J. A. Carrasco-Ochoa, and R. A. Garcia-Hernández. Document representation based on maximal frequent sequence sets. In *Proceedings of the 11th Iberoamerican Congress on Pattern Recognition (CIARP 2006)*, pages 854–863, Cancun, Mexico, 2006.

[24] E. Hovy. *The Oxford Handbook of Computational Linguistics*, chapter Text Summarization, pages 582–598. Oxford, 2003.

[25] E. Hovy and C.-Y. Lin. *Advances in Automatic Text Summarization*, chapter Automated text summarization in SUMMARIST, pages 81–94. MIT Press, Cambridge, 1999.

[26] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall advanced reference series. Prentice-Hall Inc., Upper Saddle River, NJ., 1988.

[27] A. K. Jain, M. Murty, and P. Flynn. Data clustering: A review. In *ACM Computing Surveys*, volume 31, pages 264–323. ACM, September 1999.

[28] K. Knight and D. Marcu. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, volume 139(number 1):pages 91–107, 2002.

[29] J. Kupiec, J. Pedersen, and F. Chen. A trainable document summarizer. In *Proceedings of the 18th ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 68–73, 1995.

[30] D. D. Lewis. Evaluating text categorization. In *Proceedings of the Speech and Natural Language Workshop*, pages 312–318, Asilomar, CA, 1991.

[31] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Proceedings of Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL*, Barcelona, Spain, 2004.

[32] C.-Y. Lin and E. Hovy. From single to multi-document summarization: A prototype system and its evaluation. In *Proceedings of the 40th Conference of the Association of Computational Linguistics*, pages 457–464, Philadelphia, 2002.

[33] C.-Y. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the Human Technology Conference (HLT-NAACL-2003)*, pages 71–78, Edmonton, Canada, 2003.

[34] H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, volume 2(number 2):pages 159–165, 1958.

[35] D. Marcu. *The Retorical Parsing, Summarization, and Generation of Natural Language Texts*. Ph.D. thesis, University of Toronto, 1997.

[36] D. Marcu. Improving summarization through rethorical parsing tuning. In *Proceedings of the COLING-ACL'98 Workshop on Very Large Corpora*, pages 10–16, Montreal, 1998.

[37] D. Marcu and L. Gerber. An inquiry into the nature of multidocument abstracts, extracts and their evaluation. In *Proceedings of the NAACL-2001 Workshop on Automatic Summarization*, pages 1–8, Pittsburg, USA, 2001.

[38] A. Massih-Reza and P. Gallinari. Self-supervised learning for automatic text summarization by text-span extraction. In *Proceedings of the 23rd BCS European Annual Colloquium on Information Retrieval*, pages 16–25, 2001.

[39] K. R. McKeown, J. Klavans, V. Hatzivassilouglou, R. Barzilay, and E. Eskin. Towards multidocument summarization by reformulation: progress and prospects. In *Proceedings of the 16th National Conference of the American Association For Artificial Intelligence (AAAI-1999)*, pages 453–460, 1999.

[40] K. R. McKeown and D. R. Radev. Generating summaries of multiple news articles. In *Proceedings of the 18th Anual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 74–82, 1995.

[41] T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[42] J. L. Neto, A. A. Freitas, and C. A. A. Kaestner. Automatic text summarization using a machine learning approach. In *Proceedings of the 16th Brazilian Symposium on Artificial Intelligence*, pages 205–215, Porto de Galinhas/Recife, Brazil, 2002.

[43] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Anual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318, Philadelphia, 2002.

[44] D. R. Radev, H. Jing, and M. Budzikowska. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation and user studies. In *ANLP/NAACL Workshop on Summarization*, pages 21–30, Seattle, USA, 2000.

[45] D. R. Radev, H. Jing, M. Stys, and D. Tam. Centroid-based summarization of multiple documents. In *Information Processing and Management*, pages 919–938, 2004.

[46] D. R. Radev and K. R. McKeown. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, volume 24(number 3):pages 469–500, 1998.

[47] D. R. Radev, K. R. McKeown, and E. Hovy. Introduction to the special issue on summarization. *Computational Linguistics*, pages 359–408, 2002.

[48] F. Sebastiani. Machine learning in automated text categorization. In *ACM Computing Surveys*, volume 34, pages 1–47, 2002.

[49] C. E. Shannon. Prediction and entropy of printed english. In *Bell System Technical Journal*, pages 50–64, January 1951.

[50] A. Tellez-Valero, M. Montes-y-Gómez, and L. Villaseñor-Pineda. A machine learning approach to information extraction. In *Proceedings of the Sixth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2005)*, pages 535–538, Mexico City, Mexico, 2005.

[51] S. Teufel and M. Moens. Sentence extraction as a classification task. In *Proceedings of the ACL Workshop on Intelligent Text Summarization*, pages 58–65, Madrid, España, 1997.

[52] C. van Rijsbergen. *Information Retrieval*. Butterworths, London, 1979.

[53] E. Villatoro-Tello, M. Montes-y-Gómez, and L. Villaseñor-Pineda. Using word sequences as features for text summarization. In *Proceedings of the 9th International Conference on Text, Speech and Dialogue (TSD 2006)*, pages 293–300, Brno, Czech Republic, 2006.

[54] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2000.