Universidad de las Ciencias Informáticas

Facultad 3

Titulo: Biblioteca de Clases para la creación automática de resúmenes extractos.

Trabajo de Diploma para optar por el titulo de
Ingeniero en Ciencias Informáticas

Autores: Dailin Benavides Jorge
Yarisleidis Fernández Rivera

Tutor: Manuel Vázquez Acosta.

Ciudad de la Habana
Julio, 2007

DECLARACIÓN DE AUTORÍA

Nosotras declaramos que somos las únicas autoras de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Autores:	Tutor:
Dailin Benavides Jorge	Manuel Vázquez Acosta
Yarisleidis Fernández Rivera	

"Lo que da al hombre el poder no es el mero conocimiento que viene del uso de los sentidos,
sino, ese otro conocimiento más profundo que se llama Ciencia"

AGRADECIMIENTOS

Todo lo bueno del mundo es demasiado poco cuando se desea lo mejor para alguien a quien se estima, por eso agradezco a todo el que de una forma u otra ha hecho posible la realización de este sueño. En especial:

A Manuel nuestro tutor, por haber sido nuestro guía y apoyarnos en todo momento.

A mis madres Olga, Milagros por estar junto a mí en cada momento de mi vida y haber depositado en mí toda la confianza del mundo. "Las adoro"

A mi padre Abelardo por ser el mundo para mí y haberme apoyado aun cuando ha estado ausente.

A mis hermanos y a mi padrastro por siempre estar presente en mi vida.

A Luna, Yari y Yesla por convertirse en mi familia y quererme y apoyarme siempre.

A Javier por haberse mantenido junto a mí en todo momento y darme el amor y la comprensión necesaria cuando el mundo se venía abajo.

A Yuneisi, Aneyvis, Osmany, Arturo, Anita y Javier Morales (el loko) por ser mis amigos. Al resto de mis amistades que dieron su granito de arena para que este sueño fuera posible.

A todos mis profesores, por todos los conocimientos transmitidos y en especial a Pascual y Yoansy.

A Darel porque sin el gran parte de este proyecto no se hubiera realizado.

A mi tía (blanca), mi abuela y mis tíos por darme todo ese cariño.

A mi papá, mi hermana y mi madrastra, a mis vecinos y en especial a Rolandito y a mi suegro, gracias por tu ayuda.

Dailin Benavides Jorge

A Manuel, gracias por tu apoyo

A mis padres, gracias por estar a mi lado siempre, son lo mejor de mi vida.

A mis hermanas, las quiero mucho, mucho.

A Leo, mi novio de toda la vida, por darme un poco de trabajo y apoyarme siempre.

A mi familia, que en todo momento me dio su apoyo.

A Darel, sabes que siempre te estaré agradeciendo toda la ayuda que nos brindaste en este momento tan importante de mi vida.

A Dai, Yuni, Yesla, Luna, Ane, por ser más que mis amigas, mis otras hermanas, siempre las querré.

A Javier Ramírez, gracias por tu ayuda.

A Javier Morales, Osmany y Didier por sus locuras.

A Marlen, Floralis, Niurvis y Yanelis por ser mis hermanitas.

A Pascual y a Yoansy, profes gracias por su ayuda.

A mi suegro, por demostrarme su cariño y hacerme sentir como una hija más.

A todos, los que de una forma u otra han contribuido a que este sueño se haga realidad.

Yarisleidis Fernández Rivera

DEDICATORIA

A mis madres, hermanos y a mi padrastro, los quiero muchísimo.

Y en especial a la memoria de mi padre Abelardo que aun sin estar presente lo llevo siempre en mi alma y mi corazón.

A todas las personas que quiero con el alma y a las cuales va dedicada esta tesis solo quiero decirle que las llevo siempre en el corazón.

Dailin Benavides Jorge

A mis padres, por haberme apoyado en cada momento, y haber creído en mí en el más difícil, este logro es para ustedes, que saben querer y educar a la vez.

A mi familia.

A mis profesores a todo lo largo de mi vida estudiantil y en especial a los de la Secundaria Básica Luís Manuel Pozo Nápoles por haber sido mis guías y haber confiado en mi: "Lo que soy se los debo a ustedes".

A mis hermanas, novio y amigos.

A mi prima Inés y mi abuelo Juan, que ya no están junto a mí, y los extraño mucho, y que se estarían muy orgullosos de mí.

Yarisleidis Fernández Rivera

RESUMEN

En este trabajo se utiliza las técnicas de sumarización de textos y se implementan algoritmos para la construcción automática de resúmenes de textos en una biblioteca de clases, se propone una arquitectura básica de la biblioteca de clases y se realiza una validación de los resultados por medio de métricas para la evaluación.

INDICE

Introducción

La proliferación de la información en Internet, el gran volumen de ésta que se genera cada día y el acceso desde cualquier lugar, constituyen las características fundamentales de la era de la información que estamos viviendo. No obstante, no debe olvidarse que en los extremos de toda comunicación sigue habiendo seres humanos para procesar la información de salida o entrada (CARCEDO 2000).

Los seres humanos usamos nuestro cerebro para procesar las informaciones, modificarlas y producir otras nuevas, pero en la actualidad a consecuencia del auge de las tecnologías de la información e Internet, nos ha surgido una tarea difícil, pues los humanos no estamos preparados para procesar cantidades masivas de información y encontrar asuntos de interés (GARCÍA 2005).

Hoy día, cuando hacemos una consulta en Internet, como respuesta de los buscadores, recibimos una gran cantidad información y quizás deseamos solamente quedarnos con los aspectos que nos son relevantes de dicha información. Antiguamente los cursos de lectura rápida pudieron ser una alternativa cuando teníamos que leer una decena de textos para obtener información. En la actualidad se requiere obtener la información importante que está en miles o millones de textos y, por más rápida que sea nuestra lectura, obtener la esencia que nos interesa de estos textos llevaría más tiempo del que disponemos. Por esta razón, se han realizado investigaciones para desarrollar herramientas computacionales que permitan identificar los aspectos que son importantes en los textos (PORTILLA 2005).

La información, para convertirse en conocimiento, siempre ha requerido de su procesamiento y comprensión mediante los humanos, cabe resaltar entonces que nosotros para llevar a cabo ese proceso lo hacemos primero seleccionando la información de nuestro interés, la cual constituirá luego la base para elaborar un resumen que es una vía de obtener información en síntesis. A medida que avanza el desarrollo de la información obtener un resumen como vía de sintetizarla nos consume mucho tiempo (quizás más del que disponemos) y abundantes recursos humanos, es por eso que hemos tenido que recurrir a la creación de sistemas

automáticos generadores de resúmenes que permitan procesar la información en el menor tiempo posible.

El proceso de construcción automática de resúmenes de documentos consiste en, dados una fuente de información (uno o más documentos) y un demandante (usuario o aplicación), extraer el contenido de la fuente de información y presentarlo al demandante de forma condensada, comprensible, y que satisfaga sus necesidades. Por forma condensada se debe entender cualquier forma que pueda adquirir el contenido de la fuente de información en operaciones de selección, agregación o generalización (ANAYA *et al.* 2006).

El resumen automático de textos ha tenido un lento desarrollo por varios años, y recientemente se ha vuelto interesante debido al uso incremental de Internet. Ejemplos de usos que puede tener esta técnica:

- Resumir noticias a formato SMS o WAP (Wireless Application Protocol) para teléfono móvil/PDA (Personal Digital Assistance).
- En motores de búsqueda, para presentar descripciones sintetizadas de los resultados de las búsquedas.
- Para buscar en diversos lenguajes y obtener un resumen traducido automáticamente del resumen automático.

El resumen ha de cumplir no sólo la función de proporcionar elementos que estimulen o rechacen la consulta del documento original, sino que debe facilitar la obtención de un primer nivel de asimilación del problema que se aborda. La construcción automática de resúmenes de textos resulta inestimable para aquellos usuarios que tratan con grandes cantidades de documentos y precisan de una herramienta que les permita determinar la información más relevante de un texto o un conjunto de textos a fin de discrimar aquellos a los que dedicar su atención (DELORT et al.).

La construcción automática de resúmenes de documentos comparte propiedades con la obtención y la extracción de información. La obtención de información no es más que la recuperación, de entre un conjunto de documentos, aquellos que responden a determinados criterios y la extracción de información

consiste en extraer aquella que nos interesa de uno o varios documentos y generar a partir de ésta un nuevo documento que la contenga únicamente (ROCA 2001).

Actualmente existen técnicas para el análisis automático del contenido textual, las cuales procesan el lenguaje natural de los documentos para obtener desde descriptores y palabras claves hasta resúmenes, podemos decir que estas técnicas se dividen para formar parte de las dos líneas básicas de investigación en la confección automática de resúmenes que son: los resúmenes por extracción y por abstracción.

Los resúmenes por extracción actúan sobre una fuente (o varias), vistos como una colección de oraciones; y de estas oraciones se extraen y presentan aquéllas consideradas más relevantes o que responden a determinados criterios de un usuario particular (o grupo de usuarios). En este caso el resumen es un subconjunto de las oraciones del texto original (MANI and MAYBURY 1999; ROCA 2001) que no deben haber sido modificadas.

Los resúmenes por abstracción utilizan técnicas más sofisticadas de tratamiento del lenguaje, ya que el resultado no consiste en determinadas oraciones entresacadas del texto original, sino en un documento de nueva redacción generado a partir del tratamiento de la información contenida en el primero. Las técnicas necesarias para la aplicación de esta estrategia distan de haber obtenido resultados satisfactorios y pertenecen aún al campo de la investigación básica. Es por ello que, hasta el momento, los avances más significativos se han realizado en el campo de los resúmenes por extracción (ROCA 2001).

En la Universidad de las Ciencias informáticas (UCI), la creación automática de resúmenes nos sería de gran utilidad en los observatorios de ciencia y tecnología (para lograr una rápida localización de la información deseada), en los forum científicos, arbitraje, en los sistemas de recuperación de información, etc. Esta es una universidad caracterizada por la producción de software informáticos para empresas nacionales e internacionales, actualmente es muy utilizada la tecnología Plone/Zope en su producción para la creación de sistemas de

información y debido a que en estos momentos se está desarrollando un observatorio tecnológico bajo esta tecnología resultaría de gran utilidad tener creada una herramienta que nos permita generar automáticamente resúmenes.

Por lo anteriormente expuesto, se deriva el siguiente **problema** que se manifiesta en ausencia de implementaciones en Python de algoritmos que permitan la generación automática de resúmenes.

Para dar solución a este problema hemos enfocado el **objeto de estudio** hacia una investigación de la minería de texto, como área identificada por el descubrimiento de conocimientos en los textos, y la sintetización automática de textos.

Minimizando el área de extensión de nuestro trabajo y concentrándonos en nuestro problema enfocamos el **campo de acción** hacia la investigación de La generación automática de resúmenes, sus algoritmos y la creación de herramientas en Python para hacer resúmenes extractos.

En conformidad con el problema, nuestro **objetivo** es:

Desarrollar una biblioteca de clases en el lenguaje de programación Python que contenga implementaciones de algoritmos de generación automática de resúmenes.

Objetivos específicos:

1. Realizar un estudio y análisis de la información actualizada que nos permita construir el marco teórico del estado actual de los algoritmos de generación automática de textos.
2. Implementar algunos de los algoritmos revisados para definir la interfaz de la biblioteca.
3. Realizar un análisis de los resultados obtenidos del algoritmo monodocumento y validación del resultado.

Hipótesis

Con el desarrollo de una biblioteca de clases en Python con algunos algoritmos de generación automática de resúmenes se podrá obtener una herramienta para la creación automática de resúmenes que pueda ser integrada con aplicaciones que cumplan esos requerimientos.

Capítulo I: Fundamentación Teórica.

Este capítulo contiene un estudio sobre los resúmenes, profundizando en los métodos y algoritmos para su construcción.

Capítulo II: Descripción e implementación de los algoritmos propuestos para construir la biblioteca. Evaluación de los resultados.

Contiene el diseño de clases, descripción e implementación de los algoritmos de generación automática de resúmenes seleccionados para desarrollar la biblioteca de clases, además de la evaluación realizada a los resultados de los algoritmos implementados.

Capítulo 1 : Fundamentación Teórica

1.1 Introducción

El aumento exponencial de la información y la imposibilidad de sintetizarla y procesarla por parte de los humanos ha dado la necesidad de crear automáticamente resúmenes. El presente capítulo tiene el objetivo de estudiar los resúmenes, describir diversos algoritmos existentes para la creación automática de resúmenes extractos.

Procesamiento Humano del Lenguaje Textual

El lenguaje es un proceso comunicativo donde emisor y receptor procesan determinada información en función de un conocimiento lingüístico y un conocimiento compartido del mundo (WINOGRAD 1983).

La lingüística textual ha modelado los procesos de producción y recepción textual que llevan a cabo las personas en la actividad comunicativa. La recepción textual puede modelarse como una serie de fases dominantes del procesamiento que se recorre en dirección contraria a la producción (BEAUGRANDE *et al.* 1997), ocurre en actividades como la lectura, la elaboración de resúmenes o síntesis y la extracción de terminología a partir de textos, las cuales son consideradas como capacidades cognitivas humanas.

El proceso de recepción textual interviene en la recuperación de información pues está basado en el contenido de los documentos. Un documento recuperable es aquel que posee una descripción, una catalogación en una clasificación previamente establecida y un análisis de su contenido(ARNTZ and PICHT 1995). La descripción del contenido de un documento que consiste en la explicitación de los elementos más representativos de la información que transmite, soporta dos actividades: la indización y la elaboración de resúmenes. La indización es una operación terminológica, pues identifica explícitamente las unidades y expresiones representativas del contenido. La elaboración de resúmenes consiste en una operación de condensación, ya que selecciona la información más relevante del contenido textual y la expresa de manera sintética.

El resumen y el procesamiento del Lenguaje Natural (PLN)

Al implementar programas que resuman de manera automática un documento o conjunto de ellos se esta creando un programa que imita el comportamiento y la comprensión humana. La ciencia que intenta la creación de programas para máquinas que imiten el comportamiento y la comprensión humana se denomina *inteligencia artificial.*

El PLN es una disciplina de la *Inteligencia Artificial* y la rama ingenieril de la lingüística computacional. El PLN se ocupa de la formulación e investigación de mecanismos eficaces computacionalmente para la comunicación entre personas o entre personas y máquinas por medio de lenguajes naturales.

"El Procesamiento del Lenguaje Natural es el uso de computadoras para entender lenguajes (naturales) humanos tales como inglés, francés o japonés. Por 'entender' no se quiere decir que el computador tenga pensamientos, sentimientos y conocimientos humanizados, sino que el computador pueda reconocer y usar información expresada en lenguaje humano" (COVINGTON 1994).

Un sistema de PLN encapsula un modelo del lenguaje natural en algoritmos apropiados y eficientes, en donde las técnicas de modelado están ampliamente relacionadas con conocimientos de muchos otros campos (MANARIS and SLATOR 1996).

La generación de un resumen de alta calidad requiere las técnicas PLN tales como análisis del discurso, inferencia del conocimiento del mundo, análisis semántico, y generación del lenguaje que todavía están bajo investigación. Consecuentemente, la mayor parte de los sistemas automatizados actuales de resumen de un texto producen los extractos en vez de abstractos. Un extracto es una colección de las oraciones importantes en un documento, reproducidas textualmente (LIN 1999).

La minería de texto en la confección de resúmenes

La minería de textos (*text mining*) intenta proveer una visión selectiva y perfeccionada de la información contenida en documentos, sacar consecuencias

para la acción y detectar patrones no triviales e información sobre el conocimiento almacenado en las mismas (BERRY 2004). Es el proceso encargado del descubrimiento de conocimientos que no existían explícitamente en ningún texto de la colección, pero que surgen de relacionar el contenido de varios de ellos (HEARST 1999; KODRATOFF 1999). Está conformada por la recuperación y extracción de información, el análisis de textos, el resumen, el agrupamiento, la categorización y la clasificación de documentos.

Un problema de la minería de textos, en la actualidad, es encontrar documentos relevantes sobre la información que necesitamos, pero realmente la situación es incluso más compleja. Después de encontrar algunos documentos relevantes, el problema es encontrar el tiempo necesario para leerlos, por tanto, resumirlos puede contribuir a realizar una revisión en tiempo de los aspectos relevantes de ellos (GARCÍA 2005).

1.2 Resúmenes Automáticos

Un resumen es la representación sintetizada y precisa de un texto fuente a partir de uno o múltiples documentos. Debe ser de fácil lectura y comprensión, tener menor longitud que el texto fuente y ofrecer información relevante al usuario.

Los resúmenes se clasifican en dos grandes grupos, los extractos y los abstractos. En este documento se profundizará en el grupo de los extractos y se le llama extracto a un resumen que contiene la información que esta contenida en el texto fuente.

Los primeros trabajos en generación automática de resúmenes de texto datan de finales de los 50 y de la década de los 60. Durante las dos décadas siguientes no hubo gran interés por el tema. Sin embargo, a partir de los 90 la investigación en el área creció de una forma significativa. Se han desarrollado un número importante de técnicas de generación automática de resúmenes de texto que, se pueden clasificar según el nivel del análisis lingüístico realizado sobre la fuente en:

Las técnicas de extracción: Se caracterizan por un análisis superficial del texto fuente, no profundizando más allá del nivel sintáctico. El resultado es un resumen entre el 5 y 30 por ciento de la longitud fuente del documento generado a partir de la extracción de elementos significativos del texto original. Estos elementos pueden ser, por ejemplo, palabras, oraciones o párrafos. Los problemas de incoherencia, que probablemente se producirán al separar estos elementos de su contexto, pueden mitigarse mediante un proceso de revisión del texto seleccionado. La mayor ventaja que tiene este enfoque es que resulta muy robusto y fácilmente aplicable a contextos de propósito general, ya que, su independencia del dominio, e incluso del género de los documentos, es muy alta (LÓPEZ, MANUEL J MAÑA 2003).

Las técnicas de abstracción: Se basan en un análisis semántico de la fuente al menos a nivel de frase. Este análisis, de mayor profundidad que en el caso anterior, permite obtener una representación semántica del contenido del texto que puede utilizarse, junto a técnicas de generación de lenguaje natural, para confeccionar el resumen. Las fases de análisis y generación requieren gran cantidad de conocimiento del dominio, por lo que este tipo de sistemas solo es aplicable a ámbitos muy concretos y enteramente conocidos (LÓPEZ, MANUEL J MAÑA 2003).

Por otra parte, (LÓPEZ, MANUEL J. MAÑA *et al.* 1998) clasifican las técnicas empleadas en la generación de resúmenes en *estadísticas o simbólicas*, utilizadas en aplicaciones generales o específicas, respectivamente. Los sistemas estadísticos independientes del dominio generan resúmenes inconsistentes e incompletos.

Actualmente existen cuatro áreas de investigación de resúmenes automáticos, estas son el desarrollo de resumidores que puedan abordar múltiples documentos, documentos multimedia, documentos con múltiples lenguajes y documentos híbridos que mezclen el idioma y el tipo (multimedia o no) (HAHN and MANI 2000).

1.2.1 Clasificación y tipo de los resúmenes.

Según su alcance: Resumen monodocumento y resumen multidocumento. Resumen monodocumento si se trata de un único documento y multidocumento si el resumen fuese a partir de varios documentos.

Según el tipo de usuario al que esta destinado: Resumen genérico y resumen enfocado a un usuario, a un tópico o a una consulta (ANGHELUTA *et al.* 2004). Los resúmenes genéricos intentan recoger los temas principales de un documento y los adaptados al usuario confeccionan el resumen de acuerdo a los intereses del usuario al que va dirigido, esto es, sus conocimientos previos, sus ámbitos de interés o sus necesidades de información. Los resúmenes genéricos se caracterizan por seleccionar oraciones importantes del documento, pero sin tener en cuenta qué información puede ser de interés para el usuario o cuál es su dominio de conocimiento.

Según su función: Resumen informativo e Indicativo (BORKO and BERNIER 1975). Los denominados indicativos tienen como propósito anticipar al lector el contenido del texto y ayudarle a decidir sobre la relevancia del documento fuente y los informativos pretenden sustituir al texto completo incorporando toda la información nueva o trascendente. Los resúmenes informativos constituyen un subtipo de los indicativos (LÓPEZ, MANUEL J MAÑA 2003).

1.2.2 Conceptos fundamentales en la confección de resúmenes.

Los resúmenes *indicativos* e *informativos* se distinguen atendiendo a la cantidad de información que contiene el resumen o lo que se puede denominar **Significación semántica (S):** Es la cantidad de información que contiene el resumen (MANI *et al.* 2001). Está en función de su tasa de condensación o compresión, definida como el cociente entre la longitud del resumen y la longitud del texto fuente, y de la relevancia de su contenido. La tasa de compresión es un número real perteneciente al intervalo (0,1). Debido a la cantidad de texto que es excluida del resumen, una razón de compresión cercana a cero se considera alta, mientras que una cercana a 1 se considera baja (ANAYA *et al.* 2006). El valor S es mayor para resúmenes cortos.

$$S = \left(1 - \frac{longitud(R)}{longitud(T)}\right)\left(\frac{peso(R)}{peso(T)}\right)$$

Expresión 1-1: Calcula la significación semántica S de un resumen.

Como puede observarse en la Expresión (1-1), se ha supuesto la existencia de una función numérica *peso* capaz de medir la importancia de una oración. Este peso debería reflejar tanto el carácter informativo del contenido del documento como la relevancia del mismo para la aplicación a la que está destinada el resumen. Aunque esta noción es, por desgracia, difícilmente definible desde un punto de vista teórico, es la base de buena parte del trabajo en generación automática de resúmenes. Gran parte de la investigación realizada tiene como objetivo la definición de funciones que proporcionen una estimación de la importancia de determinados elementos del texto fuente. Sin embargo, esta noción de importancia o relevancia no es exclusiva del ámbito que nos ocupa sino que también se utiliza en otros como, por ejemplo, en Recuperación de Información (RI)

La relevancia: Es el cociente entre el peso del resumen y el peso del texto fuente, teniendo presente que anteriormente se debe haber definido una función de peso para medir la relevancia de cada oración. Esto es, si se representa un texto *T* como una secuencia de oraciones $T = P_1, P_2, ..., P_n$, probablemente enlazadas mediante relaciones de discurso, se podrá representar un resumen *R* como un subconjunto de oraciones de *T*.

La consistencia: Es la manera en que las oraciones aparecen ligadas en el resumen formando un todo integrado. En un texto inconsistente las frases aparecen inconexas, sin llegar a formar una unidad. El nivel de tolerancia de la inconsistencia del resumen depende de la aplicación a la que esté destinado. El nivel de consistencia depende, además, de los requisitos de formato de la salida. Una organización de texto continuo requiere un mayor nivel de consistencia que una organización en forma de esquema.

La redundancia: Dos oraciones son redundantes si enuncian la misma información o expresan ideas similares.

1.3 Técnicas de extracción

La *extracción* es la aproximación más fácil al problema de la confección automática de un resumen, puesto que no es necesario generar nuevo texto. El problema se reduce a la identificación de los elementos significativos del texto fuente, habitualmente oraciones, y a la selección de los mismos. Las técnicas basadas en extracción analizan y comparan los elementos textuales entre documentos a un nivel morfológico (Trata de cómo las palabras se construyen a partir de los morfemas).

Características

Los sistemas basados en extracción siguen una arquitectura estructurada en dos procesos: análisis y síntesis (HAHN and MANI 2000). En la fase de análisis se procesan cada una de las oraciones del texto fuente y se mide su relevancia. Para ello se utiliza una función de peso que otorga un valor numérico a cada oración. Las métricas que se suelen utilizar en la función de peso pueden clasificarse en:

Posicionales: si tienen en cuenta la posición que ocupa la oración.

Lingüísticas: si buscan ciertos patrones de expresiones indicativas

Estadísticas: si incluyen frecuencias de aparición de ciertas palabras.

En la fase de síntesis se extraen las oraciones con mejor puntuación de acuerdo al ratio de compresión deseado.

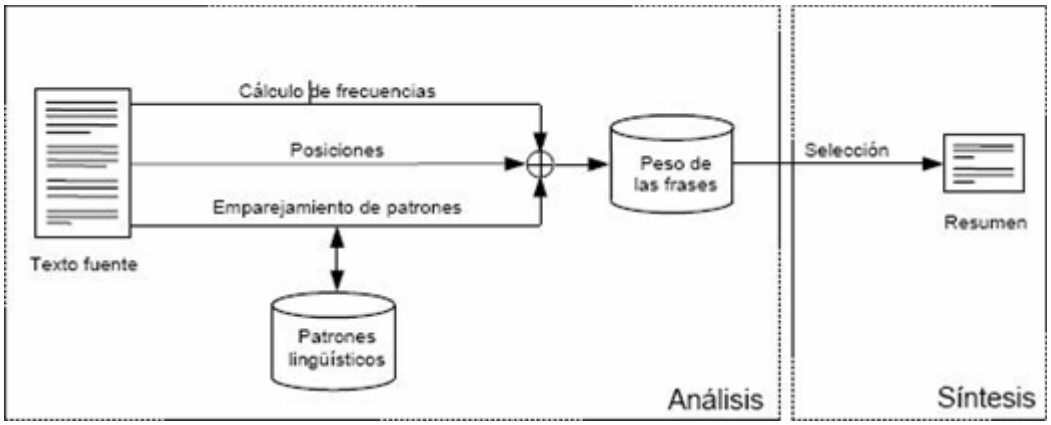

Figura 1-1 Arquitectura de un sistema de generación de resúmenes que utiliza técnicas de extracción. Este esquema es una adaptación del propuesto en (HAHN and MANI 2000)

En correspondencia con lo anterior, la confección de un resumen se transforma en una ordenación de las oraciones del documento fuente de acuerdo a su relevancia. Es decir, se trata de definir una función de peso que, dada una tasa de compresión, consiga maximizar el valor de *S* en la Expresión (1-1).

En (EDMUNDSON 1969) se propone una función de peso a partir de una combinación lineal de diferentes métricas. De esta forma, el peso de una frase (oración) *f* se calcula usando la siguiente expresión:

$$peso(f) = \sum \alpha_i^* M_i(f)$$

Expresión 1-2 Calcula el peso de una oración

Esto es, el valor obtenido al aplicar cada una de las métricas a una frase *f* se pondera mediante un peso α_i

1.4 Proceso de resumen.

De acuerdo a (LIN and HOVY 1997) existen 3 pasos para realizar el resumen de un texto: Identificación del tema, interpretación y generación.

1.4.1 Identificación del tema

El objetivo de este paso es determinar sólo los temas más importantes (centrales) en el texto. La identificación del tema puede ser conseguida por varias técnicas, incluyendo métodos basados en posición, frases indicativas, significación de conceptos y en la frecuencia de la palabra. En algunos géneros del texto, ciertas posiciones de un texto tales como el título, la primera oración en una frase, etc. tienden a llevar temas importantes. Las frases indicativas tales como `resumen', `el mejor', `en conclusión', `el más importante', `este artículo', `este documento', etc. pueden ser buenos indicadores del contenido importante. Las palabras que son más frecuentes en un texto (frecuencia de la palabra) indican importancia del contenido, a menos que sean palabras de función tales como un determinador (adjetivo o modificador que limita un sustantivo) y preposiciones. Los temas son identificados contando conceptos en vez de las palabras (Frecuencia del concepto).

1.4.2 Interpretación

Para los resúmenes extractos, los temas centrales identificados en el paso anterior se remiten al paso siguiente para la transformación posterior. Para los resúmenes abstractos sin embargo, se realiza un proceso de la interpretación. Este proceso incluye temas relacionados con la combinación o fusión, en los más generales, quitando redundancias, etc.

1.4.3 Generación

El tercer paso en el proceso del resumen según (LIN and HOVY 1997) es la generación de la salida final (resumen). Este paso incluye una gama de varios métodos de generación para las palabras o de las frases que imprime una combinación sofisticada de la frase y la generación de oraciones. Los siguientes métodos pueden ser utilizados:

Extracción: Los términos o las oraciones seleccionados en el primer paso del resumen se imprimen en la salida.

Listas tópicos: Las listas de las palabras claves más frecuentes o de los conceptos interpretados se imprimen en la salida.

Concatenación de la frase: Dos o más frases se combinan juntas.

Generación de la oración: Un generador de la oración produce nuevas oraciones. La entrada al generador es una lista de los conceptos y de sus tópicos relacionados.

1.5 Métodos para la construcción de extractos.

Existen diferentes métodos que han sido propuestos para resumir textos. (LUHN 1958) primeramente utilizó reglas basadas en la frecuencia de las palabras para identificar oraciones para resumir, fundamentadas en la intuición de que la palabra con mejor frecuencia representaba el concepto más importante del texto. (EDMUNDSON 1969) incorporó nuevos rasgos tales como frases indicativas (cue phrases), título/palabras de encabezamiento (title/heading words) y localización de la oración (sentence location) dentro del proceso de resumen.

Método basado en el título:

Este método se basa en la hipótesis de que las palabras que aparecen en los títulos, subtítulos y encabezamientos de los documentos pueden ser buenas indicadoras del contenido de los mismos. Un autor concibe el título para definir el tema del documento. El método del título compila, para cada documento, un glosario del título que consiste en todas las palabras no nulas (sustantivos, verbos, adverbios) del título, subtítulo, y los encabezamientos del documento. A las palabras en el glosario del título se le asignan pesos positivos. El peso final para cada oración es la suma de los pesos de sus palabras contenidas en el titulo. A las palabras contenidas en el título se le asignan un peso mayor que a las palabras contenidas en los subtítulos (EDMUNDSON 1969).

Método basado en la localización:

El método de la localización se basa en las siguientes hipótesis: (1) las oraciones que aparecen bajo encabezamientos son positivamente relevantes; y (2)

las oraciones que aparecen en los primeros y últimos párrafos del documento, especialmente las que constituyen la primera y última oración de cada uno. Este método utiliza el diccionario de encabezamientos que almacena la selección de palabras del corpus que aparecen en los encabezamientos, ejemplo: “Introducción”, “Objetivo”, “Conclusiones.” El método asigna pesos positivos a las palabras que se encuentran en el diccionario y además asigna pesos positivos a las oraciones según su posición ordinal en el texto. El peso final de la localización para cada oración es la suma de los pesos de las palabras contenidas en el encabezamiento y de su peso ordinal.

Método basado en las Palabras Clave:

Se basa en la hipótesis de que las palabras de un alto grado de frecuencia son positivamente relevantes y se emplea la frecuencia de los términos para seleccionar las palabras claves. El peso final de la oración es la suma de los pesos de las palabras que la constituyen (EDMUNDSON 1969; LÓPEZ, MANUEL J MAÑA 2003).

En cuanto a la elección de las palabras clave, hay diferencias en el tipo de peso que utilizan: tf , *frecuencia del término* (EDMUNDSON 1969), y $tf-idf$, el producto entre la *frecuencia del término* y la *inversa de la frecuencia del término en la colección* (TEUFEL and MOENS 1997).

Método basado en las Expresiones o palabras indicadoras:

Este método se basa en la hipótesis que el grado de relevancia de una oración es afectada por la presencia de palabras pragmáticas tales como “significativo,” “imposible”; utiliza el diccionario en el cual están depositadas las expresiones o palabras indicadoras seleccionadas de la recopilación. El diccionario que contiene las palabras pragmáticas abarca tres subdiccionarios:

Bonus: constituido por una lista de palabras que son positivamente relevantes, tales como, comparativos, superlativos y adverbios de conclusión.

Stigma: constituido por una lista de palabras que son negativamente relevantes, tales como, anáforas o expresiones que indican especulación o tienen un carácter evasivo.

Null: constituido por las palabras nulas (artículos, preposiciones, conjunciones, pronombres) que son inaplicables.

La aparición de una palabra perteneciente a las listas *bonus* y *stigma* en una frase incrementa o decrementa, respectivamente, la puntuación de dicha frase. El peso final de las expresiones o palabras indicadoras para cada oración es la suma de los pesos de las expresiones o palabras indicadoras que la constituyen (EDMUNDSON 1969; LÓPEZ, MANUEL J MAÑA 2003).

Problemas importantes para crear un extracto

1. Seleccionar las oraciones más importantes.
2. Generar resúmenes coherentes.
3. Eliminar la información repetida en el resumen (redundancia).

Se han hecho muchas investigaciones sobre técnicas para identificar las oraciones más importantes que resumen un documento del texto. Los métodos de extracción de la oración para el resumen, trabajan normalmente puntuando cada oración como candidata para ser parte del resumen, y después seleccionando el subconjunto de oraciones con la puntuación más alta.

Parámetros que aumentan a menudo la puntuación de una oración para la inclusión en el resumen:

Baseline: Cada oración puntúa según su posición en el texto. Por ejemplo en un artículo del periódico, la primera oración consigue la puntuación más alta mientras que la última oración consigue la puntuación más baja.

Título: Las palabras en el título y en las oraciones siguientes son importantes y consiguen una puntuación alta.

Frecuencia de la palabra (WF): Las palabras claves (una palabra clave es una palabra a la que se le puede asignar un significado independiente, ejemplo: los

sustantivos, los adjetivos y los verbos) que son frecuentes en el texto son más importantes que las menos frecuentes. Las oraciones con palabras claves que son las más usadas frecuentemente en el documento generalmente representan los temas del documento.

Frases indicativas: Las oraciones que contienen frases claves tales como “este informe”.

Position Score: La suposición de que ciertos géneros ponen oraciones importantes en posiciones fijas. Por ejemplo, los artículos periodísticos tienen los términos más importantes en los primeros cuatro párrafos mientras que los documentos técnicos tienen las oraciones más importantes en la sección de la conclusión.

Query Signature: Los usuarios tienen a menudo un asunto particular en mente cuando solicitan resúmenes. La pregunta del usuario afecta el resumen pues obligará al texto extraído a contener estas palabras. La puntuación normalizada se le da a las oraciones dependiendo del número de las palabras de la pregunta que contienen.

Longitud de la oración: La puntuación asignada a una oración refleja el número de palabras en la oración, normalizado por la longitud de la oración más larga del texto.

Nombre propio: Los nombres propios, tales como los nombres de personas y de lugares, son a menudo centrales en informes de noticias y las oraciones que los contienen puntúan más alto.

Conectividad léxica media: El número de términos compartidos con otras oraciones. Se asume que una oración que comparte más términos con otras oraciones es más importante.

Datos numéricos: Las oraciones que contienen datos numéricos puntúan más alto que unos sin valores numéricos.

Pronombre: Las oraciones que incluyen un pronombre (conectividad de reflejo de la co-referencia) puntúan más alto.

Días laborables y meses: Las oraciones que incluyen los días de la semana y de los meses puntúan alto.

Cita: Las oraciones que contenían citas pueden ser importantes para ciertas preguntas del usuario.

Primera oración: La primera oración de cada párrafo es la oración más importante.

Los métodos de extracción de la oración mencionados arriba son útiles, pero estos por si solos no pueden producir extractos de alta calidad. Los que utilizan técnicas a nivel de palabras como la, se han criticado en varios aspectos tales como:

Sinonimia: un concepto se puede expresar por diversas palabras. Por ejemplo el ciclo y la bicicleta refieren a la misma clase de vehículo.

Polisemia: una palabra o concepto puede tener varios significados. Por ejemplo, ciclo podía significar el ciclo vital o la bicicleta.

Frases: una frase puede tener un significado diferente al de las palabras en ella contenida.

1.6 Resumen extracto monodocumento

La forma más conocida para construir resúmenes extractos de documentos simples es teniendo un programa selector de fragmentos relevantes desde el documento y luego combinarlos en un extracto (JACKSON and MOULINIER 2002).

Técnicas para construir resúmenes extractos de un único documento.

Resumen por selección de oraciones: La generación de un resumen puede reducirse a la selección de las oraciones más relevantes de un documento, es un problema de clasificación de las oraciones en relevantes o no para formar parte del

extracto (JACKSON and MOULINIER 2002). Esta técnica tiene sus ventajas y desventajas en dependencia del texto a resumir y de la longitud del resumen deseado, por ejemplo cuando se desea resumir noticias o textos no extremadamente largos, es ventajosa, sin embargo los resúmenes resultantes son desunidos y no se leen bien.

Resumen por selección de párrafos: La utilización de componentes básicos más grandes puede contribuir a la obtención de un resumen más coherente. Una vía podría ser seleccionar de un texto los párrafos considerados más relevantes o sea los que describan el texto en su totalidad. Esta técnica es efectiva cuando los documentos están estructurados de forma tal que en unos de sus primeros párrafos o el primero describen el contenido de todo el documento o cuando el resumen solicitado es relativamente largo. A diferencia de la selección de oraciones en la práctica esta técnica no es muy usada (GARCÍA 2005).

Resumen basado en discurso: Modela inicialmente la estructura del documento que será resumido, es decir, determina la forma típica de discurso del documento a resumir (MOENS 2000). Para ello, es necesario primeramente dividir el documento en unidades de discurso coherentes. Los bloques de textos obtenidos deben reflejar los subtópicos contenidos en el texto (YAROWSKY 1992). Esta técnica es de gran utilidad debido a las variaciones de estructura que tienen los documentos y a que es muy ventajoso identificar inicialmente la presencia o ausencia de segmentos claves para la realización del resumen a partir de las unidades básicas detectadas. Generalmente funciona correctamente para un tipo particular de documentos y utilizándolo en un contexto específico.

Resumen basado en co-referencia: Las asociaciones que existen entre términos que tienen co-referencia pueden ser usadas para clasificar y seleccionar oraciones a incorporar en el resumen. Esta técnica extrae el contenido relevante de un documento partiendo de las preguntas de los usuarios (consultas). Genera resúmenes que son casi tan efectivos como el texto completo, ayudando al usuario a determinar la relevancia del documento (BALDWIN and MORTON 1998). Una desventaja es que se requiere realizar un preprocesamiento más costoso del documento.

1.7 Resumen extracto multidocumento

Los *resúmenes multidocumento* requieren la existencia de una colección de documentos que guarden cierta relación semántica. Para la generación del resumen debería identificarse la información común que comparte el conjunto de documentos, de forma que se presente de manera sintética, eliminando las redundancias. Este tipo de técnica es importante para la reducción de la sobrecarga de información. Sin embargo, también es elemental que se descubran y subrayen los aspectos únicos que pueda aportar cada documento. Es de esperar que un grupo de documentos que aborde un determinado tema cubra una gran variedad de cuestiones relacionadas con el mismo.

Evidentemente, cuanto mayor sea la similitud de los documentos más fácil será descubrir esas similitudes y confeccionar el resumen. Por otra parte, cuando el grupo de documentos sobre el que se debe generar el resumen contiene una gran cantidad de documentos, un agrupamiento previo puede ayudar a identificar grupos más homogéneos sobre los que realizar los resúmenes.

La generación de un resumen a partir de varias fuentes requiere tanto la identificación de las similitudes como de las diferencias que se producen en la colección de textos. Uno de los objetivos de la identificación de este tipo de relaciones entre los textos es evitar la redundancia en el resumen. La redundancia entre elementos puede caracterizarse en función del grado de solapamiento que se produce entre los contenidos informativos aportados por cada elemento.

De esta forma, si el contenido informativo de una frase a, expresado como $i(a)$, esta incluido dentro de otra b, es decir, $i(a) \subset i(b)$ entonces b contiene más información de la que aparece en a y puede considerarse mejor candidata para aparecer en el resumen. En este caso la presencia de a en el resumen podría considerarse redundante. Analizando el siguiente ejemplo:

(1) El central azucarero Dos Ríos, del municipio Palma Soriano de la provincia Santiago de Cuba, ha sobre cumplido el plan de producción azucarera.

(2) A más de 500 t de azúcar asciende el sobre cumplimiento del plan de azúcar para esta temporada 2005-2006 del central azucarero Dos Ríos, del municipio Palma Soriano de la provincia Santiago de Cuba, informaron fuentes periodísticas cubanas.

Se dice que las frases son equivalentes desde el punto de vista informativo si una puede sustituir a la otra sin perdida crucial de información. Dicho de otra manera, cuando se cumple que $i(a) \subset i(b)$ e $i(b) \subset i(a)$.

Índice de redundancia entre frases basado en el solapamiento de vocabulario que se produce entre las mismas:

$$R = \frac{2 * número\_palabras\_comunes}{número\_palabras\_frase1 + número\_palabras\_frase2}$$

Expresión 1-3 Calcula el índice de redundancia entre frases

Cuando el índice *R* es igual a 1 las frases son idénticas, mientras que un valor de 0 indica que no hay ninguna palabra en común.

Técnicas para construir resúmenes extractos de múltiples documentos.

Resumir secciones comunes de los documentos: Encontrar las partes importantes y relevantes, comunes en la colección de documentos, es decir, su intersección y utilizarla como un resumen (GARCÍA 2005).

*Resumir secciones comunes y secciones únicas de los documentos***:** Encontrar las partes importantes y relevantes, que tiene en común la colección de documentos y las partes relevantes que son únicas y emplearlas para la construcción del resumen.

Resumir el documento centro (más representativo): Crear el resumen de un único documento a partir del documento centro o más representativo de la colección. Esto expresado en otras palabras es utilizar el documento más relevante de la colección de documentos y hacer a partir de el un resumen monodocumento que represente al resto de los documentos.

Resumir el documento centro (más representativo) más el resto de los documentos de la colección: Crear el resumen de un único documento a partir del documento centro o más representativo de la colección y agregar a este resumen alguna representación del resto de los documentos para proporcionar un resumen que cubra todos los documentos de la colección.

Resumir el documento más reciente más el resto de los documentos de la colección: Esta técnica esta basada en la creación de un resumen del documento que tiene información más reciente y adicionar alguna representación del resto de los documentos para proveer un cubrimiento de la colección de documentos.

Resumir secciones comunes y secciones únicas de documentos teniendo en cuenta el factor tiempo: Se basa en encontrar las partes relevantes e importantes que la colección de documentos tiene en común y las partes relevantes que son únicas. Los métodos bajo esta técnica deben tener en cuenta la secuencia de tiempo de la información extraída en la generación del resumen.

1.8 Algoritmos de construcción de extractos.

Los algoritmos de construcción automática de resúmenes se pueden clasificar en cuanto a:

Los niveles de análisis lingüísticos que emplean: Algoritmos de estrategia poco profunda. Los niveles de análisis lingüísticos pueden ser, por orden de complejidad de menor a mayor: El nivel morfológico, el sintáctico, el semántico y el nivel de discurso. Los algoritmos de estrategia poca profunda en general, no analizan el texto fuente más allá del nivel sintáctico y los elementos más complejos que tienen en cuenta son las sentencias, aunque si operan sobre palabras, estas pueden ser analizadas al nivel semántico. Producen extractos y son muy robustos (ANAYA *et al.* 2006) (ver Anexo A).

Los tipos de elementos del texto sobre los que operan: Algoritmos de estrategia profunda (LIN 1999). Los algoritmos de estrategia profunda realizan el análisis al menos al nivel semántico y los elementos del texto sobre los que operan

no son menos complejos que las cláusulas. Generan abstractos y se aplican a fuentes de un dominio específico (ANAYA *et al.* 2006).

La mayoría de los algoritmos existentes de construcción de extractos son de estrategia poco profunda, y seleccionan elementos de un mismo tipo para componer el extracto, casi siempre sentencias. Las sentencias son elementos lingüísticos que expresan proposiciones o ideas semánticamente completas.

En los algoritmos de construcción de resúmenes extractos, la selección de los elementos de un texto se realiza mediante la clasificación, donde los elementos se clasifican en pertenecientes o no al extracto. Esta clasificación se realiza teniendo en cuenta algunos rasgos de los elementos del texto, que pueden ser: lingüísticos, estadísticos, comunicativos o ser rasgos específicos del dominio del texto que se resume. Por esta razón, *los algoritmos de construcción de extractos pueden ser supervisados o no supervisados.*

1.8.1 Algoritmos de construcción de resúmenes supervisados

Estos algoritmos requieren de una colección de entrenamiento formada por pares (texto fuente-resumen de referencia) y aprenden de ella algunos datos que son usados para definir el clasificador. Debido al uso de tal colección de entrenamiento estos algoritmos generan resúmenes de textos de materias específicas. Ejemplos de algoritmos supervisados:

Algoritmo de Edmundson (EDMUNDSON 1969): En este algoritmo cada sentencia del texto fuente se representa a través de un vector de cuatro componentes, que se corresponden con los valores de los rasgos palabras-pista, palabras-clave, palabras-título y localización de la sentencia. A cada sentencia *s* se le asigna una puntuación definida como $W(s) = \alpha C(s) + \beta K(s) + \gamma L(s) + \delta T(s)$, donde: *C(s), K(s), T(s), L(s)* denotan, respectivamente, los valores de los rasgos palabras-pista, palabras-clave, palabras-titulo y localización de *s* y $\alpha,\beta,\gamma,\delta$ son sus pesos asociados. El primer rasgo pondera las sentencias de acuerdo con la frecuencia de aparición de sus palabras en los resúmenes de una colección de entrenamiento. Los rasgos palabras-título y palabras-claves favorecen a las sentencias que presentan palabras del título y palabras muy frecuentes en el documento,

respectivamente. El rasgo localización favorece a las sentencias que se encuentran cerca del comienzo y del final del texto, pues se presume que estas contienen información importante para los resúmenes, por pertenecer a la introducción y a las conclusiones. El algoritmo clasifica a las *m* sentencias de mayor puntuación como pertenecientes al extracto. Este algoritmo es criticado por el carácter lineal de la función de evaluación de las sentencias, sin embargo, es considerado un prototipo entre los algoritmos de construcción de extractos debido a que la inmensa mayoría incluyen versiones de los rasgos usados por Edmundson.

Algoritmo de Kupiec (KUPIEC *et al.* 1995): En este algoritmo las sentencias de un texto fuente se representan mediante un vector de valores $(v_1,\ldots,v_n)$ correspondientes a los rasgos $R_1,\ldots, R_n$. Si se denota por $R_1(s),\ldots R_n(s)$ a los valores de estos rasgos en la sentencia s, la función de evaluación que emplea el algoritmo se define como la probabilidad de que la sentencia s representada por el vector $(v_1,\ldots, vn)$ sea incluida en el extracto, esto es,

$$W(s) = P(s \in E \mid R_1(s) = v_1 \ldots R_n(s) = v_n)$$

Expresión 1-4 Calcula el peso de una sentencia

Aplicando Naïve Bayes, esta probabilidad puede ser calculada mediante la ecuación:

$$W(s) = P(s \in E \mid R_i(s) = v_1 \ldots R_n(s) = v_n) = P(s \in E) \frac{\prod_{i=1}^{n} P(R_i(s) = v_i \mid s \in E)}{\prod_{i=1}^{n} P(R_i(s) = v_i)}$$

Expresión 1-5 Calcula la Probabilidad de que la sentencia s sea incluida en el extracto

Donde $P(S \in E)$ denota la probabilidad a priori de que una sentencia de un texto fuente sea incluida en su extracto, y $\forall K \in \{1,\ldots n\}$ los valores $P(R_k(s) = v_k \mid s \in E)$ y $P(R_k(s) = v_k)$ son constantes que denotan, respectivamente, la probabilidad de que el rasgo R_k de una sentencia que pertenece al extracto sea igual a la constante R_k y la probabilidad de que el rasgo v_k de una sentencia sea igual a v_k. Las probabilidades anteriores deben ser calculadas a partir de una colección de entrenamiento. Los rasgos usados son

todos discretos y tienen en cuenta: si la longitud de la sentencia es mayor que un umbral, la localización de la sentencia es mayor que un umbral, la localización de la sentencia dentro de uno de los diez primeros o diez últimos párrafos del texto, y la presencia de frases que indican resumen, de palabras temáticas y de siglas frecuentes del texto fuente en la sentencia. Al igual que el algoritmo de Edmundson, este clasifica a las m sentencias de mayor puntuación como pertenecientes al extracto y constituye a su vez un prototipo de algoritmo para la construcción de resúmenes extractos (ANAYA *et al.* 2006).

Existen otros algoritmos supervisados como los descritos por (MANI and BLOEDORN 1998) y (LIN 1999), donde se consideran grupos más refinados de rasgos y la generación de resúmenes extractos enfocados a un tópico. Ambos usan un árbol de decisión como clasificador.

En los algoritmos supervisados de construcción de extractos las operaciones de la fase de análisis (análisis lexicográfico del texto fuente, segmentación del mismo en sentencias y la obtención de la representación de estas) y el computo del puntaje asociado a las sentencias se pueden efectuar en un tiempo en el orden de O (|X|) donde |X| es la longitud del texto fuente. La operación de selección de las m sentencias de mayor puntuación es llevada a cabo en un tiempo que es O(|X|log|X|) al considerar que m se obtiene a partir de la razón de compresión y de la longitud del texto fuente. Por tanto, la complejidad temporal de estos algoritmos es O(|X|log|X|) (ANAYA *et al.* 2006).

La colección de entrenamiento con que se cuenta está compuesta por pares (texto fuente, abstracto) en lugar de pares (texto fuente, extracto). Esto hace necesario disponer de algoritmos que generen un extracto a partir de un abstracto. (LIN 1999) describió dos estrategias bastante generales para la solución de este problema, las cuales se diferencian en la forma de evaluar a las sentencias del texto fuente para construir el resumen.

La estrategia de emparejamiento combinado: Se le asigna a cada sentencia del texto fuente una puntuación basada en su semejanza con el abstracto completo tomado como una sentencia. La función de semejanza usada por (MANI and BLOEDORN 1998), basada en la medida del coseno, fue:

$$N_1 + \frac{\sum_{i=1}^{N_2} i_{S1}\, i_{S2}}{\sqrt{\sum_{i=1}^{N_2} i_{S1}^2 \sum_{i=1}^{N_2} i_{s2}^2}}$$

Expresión 1-6 Función de semejanza

Donde:

i_{s1} , i_{s2} :Valores respectivos de los $tf - idf$ de la palabra *i* en las sentencias s_1 y s_2 .

N_2: Numero total de palabras de s_1 y s_2 .

Luego, las sentencias se ordenan de mayor a menor, de acuerdo con su puntuación y se seleccionan para formar parte del resumen extracto las primeras que constituyen no mas del $100\,C\%$ del total, donde C es la razón de compresión deseada del extracto. Los algoritmos de emparejamiento combinado pueden ser usados también para obtener un resumen de un texto fuente enfocado a una consulta, considerando a la misma en lugar del abstracto.

La de emparejamiento individual: Se le asigna a cada sentencia del texto fuente como puntuación, el máximo valor obtenido de la comparación de esta con cada una de las sentencias del abstracto. Luego, se procede con la estrategia de emparejamiento combinado para seleccionar las sentencias del resumen.

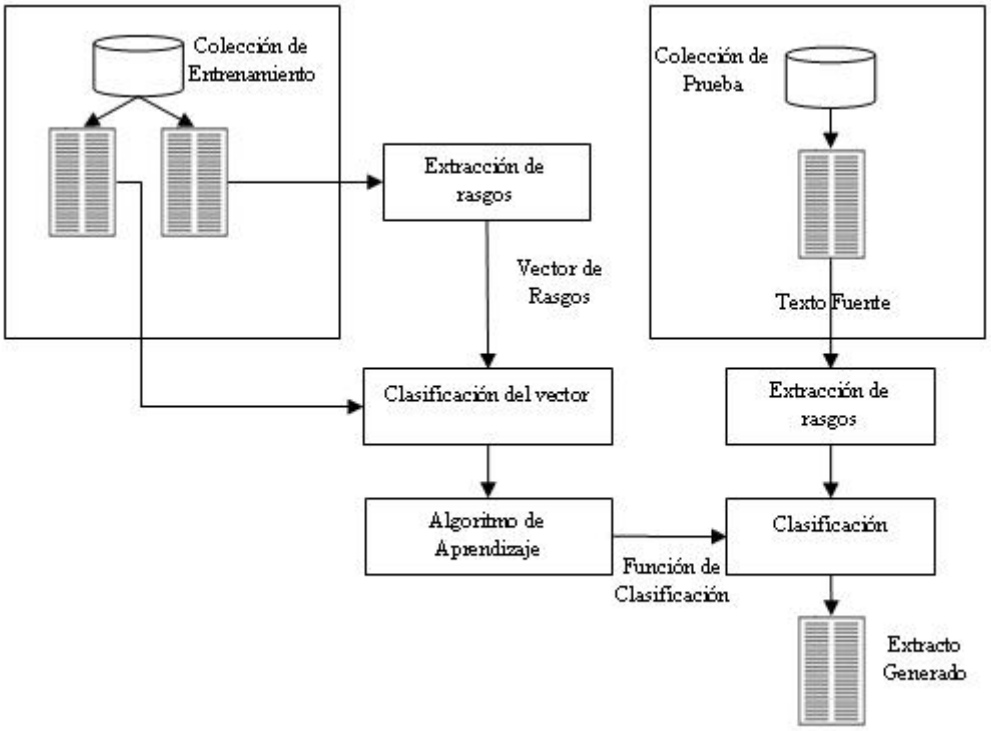

Figura 1-2 Esquema general de los algoritmos basados en una colección de entrenamiento.

Los algoritmos de emparejamiento individual tienen una complejidad temporal cuadrática con respecto a la longitud del texto fuente, mientras que los de emparejamiento combinado logran una complejidad subcuadrática.

Existen otras estrategias, como el algoritmo ávido de Marcu (MARCU 1999), que no son ni de emparejamiento combinado ni individual. En esta se genera un extracto partiendo de un texto igual al texto fuente, al que se le van eliminado iterativamente las sentencias que menos se parecen al abstracto, mientras no decrezca la semejanza entre este y el abstracto.

1.8.2 Algoritmos no supervisados

Los algoritmos no supervisados siguen básicamente dos esquemas. El primero consiste en ponderar cada elemento del texto individualmente, considerando propiedades intrínsecas de estos en el texto (estadísticas y lingüísticas), para luego clasificar los elementos de mayor peso en elementos del extracto, de manera similar al algoritmo de Edmundson. Un algoritmo que proporciona un marco bastante general para los algoritmos no supervisados del primer esquema se puede encontrar en el trabajo de (GOLDSTEIN *et al.* 1999). Los

algoritmos que siguen el segundo esquema usan el nivel de discurso del texto para construir, a partir de los elementos y sus relaciones, un grafo que luego es usado para clasificar y extraer los elementos que formaran parte del extracto. Los algoritmos no supervisados que siguen el primer esquema son muy similares al algoritmo de Edmundson (su diferencia radica en la no utilización de una colección de entrenamiento).

La coherencia de un texto está dada por las relaciones de discurso que se establecen entre sus cláusulas y sentencias. Entre estas relaciones están la elaboración, ejemplificación y explicación. La estrategia de extracción basada en la coherencia de un texto consiste en que los principales núcleos del discurso son usados para la composición del extracto (ANAYA *et al.* 2006).

Algoritmo de Barzilay basado en la cohesión léxica (BARZILAY and ELHADAD 1999): Este algoritmo para cadena léxica fuerte del texto fuente, selecciona la primera sentencia que contiene un miembro representativo de la cadena y la clasifica como perteneciente al extracto. Para el cálculo de las cadenas léxicas, se tienen en cuenta las relaciones léxicas existentes entre las palabras, auxiliándose de la base de datos léxica WordNet la fortaleza de una cadena léxica toma en consideración su longitud y homogeneidad. Un miembro de una cadena es representativo si su frecuencia de aparición en la cadena supera la frecuencia promedio.

Algoritmo de Nomoto y Matsumoto (NOMOTO and MATSUMOTO 2001): Este algoritmo define un extracto como un conjunto de sentencias extraídas de un texto fuente que cubren su esencia; y basa la construcción de tal conjunto en dos importantes propiedades del texto: *la diversidad* y *la redundancia* de los conceptos que en él ocurran. Para ello particiona el conjunto de las sentencias del texto fuente, usando un algoritmo de agrupamiento, de manera tal que cada subconjunto esté compuesto por sentencias que representan un tópico del texto. Luego define el extracto como el conjunto formado por la sentencia más importante de cada subconjunto. La elección de tal sentencia se realiza teniendo en cuenta la frecuencia de sus términos en el documento.

Los algoritmos que se basan en *la cohesión* del texto para la construcción de extractos, constituyen un grafo donde los nodos representan palabras, frases o incluso grupos de estas representados por las sentencias donde ocurren, y las aristas o los arcos (según sea el caso) representan enlaces de cohesión entre los elementos asociados a los nodos. El extracto se construye teniendo en cuenta que, mientras mayor es el grado de los nodos, más prominente es la información que este contiene. Debido al empleo de este grafo, la complejidad temporal de estos algoritmos es cuadrática con respecto a la longitud del texto fuente (ANAYA *et al.* 2006).

Los algoritmos que se basan en *la coherencia* de un texto no son muy abundantes en la literatura, debido a que la obtención de la estructura de discurso de un texto no es una tarea fácil y continúa siendo un desafío, y a que estos no garantizan la obtención de extractos coherentes. Estos algoritmos construyen un árbol etiquetado que representa la estructura de discurso del texto a resumir. Sus hojas son elementos del texto y sus nodos interiores representan la relación existente entre sus nodos hijos y tienen asociados los elementos del texto que constituyen el núcleo de esa relación. Luego, construyen un orden parcial entre los elementos del texto, teniendo en cuenta la profundidad del nodo del que constituyen su núcleo. Por ultimo, se clasifican los primeros m elementos del orden como pertenecientes al extracto. Algoritmos de este tipo se pueden encontrar en (MARCU 1997).

Otros Algoritmos:

TextRank: Recibe como entrada el texto y comienza por la construcción de un grafo que lo representa, e interconecta palabras u otras entidades del texto con relaciones significativas. Para la tarea de la extracción de la oración, la meta es darle una puntuación a las oraciones, y para esto un vértice es agregado al grafo para cada oración del texto. Las oraciones con la puntuación más alta en el grafo son seleccionadas para la inclusión en el extracto. Un aspecto importante de TextRank es que no requiere conocimiento lingüístico profundo, ni dominio o recopilaciones anotadas específicas a una lengua, que lo hace altamente portable a otros dominios, géneros, o idiomas (MIHALCEA and TARAU 2004).

Fractal Summarization: Propone generar un resumen basado en la estructura del documento. La sumarización fractal supera la sumarización tradicional, según (Fractal Summarization: Summarization Based on Fractal Theory) (YANG and WANG 2003)

Shortpath: Para documentos largos el tiempo de procesamiento de este algoritmo puede ser muy extenso, aunque las implementaciones actuales no están completamente optimizadas para la velocidad. Algunos problemas posibles con el método son que él podía mantener la redundancia dentro del texto original seleccionando oraciones similares. Podría también faltar el punto principal del texto, aunque el peso de “oraciones importantes” ayuda a evitar esto. En la práctica para trabajar absolutamente bien (SJOBERGH and ARAKI).

Fukumoto: asigna a cada palabra, oración o párrafo del texto un peso que dependerá de su distribución en el propio documento y en un contexto más amplio. El resumen se construirá seleccionando en primer lugar aquellas oraciones o párrafos que estén incluidos en un mayor número de los grupos resultantes del proceso de clasificación. La principal ventaja de este método radica en la posibilidad de ajustar los resúmenes a distintos contextos pero, al mismo tiempo, es su principal inconveniente al requerir un corpus para extraer resúmenes (FUKUMOTO 2003).

Urelevance Measure: es un algoritmo genérico, asigna un coeficiente de relevancia a cada oración y selecciona las oraciones de mayor coeficiente para constituir el resumen. Su principal desventaja es que el resumen producido puede contener incoherencia entre las oraciones (BELLAACHIA and MAHAJAN 2005).

1.9 Métodos de evaluación de los resúmenes.

Un resumen debe ser coherente, conciso, de fácil lectura y comprensión, y ofrecer información relevante al usuario de ahí la importancia practica que tiene el mismo para la sociedad, lo cual hace que la evaluación de estos resúmenes sea de gran utilidad. Esta evaluación tiene como objetivo determinar cuan adecuados y útiles son estos con relación al texto original. Existen 2 propiedades de los

resúmenes que deben ser medidas cuando se evalúa: La razón de compresión (cuánto más pequeño es el resumen que el texto original);

$$CR = \frac{longitud\_del\_resúmen}{longitud\_del\_texto\_original}$$

Expresión 1-7 Razón de Compresión

Y la razón de retensión (cuánta información es retenida);

$$RR = \frac{Información\_en\_el\_resúmen}{Información\_en\_el\_documento}$$

Expresión 1-8 Razón de Retención

La necesidad de evaluación ha estado presente desde los primeros trabajos presentados y esto ha dado lugar a que numerosos investigadores de varios países se preocupen acerca de esta problemática y demuestren su esfuerzo en solucionar el problema con soluciones colectivas tales como SUMMAC (fue la primera iniciativa de evaluación independiente y a gran escala de sistemas de generación automática de resúmenes) o la discusión y diseño de planes y estrategias de evaluación coordinadas que, posteriormente, se han plasmado en las conferencias DUC (tienen como objetivo la evaluación independiente de sistemas de generación de resúmenes).

Actualmente la comunidad científica no ha determinado cual es el método de evaluación más óptimo. (MANI *et al.* 2001) explica causas que producen este hecho tales como:

- Es difícil llegar a un acuerdo sobre la idoneidad de un resumen generado automáticamente.
- La necesidad de utilizar personas para juzgar los resúmenes encarece el proceso de evaluación y hace que sea difícil de repetir.
- Debería ser posible evaluar resúmenes de distintas tasas de compresión.

- Aunque, en determinados contextos, la legibilidad puede ser un factor crucial, no garantiza por ella misma la calidad de un resumen.
- Como hemos mencionado en varias ocasiones, el resumen puede generarse en función de las necesidades del lector y del uso al que esté destinado. Por tanto, estos factores también deben tenerse en cuenta. De nuevo, el resultado es una evaluación más compleja.

1.9.1 Clasificación de los métodos de evaluación

Los métodos de evaluación pueden clasificarse en *intrínsecos* y *extrínsecos*. Los métodos *intrínsecos* se caracterizan por analizar directamente la calidad del resumen, utilizando para ello algún criterio que permita medir su adecuación o habilidad o comparándolo con un resumen "modelo" pre-existente o creado a tal efecto por seres humanos. Los métodos *extrínsecos* juzgan la calidad de un resumen en función de su utilidad para realizar alguna otra tarea, evalúan la "calidad" de los resúmenes de manera indirecta a través de su influencia en la ejecución de una o más tareas

1.9.1.1 Método intrínseco o normativo

En este método los usuarios juzgan la calidad del resumen a través de su análisis directo, utilizando para ello algún criterio que permita medir su adecuación o fiabilidad. Los usuarios califican la naturalidad: la capacidad del resumen para cubrir las ideas claves, o bien la similitud con los resúmenes escritos por humanos (expertos en el área o no) (ZAMBRANO 2002).

Clasificación general de los métodos intrínsecos:

1. Grupo I: Los que evalúan la calidad (como obra textual).
2. Grupo II: Los que evalúan el contenido informativo de los resúmenes.

La intervención de jueces humanos en este tipo de evaluación es importante y necesaria, incluso podría puntualizarse que es imprescindible. Siendo responsables de tareas de alta envergadura tales como: opinar de la legibilidad de los resúmenes, indicar el grado de relevancia de la información seleccionada o confeccionar resúmenes que servirán de *modelo ideal* con los que medir los

generados automáticamente. Evidentemente, para poder obtener conclusiones relevantes, las evaluaciones deben llevarse a cabo sobre colecciones de cierto tamaño.

Aspectos a tener en cuenta por el método intrínseco

Calidad del resumen: Uno de los aspectos que inciden sobre esta es la legibilidad de los resúmenes. Para realizar la evaluación de este aspecto se le asigna a una n cantidad de jueces la tarea de leer los resúmenes e identificar el número de errores con respecto a estos criterios:

1. Errores en el uso de mayúsculas.
2. Orden incorrecto de las palabras.
3. Falta de concordancia en el número entre sujeto y verbo.
4. Falta de componentes importantes de la frase (por ejemplo: sujeto, verbo principal o complemento directo) que afecten a la claridad de la misma.
5. Fragmentos no relacionados unidos en la misma frase.
6. Omisión o uso incorrecto de artículos.
7. Pronombres con antecedentes incorrectos u omitidos.
8. Sustantivos para los que resulta imposible determinar claramente a quien o que se refieren.
9. Posibilidad de reemplazar sustantivos por pronombres.
10. Conjunciones utilizadas incorrectamente.
11. Información repetida innecesariamente.
12. Orden incorrecto de las frases.

*Relevancia de la información: E*l método más empleado consiste en comparar el resumen generado automáticamente y el confeccionado de forma manual (considerado “ideal”). Puede ser que este calificativo para el resumen manual no sea tan acertado por las causas explicadas anteriormente pero cuando se compara el resumen automático con este resumen de referencia, si contienen información relevante pueden ser considerados buenos. Las métricas tradicionales de IR: *Precision* (porcentaje de oraciones recuperadas que son relevantes), *Recall* (proporción de oraciones relevantes existentes en la colección que se han

recuperado) y *F-Measure* (porcentaje de calidad del resumen) han sido utilizadas con frecuencia para medir la eficacia de los resúmenes automáticos en comparación con otros confeccionados manualmente y tomados como referencia (SALTON and MCGILL 1983).

En (DONAWAY *et al.* 2000) se proponen dos medidas alternativas a la cobertura de frases: la ordenación de frases y la similitud de contenidos. Para el primero de los métodos es necesario que los jueces ordenen las frases del documento en función de su importancia. Entonces, basta con comparar esta ordenación con la proporcionada por el sistema de generación, utilizando alguna medida de correlación. El gran inconveniente de esta técnica estriba en la dificultad que supone para un juez realizar esta tarea. El segundo de los índices consiste en representar ambos textos mediante vectores y calcular su similitud.

Otro método de evaluación, propuesto en (RADEV *et al.* 2000), es el denominado índice de utilidad. La técnica se basa en la hipótesis de que no todas las frases que se seleccionan para formar parte de un resumen tienen la misma importancia. De esta manera, se pide a los jueces que emitan una valoración, entre 1 y 10, de cada una de las frases del texto fuente. De esta forma, se puede confeccionar un resumen de referencia de cualquier longitud, simplemente eligiendo las frases con mayor valoración. El índice de utilidad se calcula dividendo la suma de valoraciones de las frases seleccionadas por el sistema entre la suma de valoraciones de las frases que conforman el resumen de referencia.

La coherencia del resumen: Algunos elementos de un resumen sufren la pérdida del contexto en el que ocurren en la fuente, acarreando problemas de coherencia tales como referencias sin resolver y fisuras en la estructura de discurso. De aquí que un resumen se pueda evaluar según su coherencia, la cual puede medirse teniendo en cuenta la presencia de anáforas sin resolver y la falta de preservación de ambientes estructurados como listas y tablas en su texto. Otra medida de coherencia se basa en un algoritmo de aprendizaje supervisado, que clasifica las sentencias en coherentes o no.

El contenido del resumen: Un resumen puede ser evaluado comparando su contenido con el de un resumen de referencia o con el de su texto fuente. La

comparación del contenido de 2 textos puede realizarse una medida de solapamiento de vocabularios, como el coeficiente de Dice o la medida del coseno. Si en esta evaluación está involucrado un abstracto debe usarse algún tesauro de términos a la hora de representar los textos.

Los n-grama del resumen: Para evaluar la calidad de un resumen, en lugar de considerar las sentencias, se pueden considerar los n-gramas (un n-grama es una secuencia de n palabras consecutivas de un texto).ROUGE-n (LIN and HOVY 1997) es una medida basada en la ocurrencia de los n-gramas que evalúa la relevancia de un resumen r a partir de un conjunto de resúmenes de referencia C y se define como:

$$Rouge\ n = \frac{\sum_{t \in C} \sum_{g \in ngramas\ (t)} min\{cant(g,t), cant(g,r)\}}{\sum_{t \in C} \sum_{g \in ngramas(t)} cant(g,t)}$$

Expresión 1-9

Donde:

$ngramas(p)$: Conjunto de gramas del texto p

$cant(g, p)$: denota la cantidad de veces que aparece el *n-grama g* en *p*.

1.9.1.2 Métodos extrínsecos

Los métodos *extrínsecos* juzgan la calidad de un resumen en función de su utilidad para realizar alguna otra tarea.

En este método los usuarios juzgan la calidad del resumen de acuerdo a la manera como afecta la culminación de alguna otra tarea tales como determinar la relevancia de los tópicos de interés del texto o responder ciertas preguntas relacionadas con el contenido. Evalúa la calidad del resumen respecto a una de estas tareas, con el objetivo de medir su utilidad. Para poder evaluar se necesitan ciertas condiciones de prueba como corpus de textos y resúmenes, sistemas de recuperación de información, conjuntos de consultas y resultados relevantes

evaluados por expertos (ZAMBRANO 2002). Esta forma de evaluación exige casi siempre una activa participación de personas.

Uno de los más sencillos es el de *Lectura de comprensión*. Este método evalúa a un resumen según el porcentaje de respuestas correctas que alcanza una persona en una prueba que le es realizada después de la lectura del resumen. A diferencia de otros métodos extrínsecos, éste puede ser utilizado también para evaluar el contenido informativo de un resumen.

1.10 Conclusiones parciales

Después de realizar un estudio sobre el estado del arte del resumen automático, a partir de las consultas realizadas nacional e internacionalmente y teniendo en cuenta la no existencia de estos algoritmos implementados en Python se decide comenzar el trabajo de una forma general y para esto se seleccionan el Algoritmo basado en el Coeficiente de Relevancia de la Oración y el Algoritmo de Fukumoto que pertenecen al grupo de los no supervisados, específicamente a los que siguen el esquema de ponderación de cada elemento del texto y debido a que realizan un análisis lingüístico hasta el nivel sintáctico son clasificados como algoritmos de estrategia poco profunda .Estos algoritmos producen como resultado resúmenes genéricos ya que éstos proporcionan un sentido global del contenido del texto original o de la colección de documentos. Para la creación automática de los resúmenes de textos se realizarán los tres pasos correspondientes y se utilizará la Técnica de Extracción para su confección.

Capítulo 2 : Descripción e implementación de los algoritmos propuestos para construir la biblioteca. Evaluación. Evaluación de los resultados.

Introducción

Luego de realizar un estudio sobre el estado actual de los resúmenes automáticos, las técnicas y métodos de generación de estos así como de los algoritmos de construcción de extractos. En este capitulo se realiza una descripción sobre el diseño e implementación de la biblioteca en el lenguaje de programación Python a partir de los algoritmos seleccionados.

2.1 Algoritmos seleccionados.

Se seleccionó el algoritmo basado en el coeficiente de relevancia de la oración y el algoritmo Fukumoto para formar parte de la biblioteca de clases. Estos algoritmos producirán resúmenes genéricos e indicativos monodocumento y multidocumento respectivamente. Se clasifican entre los algoritmos de estrategia poco profunda, llevando a cabo un análisis lingüístico al nivel sintáctico, perteneciendo a su vez al grupo de los algoritmos no supervisados específicamente los siguen el esquema de ponderación de cada elemento del texto fuente.

2.1.1 Algoritmo basado en el coeficiente de relevancia de la oración.

El algoritmo basado en el coeficiente de relevancia de la oración parte de la representación en el espacio vectorial del documento.

Descripción:

Este algoritmo recibe como entrada el documento a resumir y el tamaño del resumen deseado, luego divide el documento en oraciones, pondera cada palabra por su tf , crea el vector A_i del peso de frecuencia del término, para cada oración $i \in S$ y el vector *D* del peso de la frecuencia del término para el documento completo, de acuerdo al esquema de peso seleccionado. Para cada oración $i \in S$,

calcula la puntuación relevante entre A_i y D, la cual es el *Inner Product*/*Cosine Similarity/Jaccard Co-efficient*. Selecciona la oración *k* que tenga la mayor puntuación de relevancia y la adiciona al resumen. Borra la oración *k* del conjunto *S*, y elimina todos los términos contenidos en *k* del documento. Re-calcula el peso de la frecuencia del término para el documento completo. Si el número de las oraciones en el resumen alcanza el valor predefinido, termina la operación: en caso contrario vuelve a calcular la puntuación relevante entre A_i y *D*.

Conceptos necesarios:

- Vector de frecuencia del término de paso *i*: $T_i = [t_{1i}, t_{2i}, \ldots . t_{ni}]^T$

Donde

t_{ji} : denota la frecuencia en la cual el término j ocurre en el paso i. Donde paso i podría ser una frase, sentencia, un párrafo del documento o podría ser el mismo documento entero.

- Vector peso de la frecuencia del término: $A_i = [a_{1i}, a_{2i} \ldots . a_{ni}]^T$ está definido como $aji = L(tji) * G(tji)$

Donde

$L(t_{ji})$ es el peso local del término *j* en paso *i*

$G(t_{ji})$ es el peso global del término *j* en el documento completo.

Algoritmo:

Entrada: Documento a resumir, tamaño del resumen deseado

Salida: resumen extracto basado en las oraciones más relevantes del documento.

1. Entrar el texto del documento.

2. Analizar gramaticalmente las oraciones dentro del documento donde cada sentencia termina en un ".", y adicionar cada sentencia como un miembro de una estructura de datos Vector llamada Vector "Sentencia".

3. Para cada sentencia:

- Analizar gramaticalmente los tokens.
- Poner todos los tokens únicos en una estructura de datos Tree Map donde hay una relación uno a uno entre los términos y sus frecuencias *(tf).* Adicionar el Tree map como un miembro del vector frecuencia del termino *(T)* llamado Vector Tree map. Aplicar el esquema de peso local y global a este vector para crear un vector del peso de la frecuencia del término *(A_i).*
- Simultáneamente, crear un Tree Map para el documento entero *(D)* y añadirlo al vector Tree Map.

4. Calcular la puntuación de relevancia por cualquiera de estas técnicas *Inner Product/ Cosine Similarity* y *Jaccard Co-efficient* entre cada miembro de A_i y *D.*
5. Escoger la sentencia con la mejor puntuación de relevancia y añadirlo al vector "resumen".
6. Extraer todos los términos de la sentencia citada anteriormente del documento *D* y A_i reconstruir *D*.
7. Cuando el número de sentencias en el Vector "Resumen" alcance el tamaño del resumen deseado, parar, sino ir al paso 4.

Esquemas de peso:

El algoritmo permite elegir el esquema de peso a utilizar. Lo cual permite obtener resúmenes diferentes para cada esquema. Particularmente hemos elegido para calcular el peso local el peso logarítmico (*Logarithm weight*) y para el peso global la frecuencia inversa del documento (*Inverse document frequency (IDF*)).

Peso Local L (sji):

- No_Weight: $L(S_{ji}) = tf(S_{ji})$

- Peso_Binario: $L(S_{ji}) = 1, si\ tf(S_{ji}) \geq 1,$ $sino\ L(S_{ji}) = 0$
- Peso_Aumentado: $L(S_{ji}) = 0.5 + 0.5*(tf(S_{ji})/tf(max))$ $donde,\ tf\ (max) = max\{tf(1i)\ ,\ tf(2i),..,tf(mi)\ \}$
- Peso_logarítmico: $L(S_{ji}) = log(1+tf_{(ji)})$

Peso Global G (s_{ji}):

- No_Weight: $G(S_{ji}) = 1$
- Frecuencia Inversa del Documento (idf): $G_{(j)} = log(N/n_{(j)})$

Donde, *N* es el número de oraciones en el documento y $n_{(j)}$ es el número de oraciones que contiene el término *j*.

Medidas de relevancia:

Existen diferentes medidas para calcular la puntuación de relevancia de las oraciones para este algoritmo, las cuales de acuerdo a su elección posibilitan obtener resúmenes distintos. Aunque el algoritmo de la oportunidad de generar un resumen utilizando cualquiera de estas medidas, para realizar las primeras pruebas se ha seleccionado el Inner Product, debido a que es actualmente la más utilizada.

Inner Product (IP): $IP(A_i, D) = \sum_{k=1}^{t} (a_{ik} * d_k)$

Cosine Similarity (CosSim): $CosSim(Ai, D) = \dfrac{\sum_{k=1}^{t} (a_{ik} * d_k)}{\sqrt{\sum_{k=1}^{t} a_{ik}^2} * \sqrt{\sum_{k=1}^{t} d_k^2}}$

Jaccard Co-efficient (JacCoeff):

$$JacCoeff(A_i, D) = \frac{\sum_{k=1}^{t} (a_{ik} * d_k)}{(\sqrt{\sum_{k=1}^{t} a_{ik}^2} + \sqrt{\sum_{k=1}^{t} d_k^2} - \sum_{k=1}^{t} (a_{ik} * d_k))}$$

Ventajas:

Este algoritmo al seleccionar las oraciones con mayor puntuación de relevancia asegura que el resumen producido cubra la mayor parte del contenido del documento, además al eliminar la oración seleccionada en cada iteración se asegura que la próxima oración no solape a la anterior y esto contribuye a la generación de un resumen menos redundante.

Desventajas:

Los resúmenes obtenidos como resultado de la ejecución del algoritmo presentan tendencia a ser, en parte, incoherentes; debido a que está constituido por oraciones que aún cuando pueden catalogarse como las más relevantes del texto, no constituyen una secuencia de ideas lógicas necesariamente.

2.1.2 Algoritmo de Fukumoto.

Descripción

El algoritmo requiere de una previa agrupación y clasificación de los documentos, de forma tal que todos los documentos traten de un solo tema.

Este algoritmo recibe como entrada la colección de documentos a resumir y el tamaño del resumen deseado, o sea, el ratio de compresión, luego realiza el resumen a cada documento de la colección, resumiendo el primer documento (más relevante) al porciento especificado anteriormente y el resto de los documentos al porciento aumentado en 10, devolviendo el id del documento y el resumen. Elimina a continuación las partes innecesarias y luego genera el resumen extracto multidocumento.

Algoritmo

La sumarización del texto de un multidocumento se conduce en los procedimientos siguientes:

Entrada: Documentos a resumir, tamaño del resumen deseado.

Salida: Resumen extracto.

1. El primer documento se resume en el ratio de compresión deseado.

2. Los documentos siguientes se resumen en el pociento de resumen requerido aumentado en un 10%.

3. Todos los resúmenes monodocumento son divididos en oraciones y estas se dividen en segmentos (palabras).

4. Los valores de la semejanza entre todas las oraciones se calculan y el mayor valor será la puntuación de esa oración.

5. Eliminar del resumen las oraciones que tenga mayor puntuación de semejanza hasta que se alcance el porciento de resumen deseado.

6. El resto de las oraciones se ordenan por según el orden cronológico que tenían inicialmente en los documentos para formar parte luego del resumen extracto multidocumento.

Valores de Similaridad

El algoritmo calcula el coeficiente de relevancia de las oraciones que forman parte del resumen monodocumento mediante la siguiente formula:

$$SV_i = \frac{número\_palabras\_repetidas\_d_i\_d_j}{número\_palabras\_d_j}$$

Expresión 2-1 Calcula el Valor de Similaridad

Donde:

El *número_palabras_repetidas__d_i_d_j* significa el número de palabras que contienen la oración *i* que están también contenidas en la oración *j* y *número_palablas_dj* representa el número total de palabras que contiene la oración *j*.

Ventajas

La principal ventaja de este método radica en la posibilidad de ajustar los resúmenes a distintos contextos. Mediante la eliminación de las partes innecesarias este algoritmo asegura que no existan oraciones redundantes.

Desventajas

El algoritmo no produce resúmenes del todo coherentes, debido a que representar las oraciones por su orden original en cada documento puede producir cierta falta de unidad entre las oraciones del extracto. Requiere de un corpus para generar el resumen.

2.2 Diseño e implementación de la Biblioteca de Clases.

En esta sección se muestran los pormenores involucrados en el diseño e implementación de la biblioteca de clases para el algoritmo seleccionado. El diseño debe tener flexibilidad, para que en un futuro se puedan implementar otros algoritmos. Se comienza con una breve descripción y justificación de las herramientas computacionales utilizadas. Posteriormente se continúa con las especificaciones técnicas de cada clase.

2.2.1 Metodología y herramientas empleadas en la implementación.

Durante el diseño e implementación de la biblioteca de clases se han utilizado varias herramientas que serán descritas a continuación, las cuales contribuyeron a que el producto tuviera mejor calidad.

Python:

Como requisito este trabajo ha sido implementado en el lenguaje de programación Python, aprovechando la no existencia de los algoritmos en este lenguaje y pensando en su posterior integración con otros productos que cubran ciertos requerimientos.

Es un lenguaje de programación orientado a objetos, permite ejecutar los programas en cualquier sistema operativo y/o arquitectura, es interpretado, lo que ahorra un tiempo considerable en el desarrollo del programa, pues no es necesario compilar ni enlazar. El intérprete se puede utilizar de modo interactivo, lo que facilita experimentar con características del lenguaje, escribir programas desechables o probar funciones durante el desarrollo del programa. Realizar un programa bajo este lenguaje, seguramente costaría entre la mitad o la cuarta parte del tiempo que tardaría en desarrollar el mismo programa en C/ C++ o Java esto hace que sea muy potente.

Tiene eficaces estructuras de datos de alto nivel y una solución de programación orientada a objetos simple pero eficaz. Permite la declaración dinámica de variables, es decir, no es necesaria la declaración de variables, se pueden declarar variables a la vez que le asignamos algún tipo de dato. Dispone también de un gestor de memoria que se encargará de liberar la memoria de objetos no utilizados. Además se puede combinar con otros múltiples lenguajes de programación, como: Java (Jython), C o C++. Cuenta con una amplia biblioteca de módulos que permiten un desarrollo rápido y eficiente, su sencillez también ayuda a que los programas escritos en este lenguaje sean muy sintéticos.

Eclipse:

Se ha utilizado, para la codificación de la biblioteca el ambiente de desarrollo Eclipse fundado originalmente por la IBM y actualmente desarrollado por la fundación Eclipse. Es un ambiente de desarrollo de fuente abierta y orientado principalmente a tecnología Java. El entorno integrado de desarrollo (IDE) de Eclipse emplea módulos (*plug-in*) para proporcionar toda su funcionalidad. Este mecanismo de módulos es una plataforma ligera para componentes de software, adicionalmente le permite a Eclipse extenderse usando otros lenguajes de programación como son C/C++ y Python. Se ha utilizado el *plug-in* Pydev el cual permite desarrollar programas en el lenguaje Python.

Subversion:

Con el objetivo de gestionar los cambios del código de forma segura, es decir, no perder el código y poder mantenerlo en caso de ocurrir algún accidente en la maquina se decidió utilizar el sistema de control de versiones Subversion, el cual es actualmente uno de los mas utilizados a nivel mundial. Para lograr una relación entre el ambiente de desarrollo y el control de versiones se empleó el plug-in Subclipse, el cual trabaja como cliente del Subversion desde el eclipse, permitiendo la interacción entre ambos.

Zope:

Constituye otra de las herramientas computacionales que ha sido usada en la implementación. Es un servidor de aplicaciones totalmente orientado a objetos escrito en el lenguaje de programación Python. Está publicado bajo los términos de la licencia Zope Public License (ZPL), una licencia de software libre. Se utiliza para reutilizar algunas de sus funcionalidades y hacer el trabajo más fácil y rápido. De Zope se utiliza hasta ahora el módulo *Lexicon* el cual tiene implementado una clase que se usa para eliminar las palabras de parada y el módulo *StopDict*, el cual contiene el listado de las palabras de parada en inglés y se le agrega las palabras de parada del idioma español.

Rational Rose:

Para desarrollar los diagramas de clases del diseño se emplea la herramienta CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) Rational Rose, actualmente conocida como una familia de software de IBM para el levantamiento de requerimientos, diseño, construcción, pruebas y administración de proyectos en el proceso desarrollo de software. Sus productos están centrados en la metodología del Proceso Racional Unificado o RUP (Rational Unified Process) y utiliza el lenguaje de modelado UML.

UML:

UML (Unified Modeling Language, Lenguaje Unificado de Modelado) se ha convertido en el estándar de facto para definir, organizar y visualizar los elementos

que configuran la arquitectura de una aplicación orientada a objetos. Constituye el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

2.2.2 Modelo de diseño.

Para lograr que en un futuro se puedan implementar nuevos algoritmos e incluso optimizar el existente se han empleado interfaces, y para mantener una estructura organizada utilizamos módulos a los cuales le asignamos determinada responsabilidad. Bajo este esquema ha sido realizado el modelo de diseño. En esta sección se efectuará una explicación detallada del diseño de la biblioteca.

La biblioteca de clases tiene implementado cinco módulos básicos: *Document*, *DocumentParser*, *interfaces*, *URelevanceMeasure*, *Fukumoto* y tres secundarios: *Evaluacion*, *MedidasRelevancia*, *EsquemasPeso*. En la tabla mostramos un esquema con algunos de estos componentes con sus entradas y salidas.

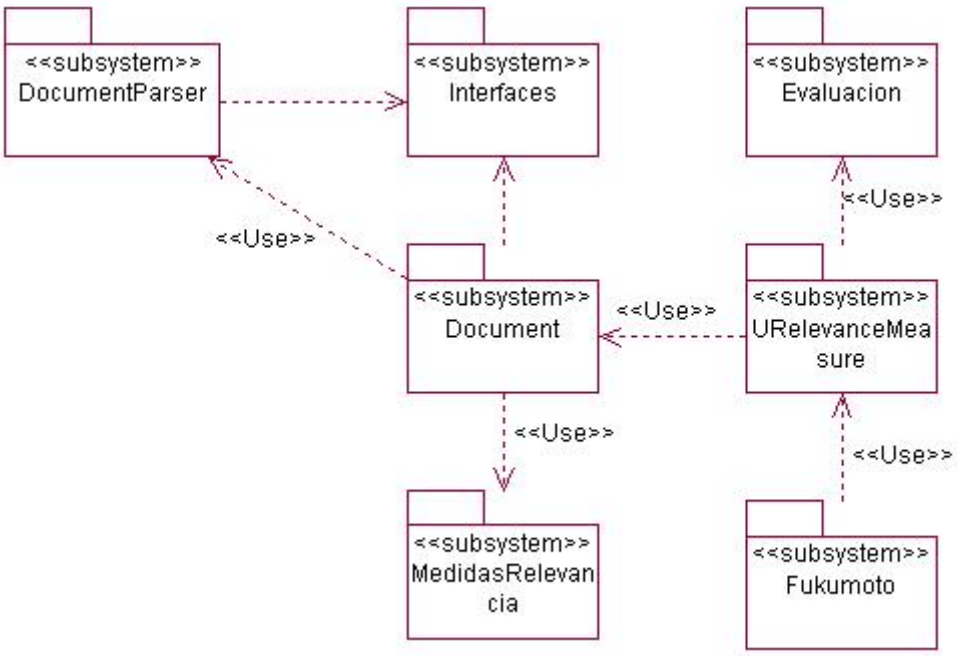

Figura 2-1 Modelo de diseño de los principales módulos de la biblioteca.

2.2.2.1 Módulo Documento.

Este módulo constituye el componente principal en la elaboración del resumen, tiene la responsabilidad de llevar el texto a una representación en un espacio vectorial; se crean los vectores peso de la frecuencia del término en el documento y peso de la frecuencia del término en la oración o párrafo (en este caso es en la oración) a partir de la frecuencia de los términos en el documento y en la oración, permitiendo calcular la puntuación de relevancias de las oraciones en el documento y utilizando como soporte los módulo *MedidasRelevancia* y *EsquemasPeso*.

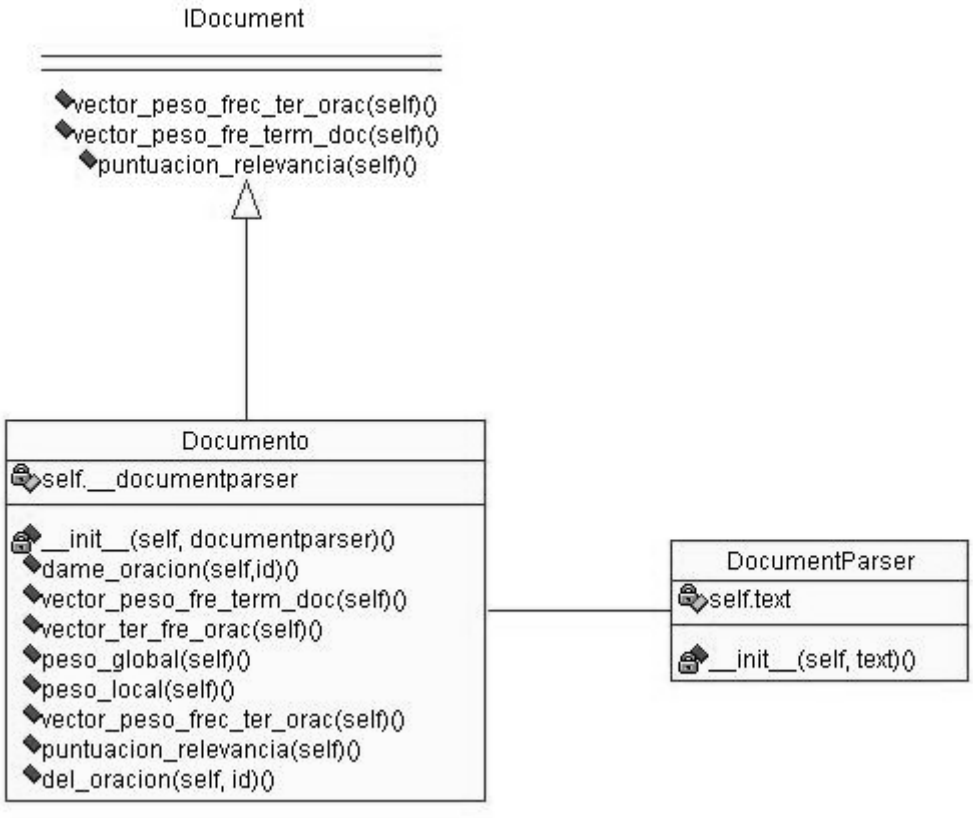

Figura 2-2 Diagrama de clases del módulo Document

2.2.2.2 Módulo MedidasRelevancia.

En este módulo están implementadas las medidas de relevancia utilizadas por el algoritmo durante el proceso de resumen. Tiene implementada tres medidas de relevancia: *Inner Product, Cosine Similarity* y *Jaccard Co-eficcient*. Actualmente

se utiliza por defecto el *Inner Product*, debido a que es la más empleada aunque se puede utilizar cualquiera de las dos medidas restantes.

2.2.2.3 Módulo EsquemasPeso.

Es responsable de implementar los diferentes esquemas de peso local y global que permiten la creación de los vectores. Tiene implementados cuatro esquemas de peso local, dos esquemas de peso global y dos métodos que permiten ordenar descendentemente una estructura en Python, por ejemplo tuplas.

2.2.2.4 Módulo DocumentParser.

Este módulo constituye el analizador gramatical, recibe como entrada el texto, divide la entrada de texto en pequeñas partes (oraciones) delimitadas por un punto (.) y las procesa. Tiene implementado un parser para el idioma español e inglés que devuelve dos estructuras manejables para hacer los vectores, el id de la oración con la oración intacta y el id de la oración con la oración dividida en tokens, además elimina los signos de puntuación y las palabras de parada con el objetivo de lograr optimizar el proceso (ver Anexo B y C).

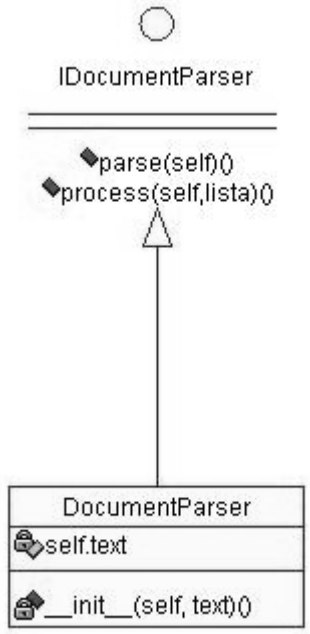

Figura 2-3 Diagrama de clases del módulo DocumentParser

2.2.2.5 Módulo Evaluacion.

Este módulo tiene la responsabilidad de evaluación de la calidad de los resúmenes obtenidos. En estos momentos tiene implementadas las medidas de evaluación Precision, Recall y F-Measure.

2.2.2.6 Módulo interfaces.

El módulo es responsable de la implementación de las interfaces, garantizando de esta forma, que las interfaces que se vayan a definir en posteriores trabajos sean implementadas aquí. Se han definido dos interfaces *IDocument* e *IDocumentParser*, para normalizar los resultados de las clases que hereden de ellas, manteniendo así, el comportamiento de futuras clases implementadas.

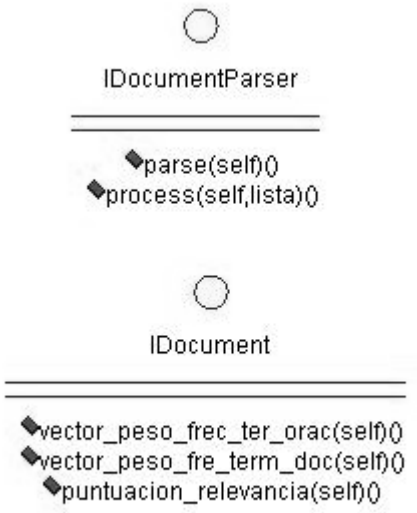

Figura 2-4 Diagrama de clases del módulo interfaces

2.2.2.7 Módulo URelevanceMeasures.

Su objetivo es contener la implementación del algoritmo monodocumento de la biblioteca. Actualmente tiene implementado el algoritmo URelevanceMeasure basado en el coeficiente de relevancia de la oración y para ello se ha construido una clase UrelevanceMeasure, encargada de realizar los pasos del algoritmo y

brindar un resumen. Antes de devolver el resumen se ordenan las oraciones con el objetivo de crear un resumen más coherente y entendible.

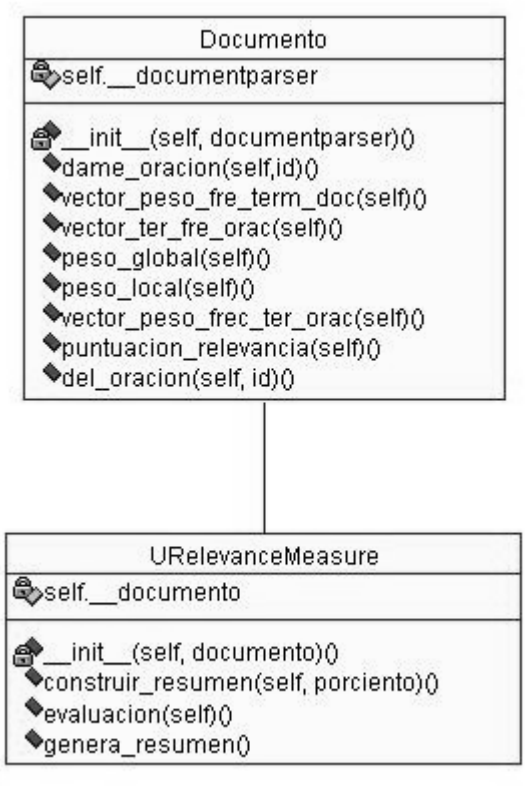

Figura 2-5 Diagrama de clases del módulo URelevanceMeasure

2.2.2.8 Módulo Fukumoto.

Su objetivo es contener la implementación del algoritmo multidocumento de la biblioteca. Actualmente tiene implementado el algoritmo Fukumoto y para ello se ha construido una clase Fukumoto, encargada de construir los vectores, calcular el valor de semejanza de las oraciones, eliminar las oraciones innecesarias y finalmente construir el resumen multidocumento.

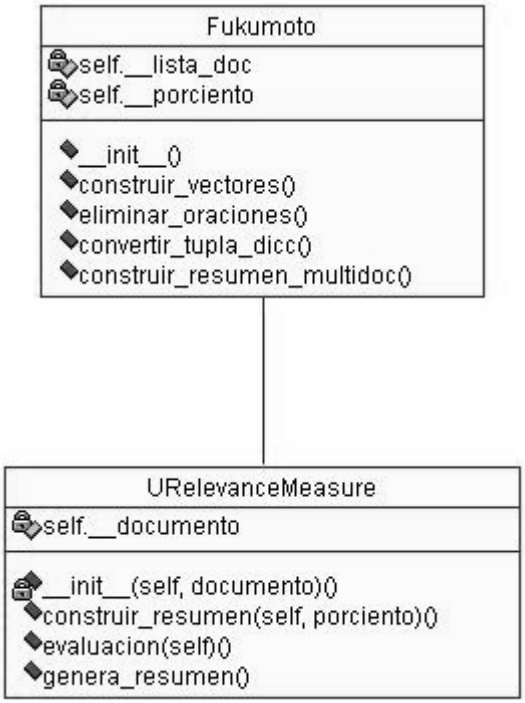

Figura 2-6 Diagrama de clases del módulo Fukumoto

2.2.3 Estándares de codificación del lenguaje de programación Python.

Con vistas a garantizar la homogeneidad del código dentro del grupo de desarrollo, se establece el estilo de código descrito a continuación.

Indentación:

Python utiliza indentación para delimitar los bloques de código, es decir, para escribir un ciclo en python el inicio y el fin de ese ciclo esta delimitado por la misma indentación mientras que en los lenguajes la indentación se representa en algún tipo de notación especifica (*{ }, if...endif,* etc.).

Comentarios:

Los comentarios deben ser oraciones completas. Si el comentario es una frase o una oración, su primera palabra debe estar en mayúscula, a menos que sea un identificador que empiece con una letra minúscula.

Si un comentario es corto, el punto al final puede ser omitido. Los bloques de comentarios generalmente consisten en uno o más párrafos construidos con oraciones completas, y cada una de estas oraciones debe terminar en un punto.

Se debe usar dos espacios después del punto final de la oración.

Los bloques de comentarios se aplican generalmente a cierto (o a todo el) código que los siga, y están indentados al mismo nivel que ese código. Cada línea del bloque de comentario empieza con el signo # y un espacio simple (a menos que sea indentado dentro del comentario).

Los párrafos dentro de un bloque de comentario están separados por una línea conteniendo un simple #.

Comentarios en línea:

Un comentario de línea es un comentario que se hace en una misma línea de una declaración. El comentario debe estar separado por dos espacios después de la declaración. Deben empezar con el símbolo # y un espacio simple.

Declaraciones:

En Python la declaración de variables es dinámica, es decir, no se tienen que declarar las variables ni tener en cuenta su tamaño, ya que son completamente dinámicas.

Espacios en blanco:

Siempre rodear estos operadores binarios con un espacio simple a cada lado: Asignación (=), Asignación aumentada o decrementada (+=, -= etc.), Comparaciones (==, <, >! =, <>, <=, >=, in, not in, is, is not), Booleanos (and, or, not).

Use espacio en blanco alrededor de operadores aritméticos.

2.2.4 Implementación.

En este epígrafe se comentan algunas partes importantes del código de la biblioteca. En sus inicios se ha concebido la implementación de dos algoritmos, los cuales sólo procesan documentos en formato .txt

Parser:

Un analizador sintáctico (parser) en informática y lingüística es un proceso que analiza secuencias de tokens para determinar su estructura gramatical respecto a una gramática formal dada, es un módulo, biblioteca o programa que se ocupa de transformar un archivo de texto en una representación interna. Un parser es así mismo un programa que reconoce si una o varias cadenas de caracteres forman parte de un determinado lenguaje, es utilizado por ejemplo en compiladores.

Análisis gramatical de los textos: luego de dividir el texto en unidades más pequeñas (tokens), se eliminan de éstos los signos de puntuación y los que constituyen palabras de parada (artículos, preposiciones, conjunciones, pronombres) (ver Anexo B y C) son eliminadas, a continuación se muestra el método encargado de realizar este proceso.

```
28    def parser(self):
29        """Método principal de este modulo, devuelve dos diccionarios
30        (1) con id_oracion y los token en la oracion
31        (2) con id_oracion y la oracion"""
32        palabra_parada=StopWordRemover()
33        tokens = (token for token in self.text.split())
34        oracion=" "
35        v_oracion={}
36        contador=0
37        v_ter_ora={}
38        for token in tokens:
39            if "." in token:
40                oracion=oracion+token
41                v_oracion[contador]=oracion
42                valor= self.__process(oracion.split())
43                ter_ora=palabra_parada.process(valor)
44                v_ter_ora[contador]= ter_ora
45                contador+=1
46                oracion=""
47            else:
48                oracion=oracion+token+" "
49        return v_ter_ora,v_oracion
```

Figura 2-7 Método que contiene analizador gramatical

Función que elimina los signos de puntuación: recibe como entrada una lista de tokens y los devuelve sin signos de puntuación la expresión regular ‘\w’, elimina todo lo que no corresponda con letras y números.

```
21    def process(self, lst):
22        """Mediante esta funcion elimino todos los signos
23         de puntuacion de los tokens y los devuelvo en español"""
24        locale.setlocale(locale.LC_ALL, 'sp')
25        rx = re.compile(r"(?L)\w+")
26        li = []
27        for s in lst:
28            li += rx.findall(s)
29        return li
```

Figura 2-8 Método que devuelve los tokens sin signos de puntuación

Método que describe el algoritmo monodocumento: Se calcula inicialmente la cantidad de oraciones que debe contener el resumen, a partir del porciento de resumen deseado, luego se adicionan las oraciones más relevantes al resumen y finalmente después de ordenarlas se construye el resumen.

```
15    def construir_resumen(self, porciento):
16        """En este metodo se construye el resumen"""
17        resumen=[]
18        resumen1=""
19        cant_orac_resumen =(len(self.__oracs_ordenadas)* porciento / 100)
20        for par in doc.puntuacion_relevancia():
21            prueba = doc.puntuacion_relevancia()[0]
22            if len(resumen)< int(cant_orac_resumen):
23                resumen.append(prueba[0])
24                doc.del_oracion(prueba[0])
25                doc.vector_peso_fre_term_doc()
26                doc.puntuacion_relevancia()
27        resumen.sort()
28        for id in resumen:
29            resumen1=resumen1+doc.dame_oracion(id)+" "
30        return resumen1
```

Figura 2-9 Método que describe los pasos del algoritmo

Método que construye el resumen multidocumento: Este método a partir del vector que contiene las oraciones con sus respectivas semejanzas que van a constituir el resumen, vector que se construye dentro de la clase Fukumoto, las ordena y construye el resumen multidocumento.

```
97    def construir_resumen_multidoc(self):
98        """Metodo que construye el resumen multidocumento"""
99        resumen = ""
100       id_docs = self.__id_doc2id_orac2vs.keys()
101       id_docs.sort()
102       for id_doc in id_docs:
103           id_oracs = self.__id_doc2id_orac2vs[id_doc].keys()
104           id_oracs.sort()
105           for id_orac in id_oracs:
106               oracion = self.__id_doc2id_orac2oracion[id_doc][id_orac]
107               resumen = resumen + oracion + " "
108       return resumen
```

Figura 2-10 Construye el resumen Multidocumento

2.3 Evaluación de los resultados.

Para poder evaluar los resultados, primeramente se estudian los métodos y métricas para la evaluación de los resúmenes. Se dice que una métrica es intrínseca o extrínseca en dependencia de si la métrica determina la calidad basada solo en el resumen, o si es basada en la utilidad de éste para completar otra tarea. Un ejemplo de medida intrínseca es la similitud del coseno del resumen del documento del que ha sido generado. Esta medida particular no es muy útil, ya

que no toma en cuenta la redundancia. Una métrica empleada normalmente para los resúmenes extractos fue propuesta por Edmundson (EDMUNDSON 1969) en el cual los jueces humanos seleccionan oraciones de los documentos para crear los resúmenes extractos de forma manual. Los resúmenes automáticos se evalúan entonces calculando la cantidad de oraciones comunes al resumen automático y al manual. En términos de Recuperación de Información, estas medidas se llaman Precision y Recall. Este método de evaluación es el más utilizado actualmente para evaluar los resúmenes extractos (GOLDSTEIN *et al.* 2000).

Pasos para realizar la evaluación de los resultados:

1. Se crea un corpus de documentos.
2. Luego se selecciona aleatoriamente una muestra de 50 documentos para realizar la evaluación.
3. Un grupo de jueces humanos realizarán los extractos manuales, llamados resúmenes de referencia.
4. Se aplican las métricas.

Explicación de los pasos:

Se prepara un corpus que esta compuesto por un total de 100 documentos de diferentes tipos de noticias tales como: política, animales, arte, literatura, salud, personalidades de la literatura, azúcar, ciencia, deporte, cultura, noticias de cuba y del mundo, historia de cuba y chocolate extraídas de diversas fuentes bibliográficas de nuestro país tales como: los periódicos: Juventud Rebelde, Granma, Vanguardia y las revistas: Bohemia y Somos Jóvenes

Luego se selecciona aleatoriamente una muestra de 50 documentos para realizarle las pruebas que garantizará que ningún factor externo, dígase el humano, influya en la selección de estos documentos de prueba. Esto se hace de la siguiente manera:

Utilizando la herramienta computacional Microsoft Excel se generan los 50 números aleatorios mediante la función random: $RAND()*(b-a)+a$ donde b es la cantidad de documentos de la población y a es la cantidad de documentos

aleatorios a generar por iteración. Particularmente se le asigna a “b” el valor de 100 documentos y a la variable” a” 1.

Para poder evaluar la calidad de los resúmenes automáticos se debe hacer una comparación con los resúmenes realizados por los jueces humanos, que son los llamados resúmenes “ideales”. Pero en el estudio realizado se demuestra que el juicio humano para la evaluación de la calidad de los resúmenes varía de persona a persona y entonces se decidió crear 5 grupos de documentos de la muestra del corpus seleccionada compuestos por 10 documentos cada uno y se le asignó a cada grupo 2 jueces que resumirán de forma manual cada documento.

Se aplicaron las métricas Recall, Precision y F-Measure que son métricas tradicionales de IR: Precision (porcentaje de oraciones recuperadas que son relevantes), Recall (proporción de oraciones relevantes existentes en la colección que se han recuperado) y F-Measure (porcentaje de calidad del resumen). Las cuales están siendo utilizadas con frecuencia para medir la eficacia de los resúmenes automáticos en comparación con otros confeccionados manualmente y tomados como referencia.

$$Precision = \frac{|E \cap R|}{|E|} \quad ó \quad Precision = \frac{c}{m}$$

$$Recall = \frac{|E \cap R|}{|R|} \quad ó \quad Recall = \frac{c}{n}$$

$$F - Measure = \frac{2Presicion * Recall}{Precision + Recall}$$

$donde$:

m : Longitud de medida en número de frases del resumen automático.

c : Número de frases coincidentes.

n : Longitud de medida en número de frases del resumen de referencia.

E, R : Conjunto de sentencias del extracto que se evalúa y el conjunto de sentencias del resumen de referencia, respectivamente.

De estas, la más utilizada es el Recall, que mide la tasa de frases del resumen de referencia presentes en el resumen generado automáticamente. El mayor inconveniente que tiene este tipo de técnicas es que pueden proporcionar resultados distintos para resúmenes que contengan la misma información.

2.3.1 Resultados obtenidos con el algoritmo monodocumento.

La tabla #1 muestra para cada documento las oraciones seleccionadas por cada juez para formar parte del resumen extracto llamado “resumen de referencia o ideal”, así como la cantidad de oraciones seleccionadas que son comunes para ambos resúmenes de referencia, analizando el grado de similitud que tendrán dichos resúmenes, demostrando que hasta los resúmenes generados por las personas pueden ser diferentes.

# Doc.	Juez 1	Juez 2	Oraciones comunes	% Similitud
1	0,2,4	0,2,7	2	66.66
2	0,1,2,7,8	0,1,2,14,15	3	60
3	0,1,2,3,4,12,16,17,32	4,10,12,16,17,22,28,29,38,39	4	
4	2,5,6,7,14	2,3,5,6,19	3	80
5	0,2,11,15	0,2,8,9	2	50
6	0,5,10,13	0,3,10,13	3	75
7	0,1,3	0,1,2	2	66.66
8	0,3	0,2	1	50
9	0,2,3,4,5,6,7,8,10,15,32,39,57,75,76,77	0,2,3,4,5,6,7,8,10,16,24,25,53,75,76,77	11	68.75
10	0,12,13	1,12,13	2	66.66
11	0,1,7,8,25,27,29,32,55,56,57,64,72,74,80,81	0,1,7,9,25,26,28,30,55,56,57,68,72,74,80,81	12	75
12	0,1,4,5,6,29,30,31,32,33,34,42,64,	0,1,3,5,6,29,30,31,35,36,53,56,64,	8	61.53
13	0,1,3,6,9	0,1,4,7,9	3	60
14	0,4	1,4	1	50
15	0,4,15	1,4,15	2	66.66
16	0,11,13,17,21,23,25,38	0,11,13,17,18,20,25,38	6	75
17	0,3	0,4	1	50
18	0,1,4	0,1,5	2	66.66
19	6,7,26,27,30,31,32	6,9,26,27,30,31,32	6	85.71
20	4,6,7,8,12,13,24,25,52,53,56,57,59	4,6,7,12,13,14,24,25,26,27,52,53,61	9	69.23

Tabla #1: Similitud entre los resúmenes ideales de 20 artículos generados por los jueces humanos.

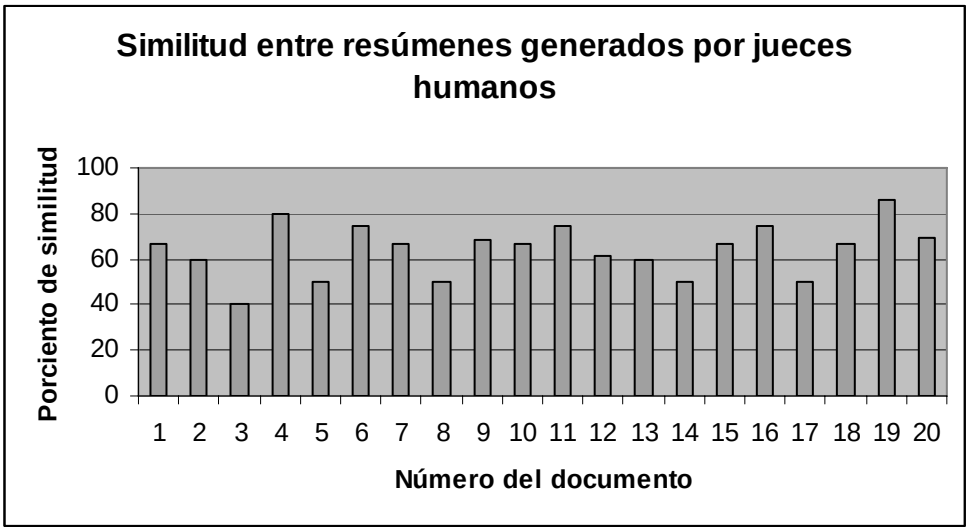

Figura 2-11 Similitud entre resúmenes ideales generados por jueces humanos para 20 artículos

La tabla #2 muestra para los 10 documentos del grupo 1 la cantidad de oraciones que tiene el resumen automático para una tasa de compresión del 20 % y los valores de Precision, Recall y F-Measure entre el resumen automático y el resumen ideal de cada juez.

#Doc	Long.resumen automático (oraciones)	J1 (P)	J2 (P)	J1(R)	J2(R)	J1(F)	J2(F)
1	2	0.5	1	0.33	0.6	0.4	0.75
2	4	0.5	0.75	0.4	0.6	0.44	0.67
3	8	0.375	0.75	0.33	0.6	0.35	0.67
4	4	0.25	0.5	0.2	0.4	0.22	0.44
5	3	0.6	0.6	0.5	0.5	0.54	0.55
6	3	0.33	0.33	0.25	0.25	0.28	0.28
7	2	0.5	0	0.33	0	0.4	0
8	1	1	0	0.5	0	0.67	0
9	15	0.2	0.26	0.18	0.15	0.19	0.19
10	3	0.6	0.6	0.6	0.6	0.6	0.6

Tabla # 2: La tabla muestra los valores de Precisión (P), Recall (R) y F-Measure (F) para cada juez del grupo de documentos #1.

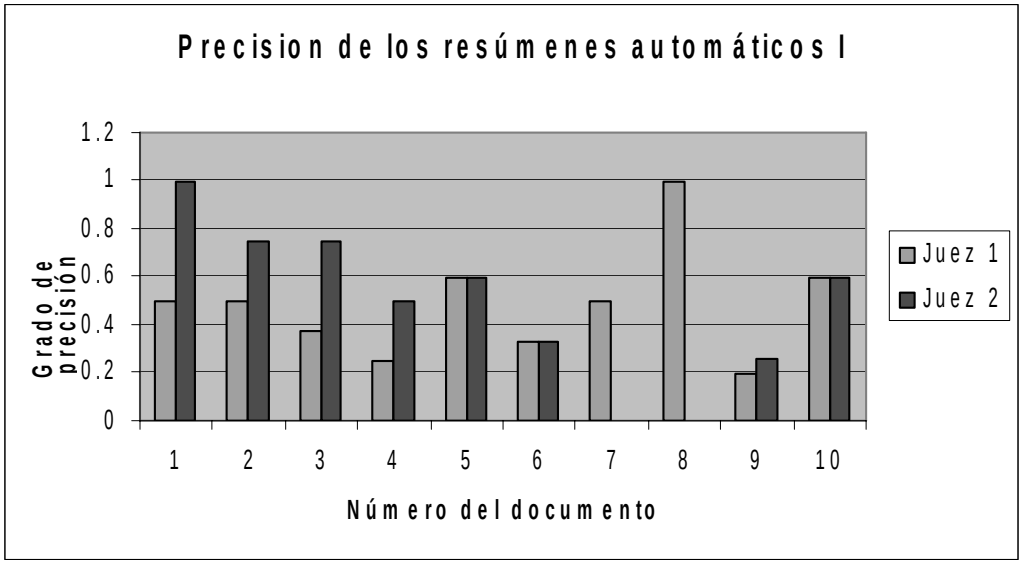

Figura 2-12 Representa la Precision de cada juez para cada documento del 1 al 10.

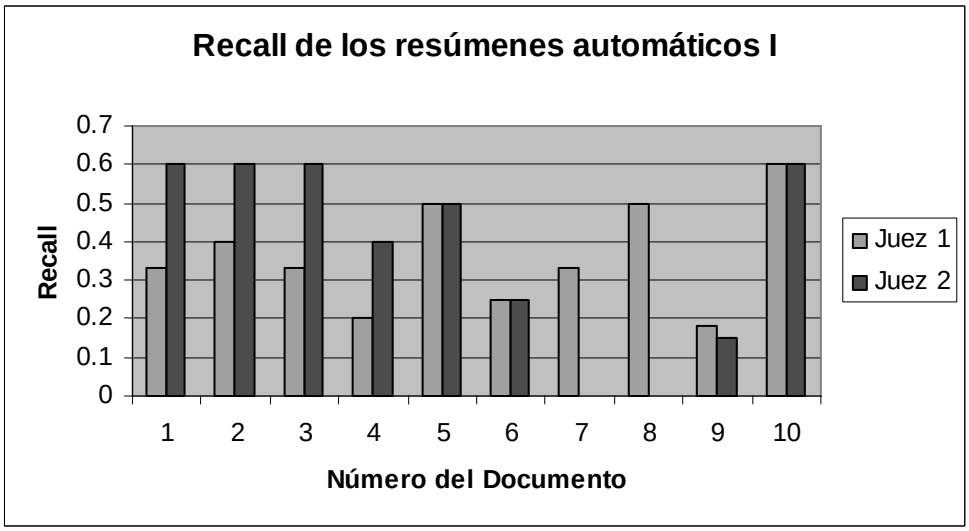

Figura 2-13 Representa el Recall de cada juez para cada documento de los documentos del 1 al 10

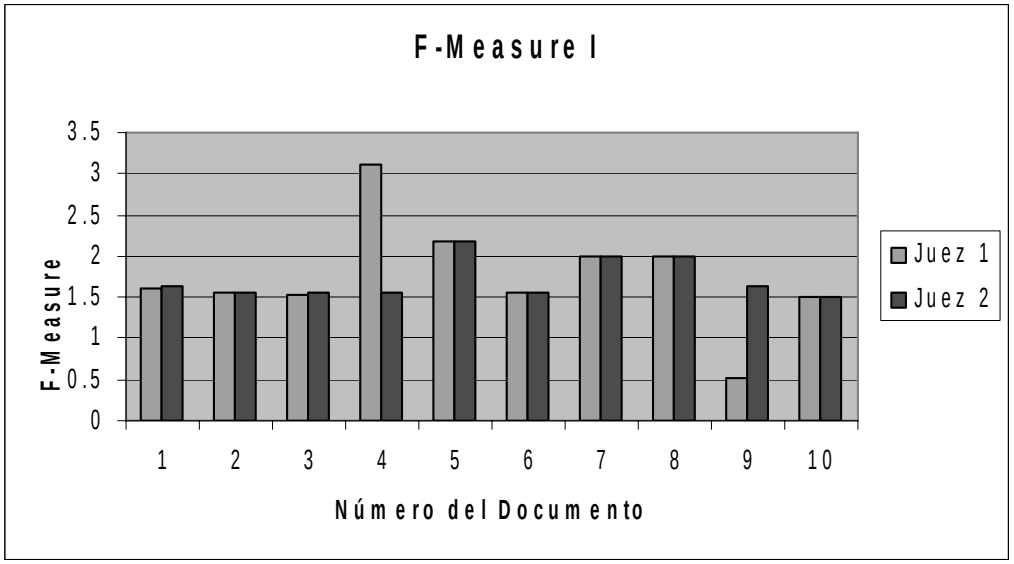

Figura 2-14 Representa el F-Measure de cada juez para cada documento del 1 al 10.

La tabla #3 muestra para los 10 documentos del grupo 2 la cantidad de oraciones que tiene el resumen automático para una tasa de compresión del 20 % y los valores de Precision, Recall y F-Measure entre el resumen automático y el resumen ideal de cada juez.

# Doc	Long.resumen automático (oraciones)	J1(P)	J2(P)	J1(R)	J2(R)	J1(F)	J2(F)
11	16	0.56	0.44	0.56	0.44	0.56	0.88
12	13	0.23	0.15	0.23	0.15	0.23	0.15
13	4	0.5	0.25	0.5	0.2	0.5	0.22
14	1	1	1	0.5	0.5	0.67	0.67
15	3	0.6	0.33	0.6	0.33	0.6	0.33
16	8	0.5	0.62	0.5	0.62	0.5	0.62
17	1	1	1	0.5	0.5	0.67	0.67
18	2	0.5	0.5	0.33	0.33	0.4	0.4
19	6	0.33	0.33	0.28	0.28	0.3	0.3
20	12	0.41	0.58	0.38	0.54	0.39	0.56

Tabla # 3: La tabla muestra los valores de Precisión (P), Recall (R) y F-Measure (F) para cada juez del grupo de documentos # 2.

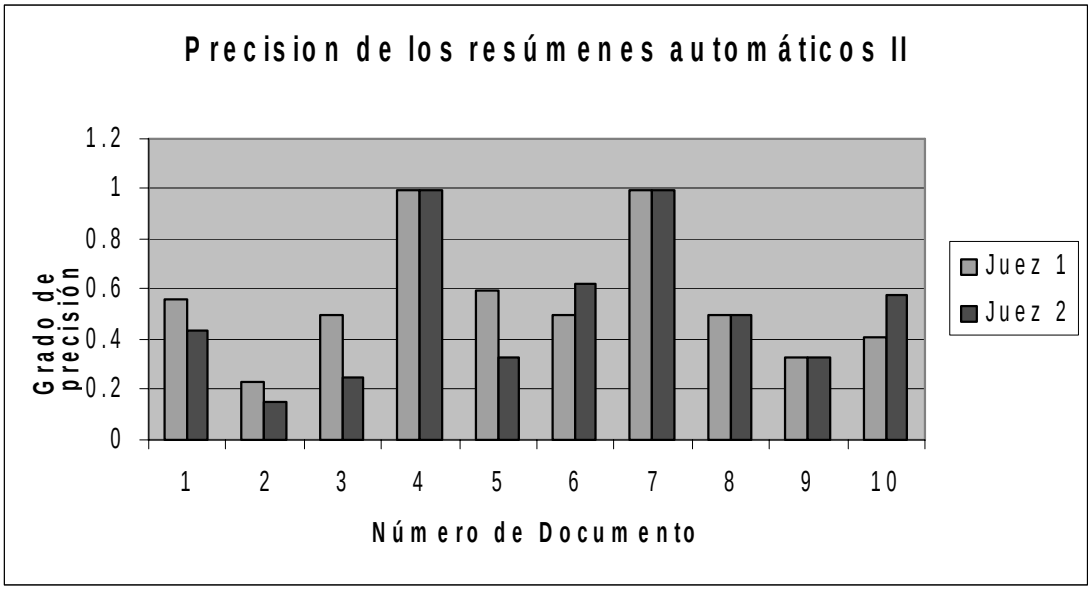

Figura 2-15 Representa la Precision de cada juez para cada documento del 11 al 20.

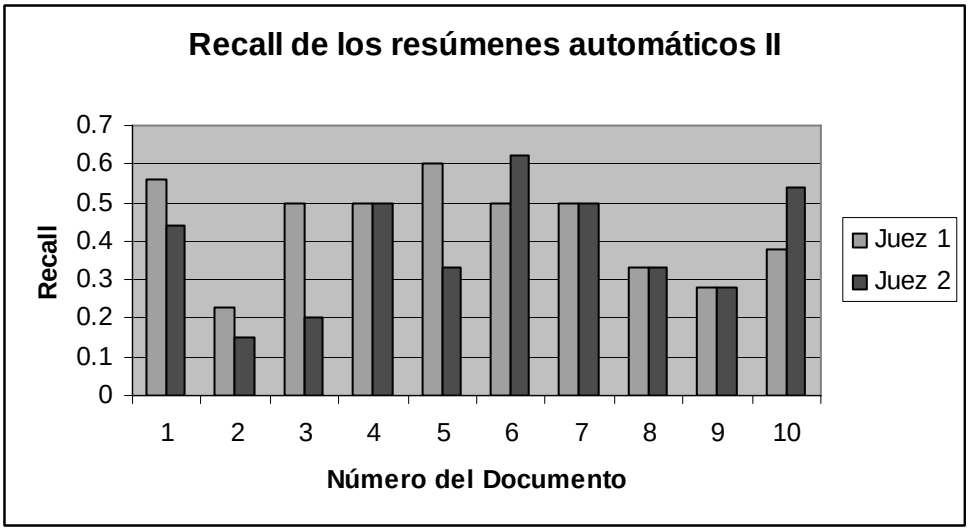

Figura 2-16 Representa el Recall de cada juez para cada documento del 11 al 20.

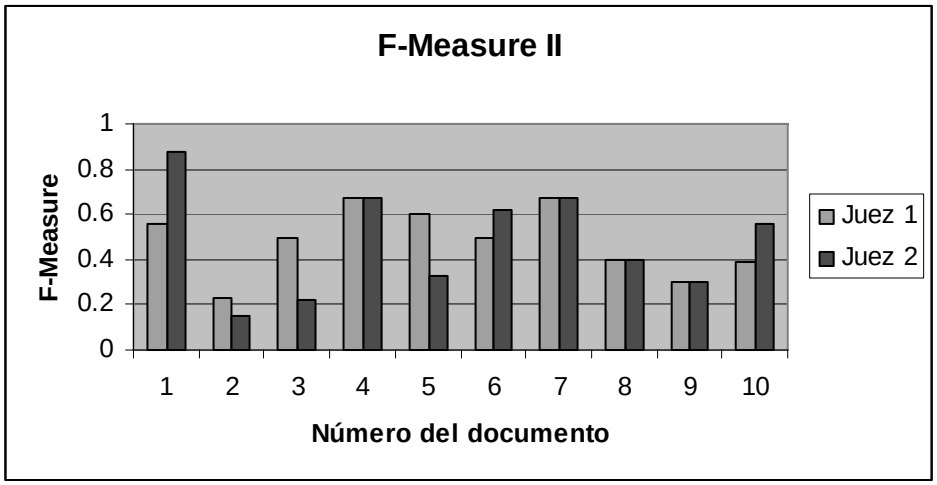

Figura 2-17 Representa el F-Measure de cada juez para cada documento del 11 al 20.

La tabla #4 muestra para los 10 documentos del grupo 3 la cantidad de oraciones que tiene el resumen automático para una tasa de compresión del 20 % y los valores de Precision, Recall y F-Measure entre el resumen automático y el resumen ideal de cada juez.

# Doc	Long.resumen automático (oraciones)	J1(P)	J2 (P)	J1(R)	J2(R)	J1(F)	J2(F)
21	1	0	0	0	0	0	0
22	1	0.75	0.83	0.69	0.77	0.72	0.8
23	1	1	1	0.5	0.5	0.67	0.67
24	4	0.5	0.5	0.4	0.4	0.44	0.44
25	5	0.6	0.6	0.5	0.5	0.55	0.55
26	11	0.55	0.73	0.5	0.67	0.52	0.7
27	3	0.33	0	0.25	0	0.28	0
28	2	0.5	0.5	0.33	0.33	0.4	0.4
29	5	0.6	0.4	0.5	0.33	0.55	0.36
30	2	1	1	0.67	0.67	0.8	0.8

Tabla # 4: La tabla muestra los valores de Precisión (P), Recall (R) y F-Measure (F) para cada juez del grupo de documentos # 3.

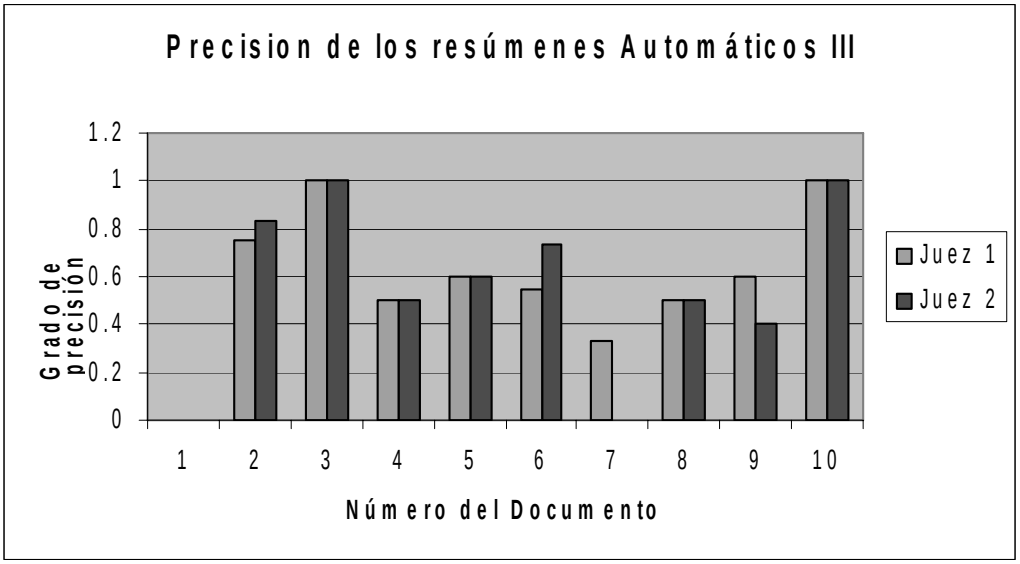

Figura 2-18 Representa la Precision de cada juez para cada documento del 21 al 30.

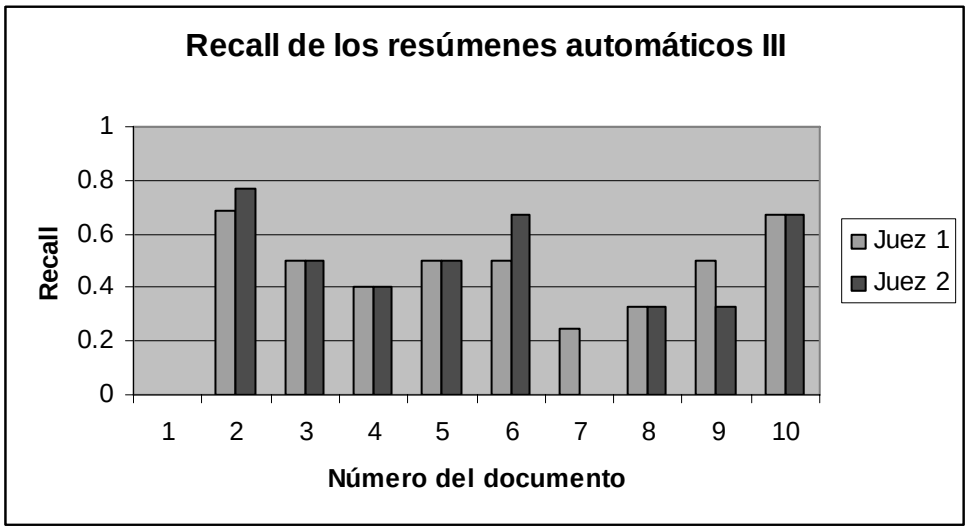

Figura 2-19 Representa el Recall de cada juez para cada documento del 21 al 30.

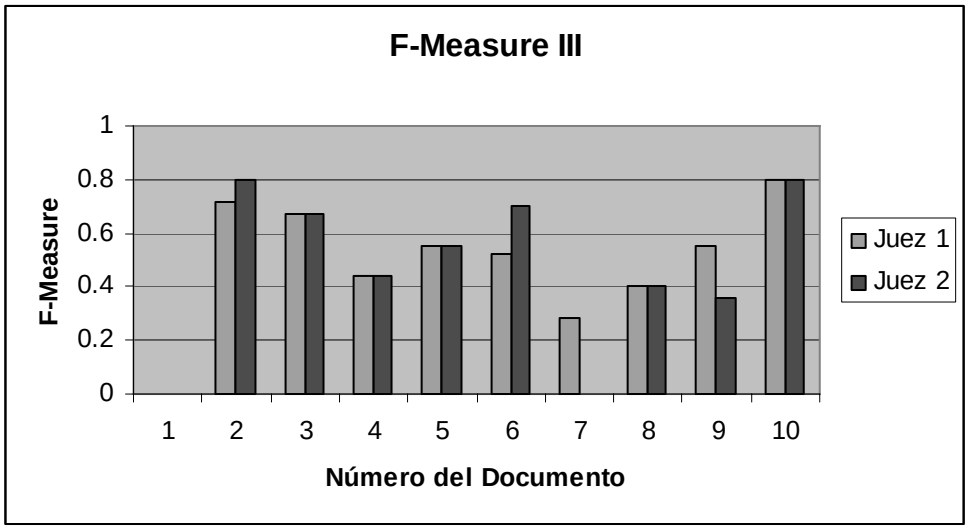

Figura 2-20 Representa el F-Measure de cada juez para cada documento del 21 al 30.

La tabla #5 muestra para los 10 documentos del grupo 4 la cantidad de oraciones que tiene el resumen automático para una tasa de compresión del 20 % y los valores de Precision, Recall y F-Measure entre el resumen automático y el resumen ideal de cada juez.

# Doc	Long.resumen automático (oraciones)	J1(P)	J2(P)	J1(R)	J2(R)	J1(F)	J2(F)
31	2	0	0.5	0	0.33	0	0.4
32	8	0.63	0.75	0.56	0.67	0.59	0.71
33	4	0.5	0.5	0.4	0.4	0.44	0.44
34	1	0	0	0	0	0	0
35	2	1	1	0.67	0.67	0.8	0.8
36	1	1	1	0.5	0.5	0.67	0.67
37	2	0.5	0.5	0.33	0.33	0.4	0.4
38	2	0.5	0.5	0.33	0.33	0.4	0.4
39	3	0.67	0.67	0.5	0.5	0.57	0.57
40	10	0.7	0.9	0.64	0.82	0.67	0.86

Tabla # 5: La tabla muestra los valores de Precisión (P), Recall (R) y F-Measure (F) para cada juez del grupo de documentos # 4.

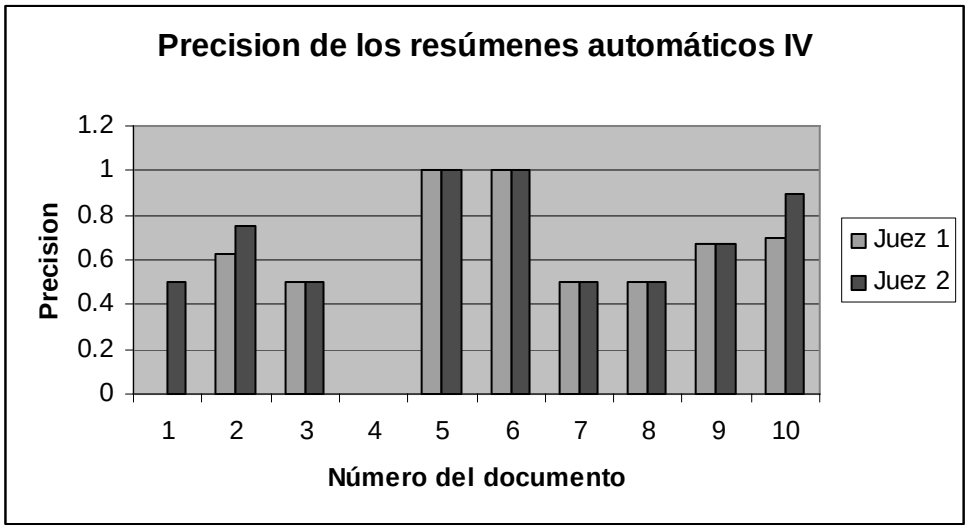

Figura 2-21 Representa la Precision de cada juez para cada documento del 31 al 40.

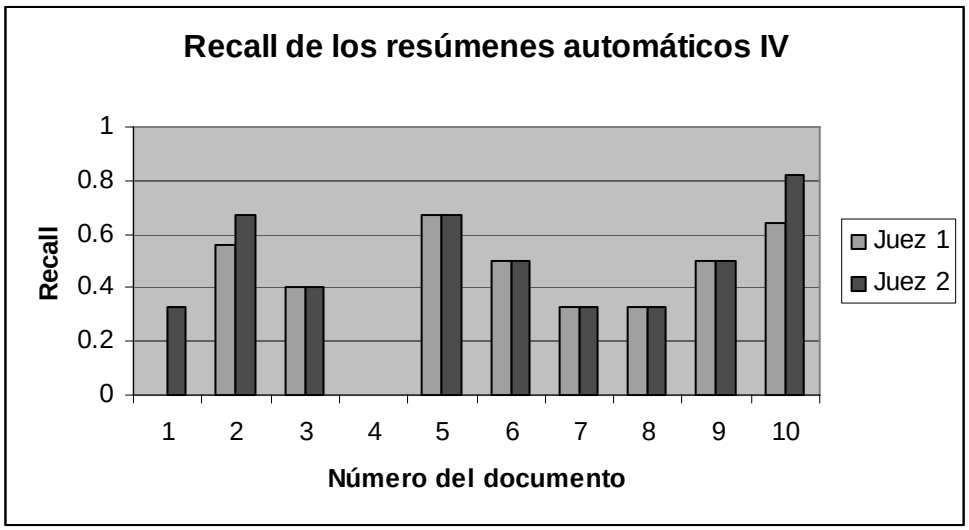

Figura 2-22 Representa el Recall de cada juez para cada documento del 31 al 40.

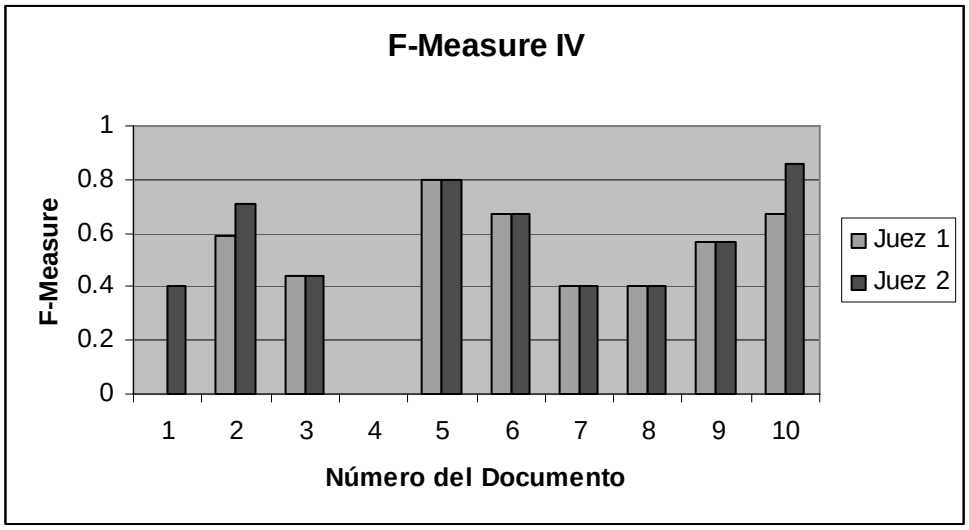

Figura 2-23 Representa el F-Measure de cada juez para cada documento del 31 al 40.

La tabla #6 muestra para los 10 documentos del grupo 5 la cantidad de oraciones que tiene el resumen automático para una tasa de compresión del 20 % y los valores de Precision, Recall y F-Measure entre el resumen automático y el resumen ideal de cada juez.

# Doc	Long.resumen automático (oraciones)	J1(P)	J2(P)	J1(R)	J2(R)	J1(F)	J2(F)
41	5	0.4	0.6	0.3	0.6	0.3	0.6
42	1	1	0	0.5	0	0.7	0
43	7	0.6	0.6	0.5	0.57	0.5	0.58
44	5	0	0.6	0	0.6	0	0.6
45	2	0.5	1	0.3	1	0.5	1
46	7	0.6	0.7	0.5	0.71	0.5	0.71
47	2	0.5	1	0.3	1	0.5	1
48	4	0.8	0.5	0.6	0.5	0.7	0.5
49	6	0.3	0.3	0.3	0.33	0.3	0.32
50	22	0.8	0.7	0.7	0.68	0.8	0.7

Tabla # 6: La tabla muestra los valores de Precisión (P), Recall (R) y F-Measure (F) para cada juez del grupo de documentos # 5.

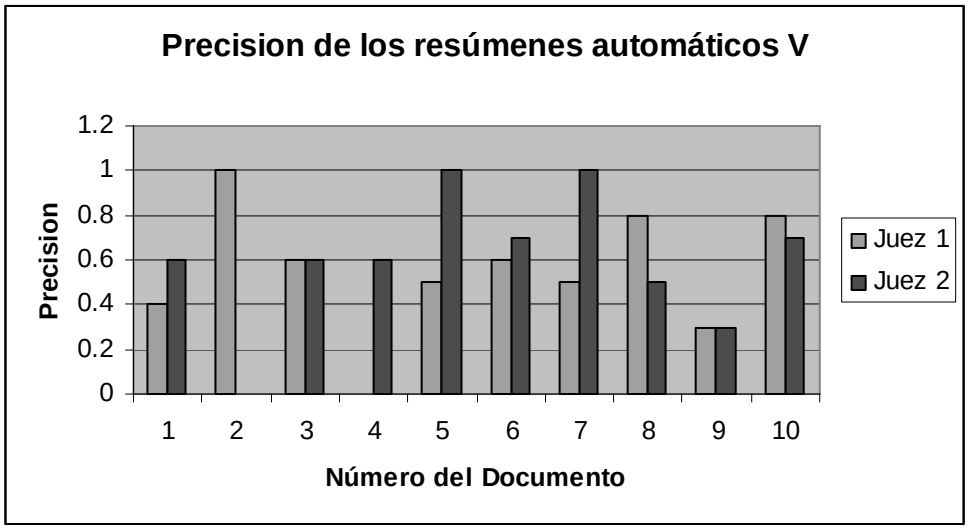

Figura 2-24 Representa la Precision de cada juez para cada documento del 41 al 50.

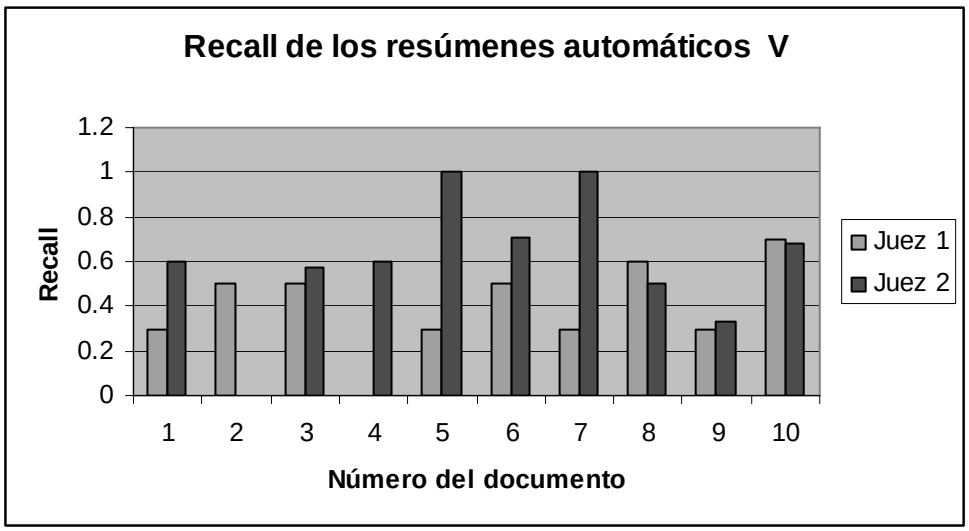

Figura 2-25 Representa el Recall de cada juez para cada documento del 41 al 50.

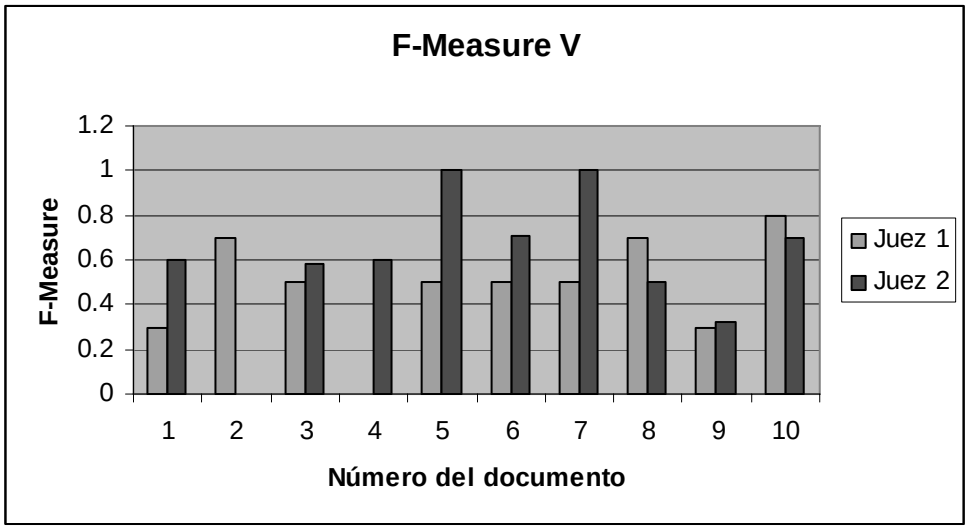

Figura 2-26 Representa el F-Measure de cada juez para cada documento del 41 al 50.

2.3.2 Resultados obtenidos con el algoritmo multidocumento

Para la evaluación de este algoritmo se ha utilizado una técnica muy sencilla en la cual intervienen jueces humanos, los cuales evalúan la calidad del resumen.

Pasos:

1. Selección de los documentos.
2. Selección de los Jueces.
3. Aplicación de la métrica de evaluación de los jueces a los resúmenes.
4. Calculo del promedio de la calidad del resumen.

Explicación de los pasos:

Se conformaron dos corpus de cinco y diez documentos respectivamente extraídos de diferentes fuentes que tratan sobre un mismo tema (Juegos panamericanos en Brasil y Posada Carriles). Se seleccionaron 8 jueces para la evaluación de cada uno de los resúmenes. Luego se realizó la evaluación de estos resumen a partir de un rango del 1 al 10 emitido por los jueces. Se calculó basándose en los resultados de los jueces el promedio de la calidad de los resúmenes. A continuación se muestran los gráficos correspondientes con los resultados.

La siguiente tabla muestra la valoración realizada por cada juez al resumen multidocumento en un rango del 1 al 10, para las dos colecciones de documentos.

No. Juez	Resumen automatico_Juegos Panamericanos	Resumen automatico_Posada Carriles
1	7	9
2	6	9
3	5	9
4	6	8
5	6	9
6	6	8
7	7	8
8	5	8

Tabla # 7: La tabla muestra los valores del resumen de cada juez en un rango de 1 a 10.

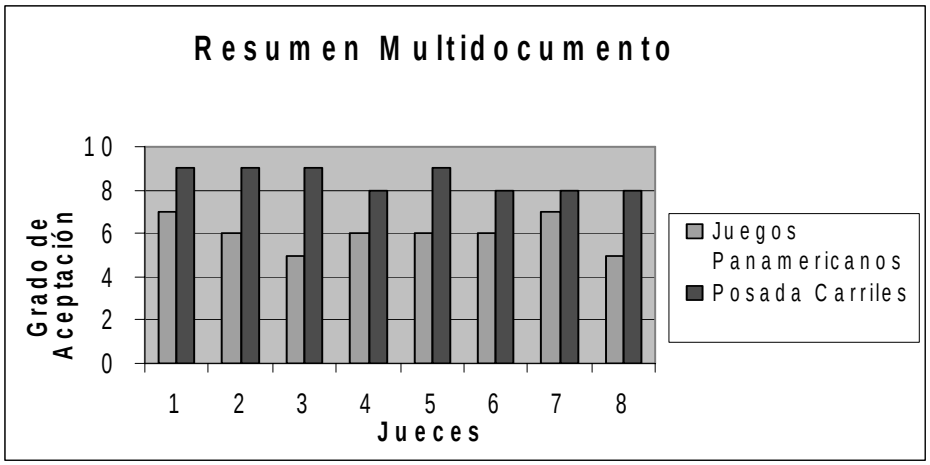

Figura 2-27 La media de calidad de los resúmenes automáticos es del 72, 50 %

2.4 Conclusiones parciales

Durante el diseño se estableció una arquitectura básica de la biblioteca lista para su posterior ampliación y a pesar de estos algoritmos no tener las entradas homogeneizadas en la arquitectura se estableció como entrada para los algoritmos el texto y el ratio de compresión.

Al evaluar los resúmenes monodocumentos obtenidos de forma automática, se obtuvo que la media de la calidad de éstos fue de un 40%, demostrando que

pueden comportarse como cualquier ser humano. La calidad de los resúmenes multidocumento depende de la medida en que los documentos a evaluar estén relacionados. La tasa de aceptación por parte de los jueces humanos fue de un 72.5%, resultado que puede calificarse de bueno.

Conclusiones

Como resultado de este trabajo la Universidad contará con una biblioteca de clases implementada en el lenguaje de programación Python, que posibilita la generación automática de resúmenes extractos, cumpliéndose de esta forma el objetivo general planteado. Asimismo, este trabajo contribuirá a aumentar la información existente del tema en el idioma español, del cual aparece una cantidad reducida, así como del código fuente de la biblioteca. Los resultados obtenidos de la implementación y las pruebas de los algoritmos confirmaron que éstos funcionaban bien.

Recomendaciones

1. Que en trabajos posteriores se pueda trabajar con cualquier tipo de componente textual y no solamente con ficheros txt, dígase: Correos electrónico, artículos de listas de discusión, postscript o documentos HTML.
2. Desarrollar un reconocedor de idioma para procesar el texto.
3. Implementar un analizador sintáctico más robusto.
4. Aumentar la cantidad de algoritmos implementados.
5. Implementar algoritmos para contextos específicos para hacer un análisis posterior de la variación de la arquitectura.
6. Añadir las palabras de parada omitidas a la representación de palabras de parada utilizadas.

Referencias bibliográficas

ANAYA, H.; A. PONS, et al. Una panorámica de la construcción de extractos de un texto. Revista Cubana de Ciencias Informáticas, 2006. 1: 55.

ANGHELUTA, R.; R. MITRA, et al. K U Leuven summarization system at DUC 2004, Interdisciplinary Center for Law & IT, 2004.

ARNTZ, R. and H. PICHT. Introducción a la terminología 1995. 382 p.

BALDWIN, B. and T. S. MORTON. Dynamic Coreference-Based Summarization. 1998. p.

BARZILAY, R. and M. ELHADAD. Using Lexical Chains for Text Summarization, [Web]. 1999. [Disponible en: http://www.cs.bgu.ac.il/~elhadad/lexical-chains.pdf

BEAUGRANDE, R. D.; W. U. DRESSLER, et al. Introducción a la linguïstica del texto. 1997. p. 8434482150

BELLAACHIA, A. and A. MAHAJAN. Information Retrieval and Data Mining Techniques for Generic text Summarization, 2005. [Disponible en:

BERRY, M. W. Survey of Text Mining. Clustering, Classification, and Retrieval., 2004. 272 p. 0387955631

BORKO, H. and C. L. BERNIER. Abstracting Concepts and Methods. Academic Press, 1975. 250 p. ISBN 0121186504

CARCEDO, F. J. A. La última revolución de la información, [Web]. www.elmundo.es, 2000. [2007]. Disponible en: http://www.elmundo.es/nuevaeconomia/2000/NE060/NE060-12b.html

COVINGTON, M. A. Natural Language Processing for Prolog Programmers. Artificial Intelligence Programs. 1994. p.

DELORT, J. Y.; B. B. MEUNIER, et al. Web Document Summarization by Context, [web]. 2007]. Disponible en: http://www2003.org/cdrom/papers/poster/p206/p206-delort.html

DONAWAY, R. L.; K. W. DRUMMEY, et al. A comparison of rankings produced by summarization evaluation measures. Association for Computational Linguistics, 2000. 69 - 78 p.

EDMUNDSON, H. P. New Methods in Automatic Extracting, [web]. 1969. [Disponible en: http://courses.ischool.berkeley.edu/i256/f06/papers/edmonson69.pdf

FUKUMOTO, J. Text summarization based on itemized sentences and similar parts detection between documents, [web]. Ritsumeikan University, 2003. [2007]. Disponible en: http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings3/NTCIR3-TSC-FukumotoJ.pdf

GARCÍA, L. A. Modelo para el agrupamiento de documentos afines y su ulterior resumen a través de la representación espacio vectorial de un corpus textual. Villa Clara Universidad Central “Marta Abreu” de Las Villas, 2005. 132. p.

GOLDSTEIN, J.; M. KANTROWITZ, et al. Summarizing Text Documents: Sentence Selection and Evaluation Metrics, 1999. [Disponible en: http://citeseer.ist.psu.edu/correct/197022

GOLDSTEIN, J.; V. MITTAL, et al. Multi-Document Summarization By Sentence Extraction, 2000. [Disponible en: http://courses.ischool.berkeley.edu/i256/f06/papers/goldstein00

HAHN, U. and I. MANI. The Challenges of Automatic Summarization. IEEE Computer Society Press 2000. p. 0018 9162

HEARST, M. A. Untangling Text Data Mining. School of Information Management & Systems University of California, Berkeley, 1999.

JACKSON, P. and I. MOULINIER. Natural Language Processing for Online Applications: Text Retrieval, Extraction, and Categorization. 2002. p. 1588112500

KODRATOFF, Y. Knowledge discovery in texts: A definition, and applications., 1999.

KUPIEC, J.; J. PEDERSEN, et al. A Trainable Document Summarizer. 1995. p. en este hay otro algoritmo.... 0-89791-714-6

LIN, C. Y. Training a selection function for extraction. 1999. p. puede que tenga un algoritmo... 1-58113-146-1

LIN, C. Y. and E. HOVY. Identify Topics by Position. Information Sciencies Institute, 1997. p.

LÓPEZ, M. J. M. Generación automática de resúmenes de texto para el acceso a la información.: departamento de Informática, Universidad de Vigo, 2003. 209. p.

LÓPEZ, M. J. M.; M. D. B. RODRÍGUEZ, et al. Diseño y evaluación de un generador de resúmenes de texto con modelado de usuario en un entorno de recuperación de información, 1998. [Disponible en:

LUHN, H. P. The Automatic Creation of Literature Abstracts*, 1958. [Disponible en: http://www.research.ibm.com/journal/rd/022/luhn.pdf

MANARIS, B. Z. and B. M. SLATOR. Interactive Natural Language Processing: Building on Success IEEE Computer Society Press 1996. p. 0018-9162

MANI, I. and E. BLOEDORN. Machine Learning of Generic and User-Focused Summarization, The MITRE Corporation, 1998. [Disponible en: http://arxiv.org/PS_cache/cs/pdf/9811/9811006v1.pdf

MANI, I.; B. GATES, et al. Improving Summaries by Revising Them, 2001. [Disponible en: http://acl.ldc.upenn.edu/P/P99/P99-1072.pdf

MANI, I. and M. T. MAYBURY Advances in Automatic Text Summarization Mit Press, 1999.

MARCU, D. A Decision-Based Approach to Rhetorical Parsing, 1999. [Disponible en: http://citeseer.ist.psu.edu/correct/288976

---. The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts. Canada, University of toronto, 1997. p.

MIHALCEA, R. and P. TARAU. TextRank: Bringing Order into Texts, 2004. [Disponible en: http://www.cs.unt.edu/~rada/papers/mihalcea.emnlp04.pdf

MOENS, M. F. Automatic Indexing and Abstracting of Document Texts. Kluwer Academic, 2000. p.

NOMOTO, T. and Y. MATSUMOTO. A new approach to unsupervised text summarization. ACM Press 2001. p. 1-58113-331-6

PORTILLA, J. A. D. R. Cómo averiguar a donde repercute la investigación científica: Para qué sirve la minería de textos

[web]. www.cie.unam.mx, 2005. [2007]. Disponible en: http://www.cie.unam.mx/~arp/mineria.html

RADEV, D. R.; H. JING, et al. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies, 2000. [Disponible en: http://arxiv.org/ftp/cs/papers/0005/0005020.pdf

ROCA, S. C. Sistemas de resumen automático de documentos. d´humanitats, 2001. 3.

SALTON, G. and M. MCGILL. Introduction to Modern Information Retrieval McGraw-Hill Companies, 1983. p. 978-0070544840

SJOBERGH, J. and K. ARAKI. Extraction based summarization using a shortest path algorithm. Disponible en: http://www.nada.kth.se/~jsh/publications/shortpath.pdf

TEUFEL, S. and M. MOENS. Sentence Extraction as a Classi¯cation Task., 1997. [Disponible en: http://citeseer.ist.psu.edu/correct/551628

WINOGRAD, T. Language As a Cognitive Process. Addison Wesley Pub. Co., 1983 p. 9780201085716

YANG, C. C. and F. L. WANG. Fractal Summarization for Mobile Devices to Access Large Documents on the Web, 2003. [Disponible en: http://www2003.org/cdrom/papers/refereed/p681/p681-yang-html/p681-yang.html

YAROWSKY, D. Word Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora, 1992. [Disponible en:

ZAMBRANO, H. Y. C. Una técnica para la extracción automática de resúmenes basada en una gramática de estilo. Facultad de Ingenieria, Universidad de los Andes, 2002. 97. p.

Glosario

Agrupamiento: Consiste en encontrar grupos de documentos que están relacionados por tópicos similares y extraer las palabras claves que son consideradas en esa clasificación (BERRY 2004).

Análisis de textos: Comprende todas las técnicas relacionadas con el análisis léxico, sintáctico y semántico de los textos (JACKSON and MOULINIER 2002).

Categorización: Es el proceso de clasificar los documentos por sus contenidos (BERRY 2004; JACKSON and MOULINIER 2002).

Clasificación: Se usa como un término más amplio que la categorización, para incluir cualquier asignación de documentos a clases, no necesariamente basados en el contenido (BERRY 2004; JACKSON and MOULINIER 2002).

Coherencia: Conectividad de contenido subyacente de un texto. La coherencia regula la posibilidad de que sean accesibles entre sí e interactúan de un modo relevante los componentes del mundo textual, es decir, la configuración de los conceptos y de las relaciones que subyacen bajo la superficie del texto (Beaugrande y Dressler, 1997).

Cohesión: Conectividad superficial de un texto. La cohesión representa la función comunicativa de la sintaxis que dirigen y mediatiza la operación de acceso a elementos lingüísticos. (Beaugrande y Dressler, 1997).

Concepto: Estructura de conocimiento (o contenido cognitivo) que el hablante puede activar o recuperar en su mente con mayor o menor unidad o congruencia (Beaugrande y Dressler, 1997).

Extracción de la información: Es el procesamiento de colecciones de textos ya seleccionados y para transformarlas en información que pueda ser comprendida y analizada más fácilmente (ZAMBRANO 2002).

Recuperación de información: Se encarga de recuperar documentos que puedan ser considerados relevantes para la tarea a desarrollar. Los usuarios del sistema pueden especificar conjuntos de documentos, pero el sistema debe ser capaz de filtrarlos y dejar fuera aquellos que se consideran irrelevantes [DIX97] [FRA92].

Resumen: Es el proceso de extraer conocimiento a partir de una fuente de información y presentar el contenido más importante al usuario en una forma condensada y sensitiva para las necesidades de la aplicación o del usuario (BERRY 2004; JACKSON and MOULINIER 2002).