Universidad de las Ciencias Informáticas

"Algoritmo para la Generación Automática de Resúmenes de un Documento HTML"

Tesis presentada en opción al título de Máster en Informática Aplicada

Autora: Lic. Isachi Abreu Gil

Tutores: Dr. Alcides Cabrera Campos
Dr. Efrén Vásquez Silva

Ciudad de La Habana, noviembre del 2009

"Año del 50 Aniversario del Triunfo de la Revolución"

Cuba

Agradecimientos:

Quiero agradecer a los Drs. Alcides Cabrera Campos y Efrén Vásquez Silva por su gran apoyo y ayuda incondicional.

A mis amigos Moe, Lianne, Yane, Puebla y Julio por estar siempre que los necesité.

A mis pipos y abuelita Bertha por quererme tanto.

A mi Jose por ser el mejor esposo del mundo y mi ejemplo de profesional.

A TODOS MUCHAS GRACIAS.

DECLARACIÓN JURADA DE AUTORÍA

Yo Isachi Abreu Gil, con carné de identidad 82111325034, declaro que soy el autor principal del resultado que expongo en el presente trabajo titulado “Algoritmo para la Generación Automática de Resúmenes de un Documento HTML”, para optar por el título de Máster en Informática Aplicada.

El trabajo fue desarrollado durante el período comprendido entre el 2008-2009 en colaboración de Dr. Efrén Vásquez Silva y Dr. Alcides Cabrera Campos, quienes me reconocen la autoría principal del resultado expuesto en esta investigación.

Finalmente declaro que todo lo anteriormente expuesto se ajusta a la verdad, y asumo la responsabilidad moral y jurídica que se derive de este juramento profesional.

Y para que así conste, firmo la presente declaración jurada de autoría en Ciudad de La Habana a los ___ días del mes de _____ del año _____.

<Firma del maestrante en tinta azul>

Resumen

En los últimos años el rápido crecimiento de Internet, ha traído consigo un vertiginoso aumento de la información disponible, en este sentido la Generación Automática de Resúmenes juega un papel de suma importancia. Los algoritmos encontrados en la literatura no hacen uso de la información de marcado accesible desde la propia página web, por lo que no tienen en cuenta información relativa a la intención del autor en el momento de crear el documento. En el presente trabajo se desarrolló un algoritmo para la Generación Automática de Resúmenes de páginas web, que utiliza información de marcado presente en el código HTML, se define una función para determinar la relevancia de un término en el contenido de un documento y se propuso un método para identificar el idioma. Para evaluar la calidad del algoritmo se aplicaron las métricas ROUGE-1, ROUGE-2, ROUGE-L y ROUGE-W y se compararon los resultados obtenidos con los sistemas comerciales Copernic Summarizer, Pertinence Summarizer y Swensun, obteniéndose resultados significativamente superiores en la métrica ROUGE-1 y sólo superado por el sistema Copernic Summarizer para el resto de las métricas.

Índice

Introducción

La *World Wide Web* es un sistema de información global que ha supuesto un nuevo modelo de colaboración e interacción entre individuos (Berners-Lee, et al., 1992). Medir el tamaño completo del contenido de la web es una tarea complicada, debido a su naturaleza dinámica. Sin embargo, es posible realizar estimaciones acerca del tamaño de la "web visible" (*visible/surface web*), es decir, el conjunto de páginas accesibles desde los diferentes motores de búsqueda. En (Bharat y Broder, 1998), los autores estimaron el tamaño de la web visible para *Hotbot*[1], *Altavista*[2], *Excite*[3], e *Infoseek*[4] - los mayores motores de búsqueda del momento– en 200 millones de páginas. Además, la intersección entre los documentos indexados por cada uno de estos buscadores era de menos de un 4 %, alrededor de 2.2 millones de páginas. En (Lawrence, S. y Giles, C, 1999), un año más tarde, se hablaba de 800 millones de páginas. Hoy en día, estas cifras resultan ridículas. En (O'Neill et al., 2009) puede encontrarse un estudio detallado de la evolución de los contenidos públicos en Internet en los últimos años, mostrando que el número de páginas web indexadas actualmente superan los 15.5 billones. De todas ellas, más de 10 billones eran accesibles con el motor de búsqueda Google[5]. En el caso del buscador MSN[6], los documentos indexados eran 5 billones, y el total de documentos accesibles por Yahoo[7] y ASK/Teoma[8], era de 4 y 2 billones respectivamente.

Gracias a los avances tecnológicos de las últimas décadas, el almacenamiento y acceso a la información ya no suponen un problema, pero el tiempo sigue siendo

[1] http://www.hotbot.com/
[2] http://www.altavista.com/
[3] http://www.excite.com/
[4] http://www.infoseek.com
[5] http:///www.google.com
[6] http://www.msn.com
[7] http://www.yahoo.com
[8] http://www.ask.com

un bien valioso y limitado. De poco sirve disponer de una inmensa cantidad de datos si no se es capaz de acceder a ellos de un modo realmente provechoso.

En este marco de sobrecarga de información, la Generación Automática de Resúmenes juega un papel de suma importancia. El propósito de los resúmenes es facilitar el procesamiento de información optimizando el tiempo de lectura necesario para localizar la información requerida. La descripción compacta del contenido relevante de un documento, puede permitir el incremento de la eficiencia en el procesamiento, recuperación y clasificación del material textual.

El proceso de generación automática de resúmenes de documentos consiste en, dada una fuente de información (uno o más documentos) y un demandante (usuario o aplicación), extraer el contenido de la fuente de información y presentarlo en forma comprensible, y en una manera tal que satisfaga sus necesidades (Mani, I., 2001).

Si todas las frases dentro de un documento tuvieran la misma importancia, la tarea de generar un resumen no sería muy efectiva, pues cualquier reducción en tamaño del documento significaría la pérdida de información importante (Esaú-Villatoro, Tello., 2007). Afortunadamente, la información relevante de un documento tiende a aparecer sólo en determinadas secciones, de esta forma un algoritmo adecuado será capaz de diferenciar entre frases que contengan más o menos información relevante.

Los primeros trabajos en generación automática de resúmenes de texto datan de finales de los 50 (Luhn, H., 1958) y de la década de los 60 (Edmundson, H., 1969). Durante las dos décadas siguientes el interés por este tema no fue muy importante. Sin embargo, a partir de los 90, especialmente en los últimos años, la investigación en el área creció de una forma significativa. Prueba evidente de ello son los numerosos trabajos publicados en estos años, libros editados sobre el tema (Endres-Niggemeyer, 1998; Mani, I., y Maybury, 1999; Marcu, D., 2000; Mani, I., 2001), varias tesis doctorales((Esaú-Villatoro, Tello., 2007),(Gallo, D, 2006), (Cunha, F., 2008)), números especiales que han dedicado algunas revistas

(Information Procesing &Management (Sparck-Jones, K.y Endres-Niggemeyer, 1995), Computational Linguistics (Radev, D. et al., 2002) y Artificial Intelligence in Medicine (Spyropoulos y Karkaletsis, 2003)), talleres que varias conferencias relevantes han dedicado a la generación de resúmenes de texto (ACL/EACL (Mani y Maybury, 1997), ANLP/NAACL'00 (Hahn et al., 2000), NAACL'01(Goldstein y Lin, 2001), SIGIR/DUC'01 (Harman y Marcu, D, 2001) y ACL/DUC'02 (Harman,2002)), o las iniciativas puestas en marcha para la evaluación independiente de este tipo.

Sin querer realizar una lista exhaustiva de los sistemas comerciales existentes, sí se pueden mencionar algunos de ellos. Entre los que parecen estar principalmente pensados para grandes bases de datos se tienen a *Inxight MetaText Server*[9] de Inxight Software, Inc., *Oracle9i Text*[10], *Intelligent Miner for Text*[11] de IBM o *NetOwl Summarizer*[12] de SRA Corporation. En un segundo grupo, dedicado al ámbito de oficina, se puede mencionar sistemas como *MS-Word*(*AutoSumarize)* de Microsoft, *Pertinence Summarizer*[13] o *Copernic Summarizer*[14] de Copernic Technologies Inc. Todos los algoritmos que implementan son privativos.

Son muchas las aplicaciones comerciales disponibles relacionadas con la generación de resúmenes, lo que da una idea, tanto del interés, como de la necesidad que existe de este tipo de herramientas. Todas son sistemas de propósito general o sea aplicables a cualquier clase de documento.

En Cuba los sistemas de extracción automática de resúmenes aun constituyen proyectos en estado de gestación. Las investigaciones en este campo no son abundantes y las necesidades actuales se cubren con sistemas privativos, que no siempre de ajustan a las necesidades reales.

9 http://www.inxight.com/products/ims/.
10 http://technet.oracle.com/products/text/content.html.
11 http://www-3.ibm.com/software/data/iminer/fortext/.
12 http://www.netowl.com/products.html.
13 http://www.pertinence.net/index_en.html
14 http://www.copernic.com/en/products/summarizer/.

En el ámbito de Internet, la comunicación por medio de páginas web se puede considerar como un proceso informativo-documental (Fresno, V., 2006). En este contexto, Internet representa el medio y HTML el código por el cual un emisor (el autor de una página web) codifica un mensaje (el contenido de la propia página) que posteriormente será visualizado por un receptor.

HTML es un lenguaje de marcado que tienen la función principal de indicar a los navegadores web el modo en que deben mostrar el contenido al usuario (Musciano, C. y Kennedy, B, 2000). Si bien dispone de algunas etiquetas capaces de indicar la estructura del documento, posee otras capaces de destacar zonas dentro del contenido. Entonces se puede pensar que, así como se espera que el título de un documento aporte información sobre su contenido, las partes enfatizadas responden también a la intención del autor (Cerezo, A., 1994), convirtiendo a estas señales accesibles desde el código HTML, en información valiosa a la hora de realizar un resumen de la misma.

Lamentablemente los algoritmos para la Generación Automática de Resúmenes encontrados en la literatura no hacen uso de la información de marcado accesible desde la propia páginas web, que pudiera extrapolarse a información relativa a la intención del autor en el momento de crear el documento, intuyendo qué partes quiso destacar frente a otras o con qué elementos del discurso quiso llamar la atención del lector. La mayoría de los algoritmos se basan en técnicas estadísticas y lingüística, convirtiendo a los documentos en entidades matemáticas.

El empleo de la información de marcado, permitiría no utilizar información externa a la página que se quiere resumir, logrando que los algoritmos sean completamente independientes del tamaño actual y futuro de la web. Además, podría aplicarse en sistemas sin necesidad de contar con enormes medios de almacenamiento ni de procesamiento, evitando una exploración intensiva de colecciones de documentos correlacionados.

En este sentido se formula el siguiente **problema científico:**

¿Cómo aprovechar la información de marcado presente en los documentos HTML en la Generación Automática de Resúmenes?

Se presenta como **objeto de estudio** la Generación Automática de Resúmenes y como **campo de acción** la Generación Automática de Resúmenes en documentos HTML.

El **objetivo general** de esta investigación es el que sigue:

Desarrollar un algoritmo para la Generación Automática de Resúmenes que utilice la información de marcado presente en los documentos HTML.

Planteándose la siguiente **idea a defender**:

Es posible desarrollar un algoritmo para la Generación Automática de Resúmenes que emplee la información de marcado contenida en los documentos HTML.

Este objetivo general se articula en los siguientes objetivos específicos:

- Valorar características de algoritmos para la Generación Automática Resúmenes presentes en la literatura.
- Definir una función para determinar la relevancia de un término dentro del contenido de un documento HTML.
- Desarrollar un algoritmo para la identificación del idioma de un documento.
- Diseñar un algoritmo para la Generación Automática de Resúmenes en documentos HTML
- Comparar el nivel informativo presente en los resúmenes generados por el algoritmo desarrollado con resúmenes generados por sistemas comerciales de popularidad.

Aporte y Novedad de la investigación:

A partir de la presente investigación **se contará** con un algoritmo que podrá ser empleado por sistemas que requieran obtener de forma automatizada el resumen de una página web. Las características de dicho algoritmo permitirán que pueda ser aplicado a documentos de cualquier dominio temático y para múltiples idiomas.

Lo **novedoso** de la investigación está reflejado en el empleo de la información de marcado presente en las páginas web en la Generación Automática de Resúmenes, lo que evita recurrir a técnicas complejas para determinar la relevancia de las frases del documento.

Métodos de investigación:

Análisis y síntesis: Este método fue utilizado para a partir de la situación problemática determinar una variante de solución.

Experimental: Con el empleo de este método se logró la evaluación extrínseca del método propuesto. Además permitió la realización de experimentos para determinar umbrales, coeficiente de importancias, etc.

Medición: A partir de mediciones se pudo realizar evaluaciones al algoritmo propuesto.

Organización de la tesis:

El documento se encuentra organizado en tres capítulos tal como se describe a continuación:

Capítulo 1: *Preliminares*. En este capítulo se exponen los elementos fundamentales relacionados con la Generación Automática de Resúmenes. Se presentan características de los algoritmos de Generación de Extractos. Se exponen métodos para la evaluación y revisión de resúmenes generados. Además

se presentan elementos fundamentales del lenguaje HTML. Se realiza un análisis de los principales trabajos realizados en la Generación Automática de extractos.

Capítulo 2: *Algoritmo para la Generación Automática del Extracto de un documento HTML.* En este capítulo se presentan criterios heurísticos para determinar la relevancia de los términos dentro de un documento HTML y se define una función de relevancia. Se presenta un método para determinar el idioma de un documento. Se desarrolla un algoritmo para generar el extracto de un documento HTML.

Capítulo 3: *Evaluación*. En este capítulo se realiza la evaluación de la calidad del algoritmo desarrollado. Se presentan comparativas entre los resúmenes generados por el algoritmo propuesto y los generados por algunos sistemas comerciales empleándose cuatro métricas de evaluación.

Finalmente se presentan las Conclusiones, Recomendaciones, Referencias Bibliográficas, Siglario y el Glosario de Términos.

Capítulo 1. Preliminares

En este capítulo se introducen conceptos relacionados con la Generación Automática de Resúmenes, se analizan las principales características de los Algoritmos para la Generación Automática de Extractos. Se presentan métodos para la revisión de extractos, así como los principales métodos usados en la evaluación de los resúmenes. También se realizó un estudio sobre los algoritmos de generación de extractos presentes en la literatura. Se exponen elementos a tener en cuenta a la hora de representar un documento HTML y algunas de las funciones de ponderación presentes en la literatura.

1.1 Panorámica sobre la Generación Automática de Resúmenes

El proceso de la Generación Automática de Resúmenes de documentos consiste en, dada una fuente de información (uno o más documentos) y un demandante (usuario o aplicación), extraer el contenido de la fuente de información y presentarlo en forma comprensible, y en una manera tal que satisfaga sus necesidades (Mani, I., 2001).

Los resúmenes generados de forma automática pueden constituir tipos especiales de resúmenes, y son clasificados según su:

> **Audiencia**: Dependiendo del tipo de usuario al que está destinado el resumen, puede ser clasificado en perteneciente a la clase de los resúmenes genéricos, o bien a la clase de los resúmenes que pueden ser enfocados a un usuario, a un tópico o a una consulta (Mani, I., 2001). Los resúmenes genéricos son aquellos que están destinados a una amplia comunidad de lectores. Los resúmenes enfocados a un usuario (o a un tópico o a una consulta) están dirigidos a satisfacer a un usuario o a un grupo particular de éstos. Para la construcción de estos últimos resúmenes

se deben tener en cuenta, además del contenido de la fuente, los intereses del usuario (que son expresados generalmente a través de una consulta). Constituyen ejemplos de resúmenes enfocados a un usuario aquellos que son mostrados por los buscadores de Internet para cada documento recuperado.

Función: Un resumen, de acuerdo a su función, puede ser informativo o indicativo (Mani, I., 2001). Un resumen informativo cubre toda la información relevante de un texto fuente con un determinado nivel de detalle, mientras que un resumen indicativo no lo hace necesariamente. Los resúmenes indicativos son usados para condensar textos de poca estructuración y gran extensión tales como editoriales, ensayos, libros, etc., y pueden suministrar al usuario una función de referencia que permita seleccionar documentos o partes de estos para una lectura más profunda. Por ejemplo, un resumen indicativo de un reporte de investigación científica incluiría el ámbito y propósito del mismo, pero no los resultados ni las recomendaciones; sin embargo, un resumen informativo debe tener en cuenta todos estos aspectos. Una caracterización importante de los resúmenes informativos es que a partir de ellos se puede reconstruir por completo el texto de la fuente y sirven, en muchas ocasiones, como sustitutos del mismo. Los resúmenes indicativos proporcionan un indicio, también a determinado nivel de detalle, del contenido de un texto fuente y ayudan al lector a decidir si leer o no el texto completo. La clase de los resúmenes informativos no es disjunta de la clase de los resúmenes indicativos, pues todo resumen informativo tiene también función indicativa. Por tanto, la clase de los resúmenes informativos es un subconjunto propio de la clase de los resúmenes indicativos.

Otros términos de importancia en el área de la construcción automática de resúmenes son: **la razón de compresión** de un resumen (también conocido como razón de condensado) y el término **resumen de referencia** de un texto fuente:

- La razón de compresión de un resumen ***r*** de una fuente ***t*** se define como el cociente entre la longitud de ***r*** y la longitud de ***t*** y es un número real perteneciente al intervalo (0, 1). Ambas longitudes deben estar expresadas en la misma unidad de medida, que por lo general es la cantidad de palabras del texto. Como convención, debido a la cantidad de texto que es excluida del resumen, una razón de compresión cercana a 0 se considera alta, mientras que una cercana a 1 se considera baja.
- Para un texto fuente pueden establecerse uno o varios resúmenes estándares, los que se denominan resúmenes de referencia. Estos resúmenes se construyen total o parcialmente por personas y su utilidad se verá más adelante.

Un número de nociones básicas de la Generación Automática de Resúmenes tienen que ver con la relación entre un resumen y su fuente. De esta manera se puede hacer una distinción fundamental entre resúmenes que son abstractos y resúmenes que son extractos.

Un extracto es un resumen que consiste enteramente en material copiado de su fuente. Formalmente, en términos de la definición de construcción automática de resúmenes dada anteriormente, un extracto es un resumen obtenido mediante la aplicación de operaciones de selección al contenido de una fuente. Una operación de selección se puede definir a través de una función cualquiera f que cumple que, si $X = e_1 e_2 \cdots e_n$ es un texto, donde para todo $\kappa \in \{1,\ldots, n\}, e_k$ es un elemento de X, entonces existe un texto $Y = e_{i1} e_{i2} \cdots e_{im}$ tal que $f(N) = Y$ y existen $j_1 \cdots j_m$ tales que $1 \leq j_1 \leq \ldots \leq j_m \leq n$ y $\{i_1 \ldots i_m\} = \{j_1 \ldots j_m\}$. Los elementos de un texto pueden ser palabras, frases, cláusulas, sentencias (oraciones), párrafos, discursos o, incluso, documentos.

Los abstracto, son generados a partir de los extractos. El proceso de generación involucra crear nuevas oraciones a partir de las que han sido extraídas en un

primer paso por el sistema de generación de extractos. Para poder crear estas nuevas oraciones es necesario contar con sofisticados recursos lingüísticos que interpreten adecuadamente contenido y significado de las oraciones extraídas. Una vez hecha esta interpretación el sistema puede mezclar y/o comprimir oraciones con el objetivo de entregar al usuario un resumen más coherente. Las técnicas necesarias para la aplicación de esta estrategia distan de haber obtenido resultados satisfactorios y pertenecen aún al campo de la investigación básica.

Los algoritmos de generación automática de resúmenes, se pueden considerar funciones que computan un resumen dado una fuente, se pueden clasificar en dos grandes grupos: los de **estrategia poco profunda ó superficiales** y los de **estrategia profunda** (Mani, I., 2001). Esta clasificación se realiza en dependencia de los niveles de análisis lingüístico que emplean y de los tipos de elementos del texto sobre los que operan.

Existen cuatro niveles de análisis lingüístico, éstos son (por orden de complejidad de menor a mayor): morfológico[15], sintáctico[16], semántico[17] y discurso[18].

Los algoritmos de estrategia poco 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, éstas pueden ser analizadas a un nivel semántico. Por su parte, los algoritmos de estrategia profunda realizan el análisis al menos a nivel semántico y los elementos del texto sobre los que operan no son menos complejos que las frases.

De manera general, los algoritmos de estrategia poco profunda producen extractos y son robustos, mientras que los de estrategia profunda generan abstractos y son poco generales, o sea, se aplican a fuentes de un dominio específico (por ejemplo, Química, Medicina, Biología, Física, Psicología, etc.).

---

[15] El análisis de las palabras para extraer raíces, rasgos reflexivos, unidades léxicas compuestas y otros fenómenos.

[16] El análisis de la estructura sintáctica de la frase mediante una gramática de la lengua en cuestión.

[17] La extracción del significado de la frase, y la resolución de ambigüedades léxicas y estructurales.

[18] El análisis a nivel de texto, argumentación, narración, se tienen en cuanta los temas de la coherencia local y global, los pronombres, el estilo, etc.

La mayoría de los algoritmos existentes de construcción de extractos seleccionan elementos de un mismo tipo para componer el extracto, casi siempre sentencias. Esto último, se debe a que seleccionar palabras produce extractos bastante incoherentes y la selección de párrafos ocasiona muchos problemas con la razón de compresión.

Se considera que las sentencias son elementos lingüísticos que, por lo general, expresan proposiciones o ideas semánticamente completas y por lo que la selección de ellas contribuiría con la coherencia de los extractos generados.

La creación del resumen mediante técnicas de extracción, pese a parecer simplista, goza de cierta justificación. Aproximadamente el 80% de las frases en resúmenes creados por humanos están copiadas tal cual o con pequeñas modificaciones a partir del texto original (Mani, I., 2001). Por otro lado, el hecho de que los algoritmos de generación de abstractos sean aplicables a determinados dominios temáticos debido a la dependencia de grandes cantidades de recursos lingüísticos, quedan exentos de la presente investigación. Dando paso sólo al estudio de los algoritmos de generación de extracto. De ahora en adelante siempre que se hable de *resumen automático* se estará haciendo referencia a los creados usando técnicas de extracción.

1.1.1 Características de los algoritmos de generación de extractos.

El proceso de la generación automática de extractos se puede dividir en tres fases: la fase de ***análisis***, la fase de ***transformación*** y la fase de ***síntesis,*** (Sparck-Jones, K, 1999) dispuestas en ese mismo orden. Durante la primera fase se realiza un análisis de la fuente y se construye una representación interna de la misma. En la fase de transformación se traduce la representación interna obtenida en la fase de análisis a la representación interna del resumen por medio de operaciones de selección. Por último, la fase de síntesis transforma el extracto de su representación interna a una representación en lenguaje natural. Los límites

entre estas tres fases son muy difusos, y en la práctica ocurre que algunas de ellas se mezclan o se suprimen.

En los algoritmos de construcción de extractos, el problema de la selección de los elementos de un texto muchas veces se reduce a un problema de clasificación, donde los elementos se clasifican en pertenecientes al extracto o no.

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. Precisamente por reducirse a problemas de clasificación, los algoritmos de construcción de extractos pueden ser supervisados o no supervisados.

Los algoritmos supervisados 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 (Figura 1). Debido al uso de tal colección de entrenamiento, por lo general, estos algoritmos obtienen resúmenes de textos de materias específicas.

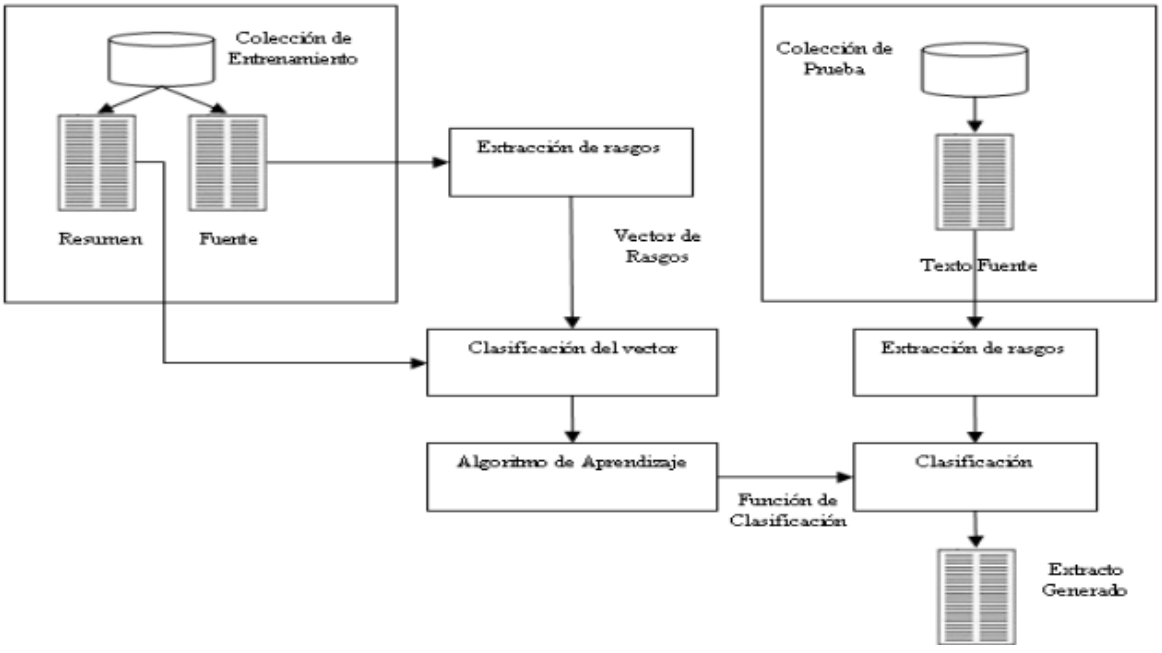

Figura 1: Esquema de los Algoritmos Supervisados

Los algoritmos no supervisados siguen básicamente dos esquemas. El primero consiste en ponderar cada elemento del texto individualmente, considerando propiedades intrínsecas de éstos en el texto (estadísticas y lingüísticas), para luego clasificar los elementos de mayor peso en elementos del extracto. Los algoritmos que siguen el segundo esquema usan el nivel de discurso del texto para construir, a partir de los elementos del texto y sus relaciones, un grafo que luego es usado para clasificar y extraer los elementos que formarán parte del extracto.

Cada una de estas estrategias tienen sus ventajas y limitantes, para el caso de los algoritmos que usan técnicas supervisadas, tienen como limitante que al depender de una colección de entrenamiento imposibilita su empleo en tareas donde la naturaleza de los documentos a resumir sea heterogénea, ya que, la tenencia de colecciones de entrenamientos generales (cualquier dominio temático e idioma) es un proceso muy complejo y prácticamente imposible. Pero al mismo tiempo la tenencia de dicha colección, permite que en ambientes específicos, los resúmenes generados sean más eficientes y satisfagan mejor la expectativa de quien los necesita. Por su lado, los algoritmos no supervisados pueden ser empleados sobre documentos de cualquier dominio temático y diferentes idiomas con pocas modificaciones, pues no necesitan de conocimiento previo para realizar la clasificación, lo cual resulta en ocasiones una desventaja, pues muchos de los algoritmos que usan estas técnicas se vuelven tan superficiales que los extractos generados son un total caos.

1.1.2 Revisión

Una vez creado un extracto se le pueden aplicar métodos de revisión, como parte de la fase de síntesis, para mejorar su coherencia y contenido informativo. El problema es que los sistemas de construcción automática de resúmenes, tienden

a generar textos incoherentes, debido a la presencia de anáforas[19], ambientes estructurados, en los textos fuentes.

Los humanos, por lo general, tienen la capacidad de revisar resúmenes, ya sean construidos por computadoras o por los propios humanos, para mejorar su consistencia, su fluidez, su coherencia. Esto lo logran, por lo general, mediante el uso de operaciones de "copiado y pegado" de elementos del texto. Estas operaciones se catalogan como revisión local (cuando son efectuadas dentro de una misma sentencia) o revisión global (cuando los elementos implicados son, o pertenecen a, sentencias distintas). Las mismas tratan de deshacerse de términos vagos o redundantes, realizar sustituciones léxicas, componer varias sentencias.

En el área de construcción automática de extractos existen dos métodos de revisión fundamentales, ellos son: el ajuste de la coherencia y la revisión completa de los extractos (Mani, I., 2001).

Ajuste superficial de la coherencia.

Como se vio anteriormente, cuando se extraen sentencias de una fuente, éstas pueden perder su contexto, originándose muchas veces con esta acción un resumen incoherente. Esto evidentemente es un problema y sus soluciones son diversas, pero tienen en común que generalmente actúan de modo superficial. Conocer si una anáfora ha perdido su referente requiere identificar a este último en el texto. Esto es generalmente bastante difícil ya que se requieren tanto conocimientos lingüísticos como conocimiento del dominio temático. Como resultado algunos sistemas, por ejemplo: en (Miike, S. at al., 1994) simplemente se excluyen todas las sentencias que contengan anáforas. Otra estrategia similar es la de eliminar la anáfora si la sentencia previa no está en el resumen siguiendo dos caminos posibles: el primero es incluir la sentencia en la que se encuentra el antecedente de la anáfora en cuestión ó, resolver la anáfora sustituyéndola por el

[19] **anáfora** es un elemento gramatical no referencial que requiere un antecedente en un dominio sintáctico local.

valor del antecedente. Estas estrategias pueden traer como inconveniente la pérdida de sentido del texto si no se determina de manera correcta el antecedente.

Los ambientes estructurados, tales como elementos de una lista, tablas, o argumentos lógicos, cuya integridad estructural necesita ser preservada en el resumen, presentan retos similares. A menudo es muy difícil analizar la estructura de alguno de estos elementos, pero puede ser simple el reconocer que se está en presencia de uno. En cualquiera de los casos, se tiene la opción de reconocerlos y excluirlos.

Revisión completa.

La revisión completa consiste en la revisión local y global de las sentencias mediante la aplicación sucesiva de operaciones de eliminación de sus componentes y agregación sintáctica de las mismas, respectivamente. Entre las operaciones más comunes de revisión local se encuentran la eliminación de expresiones de una sentencia que ocurren dentro de paréntesis y la eliminación de frases tales como ”En particular”, ”En conclusión”, etc. La revisión completa además de mejorar la coherencia de un resumen también intenta elevar su nivel de información. En (Mani, I., 2001) se describe un método de este tipo. Ellos construyen un borrador inicial de un sumario partiendo de un documento fuente y luego le añaden información adicional de soporte de la misma fuente. En vez de concatenar material en el borrador, la información presente es combinada y suprimida basada en reglas de compactación que involucran operaciones de agregación y eliminación de sentencias. Un aspecto muy importante a la hora de aplicar estos métodos de revisión es que muchos de ellos necesitan del conocimiento de la estructura sintáctica e incluso de la estructura de discurso de los extractos. Además, como se menciona anteriormente, en la mayoría de los sistemas las operaciones de revisión constituyen operaciones de post-procesamiento. De esta manera estas operaciones pueden implicar inconsistencias con las restricciones de la razón de compresión de los resúmenes que se generan.

1.1.3 Evaluación de la calidad de los resúmenes

Es deseable que un resumen sea coherente, conciso, de fácil lectura y comprensión, y que ofrezca información relevante al usuario. Uno de los primeros pasos en la formalización de la evaluación de resúmenes se dio en (Mani, I., 2001), donde se introdujo la división de los métodos de evaluación de resúmenes construidos de manera automática en métodos intrínsecos y extrínsecos.

Los **métodos intrínsecos** son aquellos que evalúan a los resúmenes como entes individuales, generalmente comparándolos con un resumen de referencia. Estos se subdividen en dos grupos, los que evalúan la calidad (como obra textual) y los que evalúan el contenido informativo de los resúmenes. Los métodos intrínsecos pueden tener en cuenta:

La coherencia del resumen. Algunos elementos de un resumen sufren la pérdida del contexto en el que ocurren en la fuente, acarreando esto problemas de coherencia tales como referencias sin resolver y fisuras en la estructura del discurso del resumen. De aquí que un resumen se pueda evaluar según su coherencia. La coherencia de un resumen se puede definir, por ejemplo, a partir del número de anáforas sin resolver y el número de ambientes estructurados, tales como listas y tablas, no preservados correctamente en su texto (Sparck-Jones, K., 1995.). Otra medida de coherencia (Piat, G, et al., 1997) se basa en un algoritmo de aprendizaje supervisado, que clasifica las sentencias en coherentes o no.

La precisión y relevancia del resumen. Las medidas de precisión, relevancia y la F-medida (Lita, L. et al., 2001) son importadas del área de Recuperación de Información. Como medidas de evaluación de resúmenes se aplican sólo a extractos que estén constituidos por sentencias del texto fuente, y necesitan de un resumen de referencia con esta misma característica. Se define como:

$$precisión = \frac{|E \cap R|}{|E|} \quad relevancia = \frac{|E \cap R|}{|R|} \quad F - medida = \frac{2 * precisión * relevancia}{precisión + relevancia}$$

donde E denotan el conjunto de sentencias del extracto que se evalúa y R el conjunto de sentencias del resumen de referencia.

Los *n*-gramas[20] del resumen. Un resumen puede ser evaluado comparando su contenido con el de un resumen de referencia o con el de su texto fuente (Drummey, K. et al., 2000). La comparación del contenido de dos textos puede realizarse usando una medida de solapamiento de vocabularios, como el Coeficiente de Dice o la medida del coseno (Salton, G. y McGill, M., 1987). Si en esta evaluación está involucrado un abstracto, debe usarse algún tesauro de términos a la hora de representar los textos.

Otras medidas de evaluación intrínsecas se pueden encontrar en (Drummey, K. et al., 2000).

Los **métodos extrínsecos** evalúan la eficiencia y el desempeño de los resúmenes en una tarea determinada. Éstos exigen casi siempre una activa participación de personas. Uno de los más sencillos es el de **Lectura de comprensión** (Mani, I., 2001). 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 (Brandow, R., et al., 1995), éste puede ser utilizado también para evaluar el contenido informativo de un resumen.

En (Hovy. E., 1999) se describen tres tipos de métodos extrínsecos que permitirían obtener medidas del grado de retención. Uno de estos métodos se trata de los denominados juegos de Shannon, de la Pregunta y de Clasificación (Shannon, C., 1951). En todos ellos era necesario recurrir a sujetos humanos que debían llevar a cabo una tarea que requería el conocimiento previo del texto original.

Por ejemplo, en el caso del juego de Shannon los sujetos debían reconstruir el documento original de manera literal; algunos de los participantes habían tenido

[20] Una *n*-grama es una secuencia de n palabras consecutivas de un texto.

acceso al mismo mientras que otros sólo habían leído el resumen. En todos los casos se informaba a los sujetos cuando se equivocaban en una letra y se les permitía un nuevo intento; la relación entre el número de intentos requeridos en ambos grupos permitía calcular el nivel de retención.

Este tipo de experimentos son enormemente costosos en tiempo y recursos y, por otro lado, evalúan la "calidad" de los resúmenes de manera indirecta a través de su influencia en la ejecución de una o más tareas.

Una medida de evaluación produce para un resumen un valor numérico que por sí solo no significa nada; pero dicho valor puede ser utilizado en conjunto con los valores obtenidos de medir (con la misma medida de evaluación) otros resúmenes de la misma fuente para establecer un orden entre los distintos resúmenes.

Con el objetivo de comparar la calidad de los algoritmos de construcción de resúmenes cada año se realiza una competición internacional que se conoce con el nombre de DUC[21] (Document Understanding Conference).

El método de evaluación más utilizado hoy en día es el que emplea el sistema ROUGE (Lin, C. y Hovy, E., 2003) el cual compara el resumen que se desea evaluar (resumen candidato) con resúmenes creados por humanos (resúmenes modelo o de referencia). La métrica que subyace al método se basa en la co-ocurrencia de n-gramas entre los resúmenes candidatos y los resúmenes modelo, existiendo diversas variantes de la misma.

Otro de los métodos de evaluación automáticos más conocidos es el *Pyramid Method* desarrollado por (Nenkova, A, et al., 2004), que parte de la idea de que no hay un único modelo de resumen ideal y, por tanto, también compara el resumen que se quiere evaluar con varios resúmenes humanos.

Un método de evaluación automático de resúmenes también relevante es el denominado *Basic Elements* (Hovy, E. et al., 2005) el cual, como los dos

[21] http://duc.nist.gov/

anteriores, compara el resumen candidato con resúmenes modelo realizados por humanos.

Este método divide cada oración de los resúmenes en un conjunto de unidades semánticas mínimas llamadas *Basic Elements (Bes).*

Finalmente, se referencia a un novedoso sistema de evaluación de resúmenes: QARLA ((Amigó, E., et al., 2005) (Amigó, E., 2006)). Este sistema tiene como componentes una serie de resúmenes modelo generados manualmente (ofrecidos por el usuario), una serie de resúmenes modelo generados automáticamente (también ofrecidos por el usuario) y una serie de métricas de similitud (ofrecidas por el sistema).

En concreto, el sistema QARLA puede aportar al usuario varias medidas:

1. una medida para evaluar la calidad de conjuntos de métricas de similitud
2. una medida para evaluar la calidad de un resumen utilizando un conjunto adecuado de métricas de similitud,
3. una medida para constatar si la serie de resúmenes automáticos del modelo es fiable o no.

Para concluir es bueno mencionar que el problema de cómo evaluar un resumen construido de manera automática, al igual que la construcción automática de resúmenes, es un problema que aún no se ha cerrado.

1.2 El vocabulario HTML

HTML es un lenguaje creado en 1989 por Tim Berners-Lee. Fue concebido con el fin de visualizar e interconectar el contenido de documentos electrónicos, por lo que consideró un conjunto pequeño de etiquetas que marcaran párrafos, títulos, hipervínculos y un poco más (Fresno, V., et al., 2006). A continuación, se asociaron comportamientos concretos a dichas etiquetas.

Con el tiempo, la ventaja que representaba la simplicidad de HTML se convirtió en un inconveniente, ya que su marcado no siempre cubría todos los aspectos de

presentación que los usuarios requerían. La solución adoptada fue el desarrollo de extensiones del lenguaje privadas, lo que complicó la estandarización. Las luchas comerciales entre las principales empresas de desarrollo de navegadores web durante los primeros años de Internet condujeron a un lenguaje HTML que, aunque universalmente utilizado e interpretado, carece de una estandarización real. HTML cumple con los dos objetivos esenciales para el diseño y visualización de un documento digital:

- Estructura un documento en elementos lógicos, como por ejemplo: encabezado, párrafo, etc.
- Especifica las operaciones tipográficas y funciones que debe ejecutar un programa visualizador sobre dichos elementos. Aunque deba considerarse como un lenguaje de marcado híbrido, su uso está orientado principalmente a la descripción de operaciones tipográficas (Musciano, C. y Kennedy, B., 2000); por tanto, se trata de un lenguaje con un carácter esencialmente procedimental.

En general, un documento HTML sigue la sintaxis de cualquier lenguaje de marcado y su estructura global es la siguiente. A partir de un elemento raíz <html> se pueden anidar otros dos elementos: <head> y <body>, correspondientes a la cabecera y cuerpo del documento. Un ejemplo sencillo de documento HTML podría ser el que sigue:

```
< html >
   < head >
      < title >  Título de la página  < /title >
   < /head >
   < body >
      < h1 > Título del contenido visible < /h1 >
        texto visible
    < font color = "#000080" > texto en diferente color < /font >
```

texto visible

< /body >

< /html >

Cabecera (<head>)

En la cabecera se incluyen definiciones generales a todo el documento; se puede agregar un fondo de pantalla, definir los colores del texto, etc. El texto contenido en este elemento <head> no resultará visible en un navegador web. Estas definiciones pueden estar relacionadas con el formato global del documento, para lo que se emplea la etiqueta <style>, o tratarse de características más cercanas a la visualización que el autor desea dar a cada elemento, y que podrían ser diferentes de las que establezca por defecto el navegador.

El elemento <title> debe ir también en la cabecera, especifica el título del documento y se muestra en la barra de título del navegador. El contenido de este elemento suele usarse como el texto con el que se guarda una página en los marcadores (bookmarks). También es el texto que muestra un motor de búsqueda en los enlaces devueltos tras una consulta. Este elemento es opcional, aunque sería muy recomendable que todo documento HTML tuviera un título.

En la cabecera también pueden incluirse códigos escritos en diferentes lenguajes interpretados, (JavaScript, PHP, ASP,…), contenidos dentro del elemento <script>. Con estos códigos se consigue implementar el acceso y recuperación de contenidos almacenados en una base de datos o simplemente aportar dinamismo al documento.

Con la etiqueta <meta> se permite introducir información para la que no se definió ningún elemento del lenguaje (información no visible desde la página web). La información almacenada en este elemento tiene gran importancia porque permite transmitir datos etiquetados semánticamente a una aplicación que posteriormente procese el documento. Un ejemplo de uso de este elemento es el siguiente:

<meta name= "keywords" content="Python, Django, framework, open-source" />
<meta name= "description" content="Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design." />

De este modo, el programador pasa una metainformación al navegador con ayuda de los atributos “name” y “content” de este elemento <meta>. Esta característica podría ser muy importante en tareas de acceso a la información web. Por Ejemplo, las metaetiquetas “keywords” y “description” están totalmente dedicadas a indicar a los robots de los buscadores como han de indexar la página web, por lo que deben contener elementos claves sobre el tema que se aborda en la página web.

Lamentablemente, diversos estudios [(Pierre, J., 2001) (Riboni, D., 2002)] han mostrado que este tipo de elementos se encuentran en menos de un 30% de las páginas analizadas.

Cuerpo (<body>)

El cuerpo de un documento HTML está formado por elementos relativos a la estructura y a cómo debe visualizarse la información contenida en el documento HTML. Dentro de esta etiqueta se incluye el texto que se desea hacer visible en la página web.

Dentro del <body> pueden utilizarse diferentes encabezados (<h1> . . . <h6>) que permiten realizar una ordenación jerárquica de los apartados en los que se quiera estructurar un documento.

En general, el vocabulario HTML tiene dos tipos de estilos: físicos y lógicos. Los estilos físicos son aquellos que siempre implican un mismo efecto tipográfico, mientras que los lógicos marcan un texto que por sus características debe tener un modo de mostrarse propio. Por ejemplo, son estilos lógicos: el elemento <address>, que codifica direcciones de correo electrónico o direcciones personales; o <blockquote>, que permite marcar citas textuales, mostrando el

texto resaltado y separándolo del texto que lo circunda. El elemento <dfn> especifica una definición y con se indica que el autor quiere destacar el contenido de ese elemento con énfasis. Con el elemento <code> se puede introducir como texto un fragmento de código fuente sin que llegue a ser interpretado por el navegador. Con <kbd> se pueden marcar textos tecleados por el usuario. Con <strike> se presenta un texto tachado, mientras que con la etiqueta se resalta el contenido. Con <var> se especifica una tipografía diferente para marcar que se trata de una variable, en el caso de que en el contenido del documento se quiera mostrar un código fuente.

Como ejemplo de estilos físicos se pueden destacar: el elemento <b>, que destaca una porción de texto en negrita; <i>, que hace lo propio, pero en cursiva; <sub> y <sup>, que permiten formatear un texto como subíndice o superíndice; los elementos <big> y <small>, que se emplean si se quiere mostrar una porción de texto en mayor o menor tamaño; o el <tt>, que muestra su contenido a modo de máquina de escribir.

Uno de los aspectos primordiales de este lenguaje es el formateo de la propia fuente. En la práctica, resulta muy común presentar texto resaltado en negrita, itálica, con otros efectos tipográficos. Se puede obtener un mismo resultado empleando estilos físicos y lógicos.

El lenguaje HTML es interpretado por los navegadores según su criterio, por lo que una misma página web puede ser mostrada de distinto modo según el navegador. Mientras que <b> significa simplemente negrita y todos los navegadores la interpretarán como negrita, es una etiqueta que indica que su contenido debe resaltarse y cada navegador será responsable de hacerlo como estime oportuno.

En la práctica, muestra el texto en negrita, pero podría ser que un navegador decidiese resaltarlo con negrita, subrayado y en color rojo. En el caso de querer aplicar un estilo de fuente itálica también existirían dos posibilidades:

<i>, que sería interpretado como itálica; y , que se interpretaría como el estilo lógico de enfatizar, aunque igualmente se suela mostrar como un texto en itálica.

A pesar de alcanzarse valiosos resultados con el uso de los estilos físicos y lógicos, la tendencia actual es utilizar las llamadas Hojas de Estilo en Cascada o CSS (del inglés: Cascading Style Sheets), las cuales permiten dar estilo al documento HTML, separando el contenido de la presentación. CSS permite a los desarrolladores web controlar el estilo y el formato de múltiples páginas web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en el CSS afectará a todas las páginas vinculadas a ese CSS, de igual forma a una misma página se le podrán aplicar diferentes estilos. No obstante, independientemente del estilo que se le aplique a una página, van a seguir existiendo ciertas etiquetas que forman parte del contenido HTML que no serán variada. Por otro lado la complejidad de encontrar qué estilos se les aplicó a cada elemento de la página resulta un proceso extremadamente costoso, además que muchos de las arañas empleadas en el proceso de recuperación de la información no obtienen los ficheros CSS asociados a las páginas. Por lo que sólo se analizarán los estilos logrados desde el propio lenguaje HTML.

Un aspecto importante es que a partir de esta información de carácter tipográfico se puede extrapolar información relativa a la intención del autor en el momento de crear el documento, intuyendo qué partes quiso destacar frente a otras o con qué elementos del discurso quiso llamar la atención del lector.

1.2.1 Comunicación por medio de páginas HTML.

La comunicación por medio de páginas web se puede considerar como un *proceso informativo documental* (Fresno, V., 2006). Un emisor codifica un mensaje en lenguaje HTML y lo transmite por un medio hacia un receptor que lo deberá decodificar. Este proceso es un proceso activo. Por un lado, el emisor utiliza las características del lenguaje para hacer llegar su mensaje al receptor, el cual

deberá interpretarlo con ayuda del conocimiento que tenga del propio lenguaje, experiencia personal y el entorno cultural en el que se ha desarrollado.

Los autores de un documento incluyen señales en el texto que marcan o acentúan las ideas importantes (Cerezo, A., 1994). Tamaños de los tipos de letra, uso de itálicas, subrayados. Orden de las palabras; las ideas más importantes suelen estar al comienzo de la frase, párrafo o texto. Los títulos de la obra, del capítulo o del apartado ayudan a resumir el contenido del texto o ponen de manifiesto la intención del autor.

De este modo, el efecto del texto sobre el lector dependerá enormemente del modo en el que se le presente la información. Esta es una de las ideas fundamentales sobre las que se apoya la presente investigación.

Desde el punto de vista de la lectura, una de las primeras consideraciones realizadas en el desarrollo de páginas web es que, aún siendo igualmente texto, su contenido puede ser distinto al que se encuentra en un texto impreso. Pero no sólo porque el hipertexto permita una lectura no lineal y un desplazamiento entre contenidos en diferentes páginas, sino por el hecho de que las personas se comportan de un modo diferente ante una pantalla que frente a una página de papel.

En un estudio realizado en 1997 por Jacob Nielsen se descubrió que la lectura de textos en pantallas de ordenador es diferente que si se lee en un texto en papel. Sólo un 16% de los usuarios de prueba leyó las páginas web mostradas de modo secuencial, frente a un 79% que, al leer un documento HTML, realizaron su lectura saltando entre los temas más importantes, fijando su atención en diferentes partes de la páginas, y no palabra por palabra como ocurre en los textos impresos (Nielsen, J., 1997).

Cuando se ojea el contenido de una página web y se salta de una parte a otra en busca de información relevante, uno de los procesos que se ponen de manifiesto más activamente es la atención, ya que el autor quiere transmitir una información y

el usuario debe buscar aquellas partes del contenido donde crea que pueda encontrar la información que precisa, sin necesidad de realizar una lectura lineal completa.

A partir de numerosos estudios realizados sobre la fase de captación de la información desde finales del siglo XIX, se puede concluir que las dimensiones físicas de los objetos que mejor captan y mantienen nuestra atención son:

*El tamaño***.** Normalmente los objetos de mayor tamaño llaman más la atención. En concreto, al doblar el tamaño aumenta el valor de la atención en un 42-60 %.

La posición. La parte superior atrae más; la mitad izquierda más que la mitad derecha. Por tanto, la mitad superior izquierda de nuestro campo visual es la que capta antes nuestra atención. Esto concuerda con los estudios generados y descritos en (Web Style Guide.com, 2nd Edition)

El color. Los estímulos en color suelen llamar más la atención del sujeto que los que poseen tonos en blanco y negro.

El movimiento. Los estímulos en movimiento captan antes y mejor la atención que los estímulos inmóviles.

1.2.2 Modelo de Objetos de Documento (DOM)

De poco serviría contar con información relativa a la intención del autor desde el código de la propia la página, si no fuera posible poder acceder a ella de un modo sencillo. El Modelo de Objetos de Documento o DOM, es la interfaz que permite acceder y manipular, los contenidos de una página web (o documento). Proporciona una representación estructurada, orientada a objetos, de los elementos individuales y el contenido de una página, con métodos para recuperar y fijar las propiedades de los objetos. Además, proporciona métodos para agregar y eliminar dichos objetos, permitiendo crear contenido dinámico (Hall, M., 2008).

El DOM está definido y administrado por *World Wide Web Consortium* (W3C), por lo que los distintos navegadores simplemente aplican las especificaciones del W3C, para dar soporte al DOM en sus aplicaciones.

A lo largo de la historia de los navegadores, se han ido aplicando en mayor o menor manera las características del DOM. A medida que se sucedían versiones, los navegadores también iban dando mayor soporte a las especificaciones del DOM.

El Árbol del Documento

Cuando un navegador carga una página, crea una representación jerárquica de los contenidos que representa, aproximadamente, su estructura HTML. Esto desemboca en una organización parecida a un árbol de nodos (Figura 2), en cada nodo se almacenan diferentes tipos de objetos que van desde etiquetas HTML, atributos, contenidos, etc. Cada objeto tendrá sus propios métodos y propiedades, por lo que cada tipo de objeto implementa la interfaz *Nodo*.

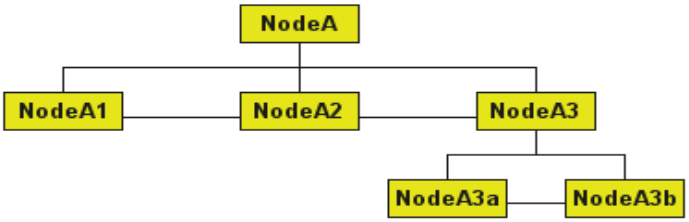

Figura 2: Árbol de Nodos

A continuación se muestran algunas relaciones a partir del ejemplo anterior:

- NodeA.firstChild -> NodeA1
- NodeA.lastChild -> NodeA3
- NodeA.childNodes.length -> 3
- NodeA.childNodes[0] -> NodeA1
- NodeA.childNodes[1] ->NodeA2

- NodeA1.parentNode -> NodeA
- NodeA1.nextSibling -> NodeA2
- NodeA3.prevSibling -> NodeA2
- NodeA3.nextSibling -> null
- NodeA.lastChild.firstChild -> NodeA3a
- NodeA3b.parentNode.parentNode -> NodeA

La interfaz Nodo también proporciona métodos para añadir, actualizar y eliminar nodos dinámicamente, tales como:

- insertBefore()
- replaceChild()
- removeChild()

Recorriendo el árbol del Documento

El árbol del documento refleja la estructura del código de una página. Cada etiqueta está representada por un nodo elemento.

El objeto documento tiene sólo un elemento hijo, dado por **document.documentElement**. Para páginas web, éste representa la etiqueta exterior HTML, siendo ésta el elemento raíz del árbol del documento y tiene como hijos los elementos HEAD y BODY que tendrán a su vez otros elementos hijos.

Con en el empleo de los métodos de la interfaz *Nodo*, se puede recorrer el árbol del documento para acceder a los nodos individuales contenidos en dicho árbol permitiendo un rápido acceso a la información de marcado y logrando que cualquier exploración del código HTML se pueda realizar de forma sencilla.

1.2.3 Representación automática de Documentos HTML

Representar de un modo adecuado un documento resulta una tarea fundamental y debería ser, por tanto, la primera acción a realizar. La representación debería ser

fiel, en primer lugar, al contenido del documento, incluyendo la información necesaria para poder extraer el conocimiento útil que se espera obtener y, a la vez, debería ser adecuada a las especificaciones de los algoritmos que se empleen a continuación.

Gran parte de los modelos de representación de documentos coinciden en el uso de la palabra como elemento fundamental en la representación de la información textual. De este modo, una representación, en última instancia, debería ser un conjunto de rasgos que, de una manera u otra, representen el contenido del documento. Este conjunto de rasgos formaría el vocabulario V , conjunto de objetos X dentro del modelo de representación, y tendrá asociado un valor de relevancia r_{ij} para cada rasgo t_i dentro del contenido de un documento d_j .

Cuando se trata de representar un documento HTML, el problema puede enfocarse desde diversos ángulos (Fresno, V., 2006), dependiendo de los elementos que se quieran considerar. Así, una página web puede verse, fundamentalmente, como la suma de:

Texto enriquecido: es decir, combinación entre el contenido de texto de una página y una información específica, en forma de anotaciones y capaz de aportar información tipográfica sobre cómo debe mostrarse el contenido.

Metainformación: metaetiquetas en las que se puede incluir información relativa a la propia página web, como puede ser: autores, palabras claves (keywords) que describan el contenido y ayuden en el proceso de exploración automática.

Estructura de hiperenlaces o hyperlinks: lo más característico de un documento web. Los hiperenlaces de una página web son referencias a documentos, o determinadas partes de documentos, en una relación 1 a 1 unidireccional.

En función de cualquiera de estos elementos, un modelo de representación de documentos deberá definir el espacio matemático de representación de la página web.

A la hora de representar un documento HTML, en primer lugar, podrá considerarse como un texto. Aunque se considera que la información de marcado es de sumo significado y debe tenerse en cuenta.

1.2.4 Modelos vectoriales

Los modelos de representación vectoriales (Salton, G. y Lesk, 1965), son un tipo dentro del conjunto de técnicas de representación de documentos que han sido muy empleadas en sistemas de IR[22] (Recuperación de información), TC[23] (Categorización Textual) y DC[24] (Agrupamiento de Documentos) en los últimos años. Las representaciones vectoriales resultan muy sencillas y descansan sobre la premisa de que el significado de un documento puede derivarse del conjunto de rasgos presentes en el mismo. Representan modelos formales y pueden considerarse "basados en rasgos", estos rasgos serán, de un modo u otro, los vectores generadores de un espacio vectorial. Los documentos se modelan como conjuntos de rasgos que pueden ser individualmente tratados y pesados. De este modo, en tareas de TC y DC, los documentos pasan a ser representados como vectores dentro de un espacio euclídeo, de forma que midiendo la distancia entre dos vectores se trata de estimar su similitud como indicador de cercanía semántica.

En la mayoría de los casos, estos modelos no tratan de reducir las dimensiones del espacio, colapsándolas en un subconjunto más reducido, y consideran cada rasgo como un objeto independiente. A pesar de esto, no son simples ficheros que guardan información de relación entre rasgo y documentos, sino que representan modelos más flexibles, al permitir realizar el pesado de cada rasgo individualmente (por medio de funciones de relevancia), de forma que éste pueda considerarse más o menos importante dentro de un documento o de la colección.

---

[22] Localización, dentro de una colección de documentos, de un subconjunto relevante para una consulta formulada por un usuario.

[23] Asignación de un documento a una categoría previamente conocida.

[24] Agrupación de documentos con características similares.

1.2.5 Funciones de ponderación o de relevancia.

En la literatura pueden encontrarse multitud de funciones de ponderación empleadas para calcular la importancia, o relevancia, de un rasgo en el contenido de un texto. Estas funciones pueden emplear parámetros diferentes según los casos; desde la frecuencia de aparición de un rasgo en el documento o en la colección, hasta probabilidades condicionadas de un rasgo a una clase en problemas de TC.

Las funciones de ponderación se basan fundamentalmente en un "conteo" de frecuencias, ya sea dentro del documento a representar, o en el conjunto de documentos de la colección. En primer lugar, pueden distinguirse funciones de carácter "local" y "global". `

Se consideran funciones de ponderación **"local"** aquellas que toman únicamente información del propio documento para obtener una representación, sin necesidad de ninguna información externa siendo estás las más usada el algoritmos de carácter no supervisado y se consideran **"global"** a aquella que toma información de la colección o sea externa del propio documento.

Del total de funciones que pueden encontrarse en la literatura, algunas de las más populares son:

Funciones locales:

Función binaria (Binary, Bin) (Salton, G. y McGill, M. , 1987). El método de ponderación más sencillo, considera únicamente la presencia o ausencia de un rasgo en un documento para calcular su relevancia dentro del mismo. La función de relevancia es un valor {0,1} y se puede expresar como:

$$F = \begin{cases} 1 & si\ el\ término \in \text{documento} \\ 0 & e.o.c \end{cases}$$

Frecuencia de aparición o TF (Term Frequency) (Salton, G. y McGill, M., 1987). Cada término tiene una importancia proporcional a la cantidad de veces que

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

Funciones globales:

Frecuencia del Término X Frecuencia Inversa del Documento (TF-IDF) (Salton, G. y McGill, M., 1987)*:* Mientras el factor TF tiene que ver con la frecuencia de un término en un documento, el IDF (Inverse Document Frequency) tiene que ver con la frecuencia de un término en la colección de documentos. Así, la importancia de un término es inversamente proporcional al número de documentos que lo contiene:

$$w(t,d) = TF(t,d) * IDF(t)$$
$$IDF(t) = \log(N/df(t))$$

Donde *N* es el número de documentos de la colección ζ y df(t) es el número de documentos que contienen a t. Es decir, mientras menos documentos contengan al término t mayor es su IDF(t). Por el contrario, si todos los documentos de la colección contienen al término t entonces IDF(t) es cero. El factor TF(t,d) contribuye a mejorar la relevancia y el factor IDF(t) contribuye a mejorar la precisión, pues representa la especificidad del término, distinguiendo los documentos en los que éste aparece de aquellos en los que no aparece. El IDF(t)

es útil como indicador de la bondad del término t como discriminador de documentos.

Si se tiene en cuenta el tamaño actual y crecimiento de Internet, el costo de introducir información de contexto, frente a considerar únicamente información contenida en la página web, puede resultar muy elevado.

En el caso de páginas web, esta dependencia externa implicaría considerar el total de los documentos contenidos en la web o, al menos, un subconjunto suficientemente significativo del ámbito donde se aplica la Generación de Resúmenes.

De este modo, cualquier función de ponderación local sería completamente independiente del tamaño actual y futuro de la web. Además, podría aplicarse en sistemas sin necesidad de contar con enormes medios de almacenamiento ni procesamiento, ni tampoco requerirá una exploración intensiva de colecciones de documentos correlacionados.

A continuación se presenta la función de ponderación local propuesta por Víctor Fresno en el marco de su tesis Doctoral (Fresno, V., 2006). Se ha hecho una distinción de esta función de ponderación local del resto, con toda intención, pues ha sido la única función de ponderación encontrada en la literatura que hace uso de determinadas características del lenguaje HTML.

ACC[25]*:* Emplea para determinar la relevancia de un término dentro del contenido de un de una página web una función lineal que combina una serie de criterios heurísticos extraídos del proceso de escritura y lectura de páginas web.

ACC se expresa como sigue:

$$\mathrm{ACC}\ (t_i, \mathrm{d}_j) = \mathrm{C}_1 f_{\mathrm{criterio1}}(t_i, \mathrm{d}_j) + \ldots + \mathrm{C}_n f_{\mathrm{criterio\ n}}(t_i, \mathrm{d}_j)$$

[25] Analytical Combination of Criteria

Donde, $ACC\ (t_i, d_j)$ representa la relevancia del termino t_i en el contenido del documento d_j. De este modo, $C_k\ f_{criterio\ k}(t_i, d_j)$ representa el aporte del criterio k-ésimo a la relevancia final del término t_i en el documento *dj*. El coeficiente C_k indica el valor de importancia que se le dio al criterio n.

Los criterio heurísticos propuestos por Fresno fueron: *frecuencia, título, posición y enfatizados* y los fundamentó como sigue:

Frecuencia: La frecuencia con la que aparece un término en un documento debe ser un factor determinante a la hora de establecer su relevancia. Pues el autor de la página podía ayudar a orientar al lector por medio de la repetición de términos significativos (Fresno, V., 2006).

Título: Los términos que se encuentran situados en el título de un documento entre las etiqueta <TITLE> <TITLE> deberían considerarse con una relevancia elevada dentro de la página web, ya que cabe esperar que resuman el contenido del documento. Un título informativo y concreto ayuda al lector a orientarse. Sin embargo, el hecho de que el contenido de este elemento no sea visible en el cuerpo del documento HTML hace que este elemento no se encuentre en la mayor parte de las páginas web. (Fresno, V., 2006).

Posición: Se asume que las páginas web tienen un carácter más bien expositivo y se plantea además que los textos con este carácter están estructurados en Introducción, Desarrollo y Conclusiones, considerando más representativa una oración que se encuentre en la primera y la última parte del texto, frente a otra que aparezca en la parte central del mismo. Una persona se puede orientar durante la lectura de un texto si encuentra una introducción que especifique el tema que se desarrolla en el texto que está leyendo. Por tanto, la posición de un rasgo dentro de un documento puede resultar muy útil para encontrar su relevancia dentro del documento (Fresno, V., 2006).

Enfatizado: El lenguaje HTML tiene etiquetas cuya función es la de destacar determinadas partes de un texto frente a otras (..., <u>...</u>, ..., ..., <h1> - <h6> o ...). El texto marcado con estas etiquetas llama la atención del usuario y, en muchos casos, basta con tomar estos fragmentos enfatizados para crearse una idea sobre el contenido de un documento. No se consideran los colores de las fuentes como elementos de enfatizado, ya que lo que llama la atención de un usuario es el contraste más que el color. Como una página web puede tener una imagen de fondo, un cambio de color en la fuente puede ser simplemente para establecer un contraste alto entre el fondo y el texto.

A partir de estos criterios, ACC queda definida como sigue:

$$ACC_{0,3/0,15/0,25/0,3}(t_i, d_j) = 0,3f_{frec}(t_i, d_j) + 0,15f_{tit}(t_i, d_j) + 0,25f_{enf}(t_i, d_j) + 0,3f_{pos}(t_i, d_j)$$

Donde, $f_{frec}(t_i, d_j)$ representa la frecuencia del término i en el contenido del documento j multiplicado por un coeficiente que determina la importancia del criterio frecuencia(0,3), $f_{tit}(t_i, d_j)$ representa la frecuencia del término i en el título del documento j por un coeficiente de importancia (0,15), $f_{enf}(t_i, d_j)$ representa la frecuencia del término i en el conjunto de palabras enfatizadas del documento j multiplicado por un coeficiente que determina la importancia del criterio frecuencia (0,25) y por último $f_{pos}(t_i, d_j)$ representa la frecuencia del término i en el conjunto de palabras en posiciones importantes del documento j multiplicado por un coeficiente que determina la importancia del criterio frecuencia (0,3).
De aquí que los criterios de mayor importancia según Fresno, son los relacionados con el título y con la posición.

Se concuerda plenamente con Fresno en el empleo de una función de combinación lineal de los criterios para la ponderación de los términos dentro de un documento HTML, pues posiblemente, ningún criterio heurístico sea lo

suficientemente bueno por sí sólo y según las situaciones, unos funcionen mejor que otros.

En cuanto a los criterios heurísticos seleccionados por Fresno, se considera que la selección de estos debe estar en dependencia de la tarea en la que se quieran emplear, para el caso de la GAR se puede pensar en otros criterios heurísticos que aporten elementos más significativos a la hora de generar el resumen de una página web.

1.2.6 Selección del vocabulario

En esta sección, se introducen algunos aspectos relacionados con la selección de rasgos como elementos de transformación de una información, que inicialmente es de carácter cualitativo y que debe ser transformada a un conjunto de objetos X dentro de un espacio medible $< X, B >$, de carácter cuantitativo.

Como selección de vocabulario pueden entenderse aquellas fases que transforman un texto en el conjunto de rasgos que lo podrán representar; las más comunes cuando la representación se basa en las palabras individuales son las siguientes:

Análisis léxico: Fase donde se analiza un texto para distinguir las cadenas que formarán parte de la representación. Para encontrar tales cadenas se tienen en cuenta, por ejemplo: delimitadores como espacios en blanco, signos de puntuación, guiones y otros signos especiales.

Lematización: El proceso de lematización es aquel donde a cada forma flexiva se le asigna su lema. La obtención de raíces o lemas es una técnica que utilizan los sistemas de RI para aumentar su efectividad y reducir el tamaño de los archivos de indexación. Este proceso consigue obtener un único término a partir de palabras con el mismo significado pero que difieren esencialmente en su morfología (Frakes, 1992; Krovetz, 1993), por ejemplo, se puede considerar la obtención del término niño a partir de niños y niñita. En el caso de los verbos, por ejemplo, se obtiene el infinitivo amar a partir de amo y amará. Obteniendo como

resultado una misma forma canónica para las diferentes variantes morfológicas de un término, que no tiene porque ser necesariamente su raíz lingüística.

Este proceso comprende la eliminación de los plurales, de ciertos prefijos y sufijos, de las conjugaciones verbales y su reducción al infinitivo, etc. Ejemplos de herramientas multilingües que realizan la lematización son FreeLing[26] o TreeTagger[27] . Un inconveniente de la realización de la lematización es que estos algoritmos necesitan del conocimiento previo del idioma en el que está escrito el documento.

Eliminación de StopWords: En este proceso se eliminan aquellas palabras que se emplean para articular el discurso, tales como artículos, preposiciones, conjunciones, etc., pero que no tienen por sí solas una semántica relevante en el contenido de un texto. Se considera que este tipo de palabras no tienen capacidad discriminante. La comunidad científica dispone de listas de "stopwords" para numerosos idiomas, entre las que se incluyen también algunos verbos, adverbios o adjetivos de uso frecuente. La idea de eliminar estas palabras surgió de un trabajo de Salton, (Salton, G. y McGill, M. , 1987), en el que se constató que se obtenía mejores resultados en tareas de IR, cuando los documentos eran menos similares entre sí, es decir, si se les quitaban muchas de las palabras que compartían, logrando reducir así la densidad del espacio de los documentos.

1.3 Principales aproximaciones en Generación Automática de Extractos

Los orígenes en el campo de estudio de los resúmenes automáticos se remontan a los trabajos de Luhn 1958 y Edmundson 1969 ambos considerados actualmente los pioneros en esta área debido a que fueron ellos los primeros en proponer técnicas y desarrollar sistemas para la obtención automática de resúmenes con un enfoque puramente extractivo.

[26] http://garraf.epsevg.upc.es/freeling/
[27] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html

Hans Peter Luhn (Luhn H., 1958), fue el primero en proponer un método estadístico para extraer las sentencias más significativas de un texto y construir un resumen del mismo. Para ello proponía determinar en primer lugar la significatividad de las distintas palabras, suponiendo que las más frecuentes (a excepción de las palabras vacías) serían las más importantes. Posteriormente se asignaría a cada sentencia un peso en función del número de palabras importantes que incluyese. Una vez obtenida la puntuación de todas las sentencias del documento sería posible ordenarlas de mayor a menor importancia y seleccionar un subconjunto de las más significativas como resumen del texto original.

Es importante destacar que Luhn también vio la necesidad de adaptar los resúmenes automáticos a los distintos intereses de los usuarios. Para lograr esto proponía asignar una "prima" a las palabras utilizadas por el usuario para describir su necesidad de información de tal modo que las sentencias que contuviesen dichas palabras obtuviesen mayores puntuaciones y pasasen al resumen con mayor facilidad.

Las investigaciones realizadas por Luhn fueron enriquecidas unos años más tarde con los aportes de Harold P. Edmundson (Edmundson, H. 1969), quien emplea cuatro rasgos distintos para asignar pesos a las sentencias del documento. Utiliza una lista de palabras que proporcionan pistas sobre la relevancia o irrelevancia de las sentencias, la utilización de palabras frecuentes (y no vacías) como indicadores de relevancia (semejante al método de Luhn), el uso de palabras del título del documento como indicadores positivos y heurísticas basadas en la posición de las sentencias en el texto. Cada uno de estos métodos contribuía al peso final de las sentencias de manera independiente y configurable. Un aspecto significativo en el trabajo desarrollado por Edmundson, lo constituye el hecho de ser uno de los primeros en señalar la necesidad de evaluar los sistemas de extracción de resúmenes. Para ello comparaba sus resultados con resúmenes producidos por evaluadores humanos.

Durante las dos décadas siguientes el interés por este tema no fue muy importante. Sin embargo, a partir de los años 90 y, muy especialmente en los últimos años, la investigación en el área creció de una forma significativa.

En el año 1994 Gerard Salton y Amit Singhal (Salton, G. y Singhal, 1994), parten de identificar en primer lugar los distintos temas tratados en un documento así como los párrafos del texto que se refieren al mismo asunto para, posteriormente, emplear una selección de párrafos como resumen del documento. Se trata de realizar una clasificación automática de párrafos controlada mediante un umbral de similitud que será inversamente proporcional al número de temas que se deseen "descubrir" en el documento.

Un año más tarde, en 1995, Julian Kupiec, Jan Pedersen y Francine Chen (Kupiec, J. *et al.,* 1995), desarrollan una nueva técnica en la que emplean colección de entrenamiento (documentos, resumen creado manualmente) para un clasificador bayesiano que debía determinar qué sentencias de un documento deberían formar parte de un resumen y cuáles no. Proponen un sistema que determinaba para cada sentencia la probabilidad de pertenencia al resumen final y extraía las más probables. Al reducir los documentos a un 25% del tamaño original seleccionaba un 84% de las sentencias elegidas por los expertos humanos y para resúmenes más cortos resultaba sustancialmente superior al estudio anterior, que representaba solamente el inicio del documento.

Siguiendo esta misma línea se destacan las investigaciones nuevamente de Salton y Singhal, pero esta vez de conjunto con Chris Buckley y Mandar Mitra, en esta ocasión tratan de avanzar desde la identificación y clasificación de párrafos hacia la identificación de pasajes, o sea, "fragmentos de texto que exhiben consistencia interna y que pueden distinguirse del resto del texto circundante" (Salton, G. *et al*., 1996).

Este mismo equipo en el año 1997 (Salton, G. *et al*., 1997), desarrollan aún más algunas de las ideas expuestas en años anteriores sobre la utilización del grafo de relaciones entre pasajes para la extracción de aquellos más significativos y la construcción de un resumen automático.

Ese mismo año un equipo conformado por los japoneses Fumiyo Fukumoto, Yoshimi Suzuki y Jun-ichi Fukumoto desarrolla una nueva técnica en la que asignan a cada palabra del texto un peso que dependerá de su distribución en el propio documento y en un contexto más amplio. Plantean que una palabra será palabra clave si su dispersión a nivel de párrafo es menor que a nivel de documento y ésta a su vez es menor que la del término en el dominio. Para cada palabra no vacía se determina su peso dentro del párrafo, el documento y el dominio y se seleccionan aquellas que verifican los dos criterios anteriormente expuestos. Luego cada párrafo del documento se representa mediante un vector que sólo incluirá las correspondientes palabras clave y se realiza una clasificación automática de manera análoga a la de Salton. El resumen se construirá seleccionando en primer lugar aquellos párrafos que estén incluidos en un mayor número de los grupos resultantes del proceso de clasificación. Es importante destacar que la principal ventaja de este método radica en la posibilidad de ajustar los resúmenes a distintos contextos aunque a la vez, es su principal inconveniente al requerir un corpus para extraer resúmenes (Fukumoto F., et al. ,1997).

En el propio año 1997 Regina Barzilay y Michael Elhadad plantean la utilidad de las cadenas léxicas como elemento facilitador en la extracción de resúmenes. Una cadena léxica es una secuencia de palabras semánticamente relacionadas que aparecen en un texto y que pueden ser adyacentes o encontrarse dispersas a lo largo del documento. Para encontrar dichas cadenas léxicas en un texto genérico es necesario utilizar recursos como WordNet (Miller, G., 1990) que proporcionan la información necesaria sobre las posibles relaciones entre distintas palabras. Así pues, Barzilay y Elhadad encuentran en primer lugar cadenas léxicas en el texto, seguidamente asignan a cada cadena léxica una puntuación y, por último, seleccionan aquellas sentencias que mejor satisfacen a las cadenas léxicas de mayor puntuación (Barzilay, R. y Elhadad, M., 1999).

En el año 2001 Meru Brunn, Yllias Chali y Christopher Pinchak desarrollaron un trabajo muy similar al de Barzilay y Elhadad, concluyendo además que las cadenas léxicas pueden resultar muy interesantes para introducir conocimiento lingüístico en los métodos extractivos (Brunn, M. *et al*., 2001).

Ese mismo año Hilda Hardy, Nobuyuki Shimizu, Tomek Strzalkowski, Ting Liu, Xinyang Zhang y G. Bowden Wise (Hardy, H. *et al*., 2001), describen un sistema para construir resúmenes a partir de varios documentos (decenas o cientos). Para ello, dividen cada documento en párrafos que son clasificados automáticamente empleando una medida de similitud basada en n-gramas de palabras. Una vez descubiertos los distintos grupos se selecciona un párrafo de cada uno para construir el documento final (McKeown, K. et al., 2001), (Fuentes, M. et al., 2003) y (Doran, W. et al., 2004) también han utilizado cadenas léxicas como método de puntuación.

Jing y McKeown (2001) de la Universidad de Columbia estudiaron diversas tareas de post-procesamiento de los resúmenes extractivos para mejorar su calidad (Jing, H. y McKeown, K., 2001). Aun cuando otros autores (Mani, I. *et al*., 1999) ya trataron dicho problema el interés de este trabajo radica en la forma en que se aborda: Jing y McKeown desarrollaron una técnica que permite en primer lugar analizar la relación entre un resumen manual (creado por un humano) y el documento original a fin de determinar, por un lado, las sentencias "extraídas" y, por otro, las fases de reducción, combinación y reordenamiento a que fueron sometidas.

En (Nomoto, T. y Matsumoto, Y., 2001) se utilizan técnicas de agrupamiento para identificar los diferentes temas que aborda un documento. El resumen se confecciona seleccionando la frase más importante de cada uno de los temas. El algoritmo de agrupamiento seleccionado, se caracteriza por estimar matemáticamente el número de grupos a considerar (de otro modo, este número debe ser proporcionado por el usuario). La relevancia de una frase se estima sumando los pesos tf·idf de las palabras de la frase que forman parte del índice. En una comparación con resúmenes construidos teniendo en cuenta sólo el criterio de relevancia de las frases, el enfoque propuesto resulta más efectivo para distintas tasas de compresión.

Erkan y Radev (Erkan, G. y Radev, D., 2004a) han desarrollaron una nueva medida de "centralidad" para las sentencias, denominada LexPageRank, basada en la idea de "prestigio" de las redes sociales y análogas al PageRank de Google.

El valor de LexPageRank para una sentencia S se define como la suma de los valores LexPageRank de aquellas sentencias similares a S, donde la similitud se determina mediante la función del coseno. Esta última versión de MEAD resultó uno de los mejores participantes en cuatro de las cinco tareas de DUC 2004 (Erkan, G. y Radev, D., 2004b).

Por su parte, Vanderwende, Banko y Menezes (2006) utilizan PageRank para determinar qué elementos de un documento son los más relevantes aunque sus resúmenes son generados y no construidos a partir de sentencias extraídas literalmente de los documentos (Vanderwende, L. et al., 2006). Guardando cierta relación con los desarrollados por Salton et al. (1996) que también emplearon grafos para analizar los contenidos de un texto.

Daniel Gallo, (2006) propone una técnica llama *blindLight*, basada en el uso de vectores de n-gramas de longitud variable, Cada n-grama tiene asociado un peso que indica cuan significativa resulta su aparición en el documento. Quedándose finalmente con aquellos n-gramas que resultaron ser los más relevantes (Gallo, 2006).

Esau Villatoro Tello, (2007) propone un método para la generación de resúmenes de varios documentos, aplicando algoritmos de agrupamiento, con el objetivo de organizar la información por sub-temas de eliminar redundancias y controlar niveles de compresión (Esau-Villatoro, T., 2007).

Iria da Cunha Fanego, (2008), propone un método para la generación automática de resúmenes de documentos médico en español, basado en análisis de un corpus (Cunha, I., 2008).

Concluido el análisis, se pudo apreciar que los algoritmos encontrados trabajan sobre el texto plano, o sea, el texto sin ningún tipo de marca proveniente del formato, todos utilizan métodos estadísticos y/o lingüísticos para la selección de las frases significativas del texto. En muchos casos hacen uso de atributos que son dependientes del dominio temático e incluso del idioma de los documentos. Convirtiendo al texto en una entidad matemática.

En caso de los documentos HTML, donde está enfocado nuestro campo de acción, la información de marcado empleada por los autores de las páginas web para resaltar frases consideradas por ellos más relevantes que otras, podrían ser de suma importancia si de selección de frases importantes en documentos HTML se trata, esta información accesible desde el propio código de la página podría emplearse como heurística para determinar la relevancia de una frase en un documento. Lográndose independencia de dominio temático e idioma.

1.4 Conclusiones

A partir de las ideas expuestas en las secciones precedentes se decidió desarrollar un algoritmo no supervisado para la generación automática de extractos de páginas web, empleando para determinar la relevancia de los términos, una función de ponderación local similar a la presentada por Fresno(Fresno, V., 2006) que combine criterio heurísticos acordes a la Generación Automática de Extractos, logrando de esta forma explotar toda la información relativa a la intención del autor en el momento de crear el documento. Enajenándose de toda dependencia de dominio temático.

Capítulo 2. Algoritmo para la Generación Automática del Extracto de un Documento HTML

En el presente capítulo se realiza la propuesta de un algoritmo no supervisado para la generación automática de extracto de un documento HTML, la cual se desglosa en las mismas fases de un algoritmo de generación de extractos.

El algoritmo se sustenta en una serie de criterios heurísticos usados para la extracción del contenido relevante. Se propone además un método para la detección del idioma de los documentos. Se definen las funciones de captura de cada uno de los criterios heurísticos considerados y se establecen los coeficientes que fijan la importancia que se le quiera dar a cada criterio.

2.1 Criterios Heurísticos

El adjetivo "heurístico" significa "medio para descubrir" y está relacionado con el término griego heuriskein que significa hallar, inventar. Por heurístico se entiende un criterio, estrategia o método empleado para simplificar la resolución de problemas complejos. Las heurísticas suponen, por tanto, un conocimiento que se obtiene a través de la experiencia.

Primeramente se analizarán los criterio heurísticos *frecuencia, título, posición y enfatizado,* combinados en la función ACC (Fresno, V., 2006).

El criterio heurístico relacionado con la ***frecuencia*** de aparición de un término en el contenido de un documento, ha resultado el parámetro más utilizado por la mayoría de las funciones de ponderación encontradas, y es un elemento muy tenido en cuenta por la mayoría de los algoritmos para la GAR presentes en la literatura. Se concuerda con Fresno, cuando plantea que este criterio no debe considerarse aisladamente, ya que esto podría potenciar palabras de uso común,

palabras muy utilizadas en el discurso pero que no permiten distinguir claramente contenidos de documentos con temáticas diferentes.

El hecho de que el contenido ***título*** <title></title> no sea visible en el cuerpo del documento HTML sino usualmente en el título de la ventana del navegador, hace que no forme parte del contenido que más llama la atención del usuario. Es común que no se encuentre en la mayor parte de las páginas web o sea generado de forma automática, no estando relacionado con el contenido del documento específico, sino del sitio del cual forma parte. En el marco de la presente investigación se realizó un análisis del título que presentaban cerca de 2000 páginas web y de ellas sólo en 5% presentaban un título acorde con el contenido de la página. Siendo esta la razón por lo que este criterio heurístico no se tendrá en cuenta.

En cuanto al criterio ***posición,*** se coincide con lo planteado por Fresno en que las oraciones más representativas se presentan en la *introducción* y *conclusiones* del documento. Pero se discrepa en considerar este criterio como un medidor de relevancia, debido a que la mayoría de las páginas web no se encuentran estructuradas en Introducción, Desarrollo y Conclusiones, luego, determinar estos límites resultaría una tarea compleja e inexacta.

En cuanto a los ***enfatizados***, Fresno considera que las etiquetas HTML que más captan la atención del lector son (..., <u>...</u>, ..., ..., <h1> - <h6> ...), sin establecer distinciones entre ellas.

Por lo que se propone separar a las etiquetas <h1> - <h6> del resto de los enfatizados, debido a que la función principal de estas etiquetas es adelantar el contenido del texto. Los encabezados ayudan al lector a construir un marco conceptual para la decodificación de un texto [(Lorch, R., 1993), (Sanchez, R. et al, 2001)], influyendo de forma muy positiva a la hora de llevarse una idea general del contenido del documento. Visto de esta forma, se consideró en un nuevo criterio heurístico, ***encabezados*** y el cual será tratado como un criterio heurístico independiente del resto de los enfatizados.

El atributo *alt* que trae consigo las etiquetas *<img>*, es usado para almacenar un texto alternativo que se corresponde con la imagen para ser utilizado en situaciones donde las imágenes no están disponibles, si por alguna razón la imagen no es cargada por el navegador, el texto alternativo será mostrado en su lugar, por lo que debe tener gran correspondencia con lo que se visualizará. Partiendo de la heurística de que las imágenes que aparecen en una páginas web está muy ligadas con su texto y que el texto alternativo de una imagen da una idea resumida del contenido de la misma, esta información se ha de tener en cuenta. Por lo que se considerara el ***texto alternativo*** como un criterio heurístico.

Concluyendo que los criterios heurísticos que aportarán mayor información para determinar la relevancia de un término son:

- Frecuencia en el contenido de la página.
- Aparición en encabezados de la página (<h1> – <h6>).
- Aparición con algún enfatizado dentro de la página(..., <u>...</u>, ..., ..., <big>...</big>, <cite>...</cite>, strong>...)
- Presencia en textos alternativos de imágenes presentes en la página.

2.2 Función de relevancia definida

En esta sección se propone una variantes de la *ACC* (Fresno, V., 2006), la cual utilizará los criterios heurísticos expuestos anteriormente y una de las ideas fundamentales tenidas en cuenta es que *mejor es combinar diferentes criterios que considerar cada una de ellos por separado* (Fresno, V., 2006)

Bastaría con definir funciones de captura para cada uno de los criterios a combinar y a continuación, ponderar cada una de ellas con un determinado peso. Así, es posible establecer un peso diferente para cada criterio, de modo que un criterio podría aportar más que otro en esta combinación final.

2.2.1 Definición de las funciones de captura para los criterios heurísticos considerados.

Frecuencia

La función de ponderación de la frecuencia de un determinado término t_i en el contenido de un documento d_j se expresa como:

$$f_{frec}(t_i, d_j) = f_{ij} / N_j$$

Siendo f_{ij} la frecuencia del término t_i en d_j y N_j la cantidad de rasgos presentes en el documento d_j. Esta definición asegura valores normalizados para la función, de forma que , $\sum_{1..k} f_{frec}(t_i, d_j) = 1$ donde k es el número de términos diferentes en d_j.

Encabezado

La función de ponderación respecto al criterio Encabezado para un determinado término t_i en el contenido de un documento d_j se expresa como:

$$f_{encab}(t_i, d_j) = f_{ij} / N_{encab(j)}$$

Siendo t_{ij} la frecuencia del rasgo t_i en el conjunto de términos que forman parte de los encabezados del documento d_j y N_{tit} el número total de términos que aparecen en encabezados del documento d_j. Al igual que en el caso del criterio frecuencia, se cumple la condición de normalización $\sum_{1..k} f_{encab}(t_i, d_j) = 1$ donde k es el número de términos diferentes presentes en el conjunto de los términos presentes en encabezados del documento.

Enfatizado

La función de ponderación respecto al enfatizado para un rasgo t_i en el contenido de un documento d_j se expresa como:

$$f_{enf}(t_i, d_j) = f_{ij} / N_{enf}(j)$$

Siendo f_{ij} la frecuencia del rasgo t_i en el conjunto de elementos enfatizados del documento d_j y N_{enf} el número total de rasgos enfatizados en el documento. Como en los casos anteriores, ésta función está normalizada, de modo que:

$\sum_{1...k} f_{enf}(t_i, d_j) = 1$, donde k es el número de términos diferentes presentes en el conjunto de elementos enfatizados.

Textos Alternativos

La función de ponderación respecto a los Textos Alternativos para un rasgo t_i en el contenido de un documento d_j se expresa como:

$$f_{textalt}(t_i, d_j) = f_{ij} / N_{textalt}(j)$$

Siendo f_{ij} la frecuencia del rasgo t_i en el conjunto de elementos presentes el conjunto de términos que componen los textos alternativos presentes en el documento y $N_{textalt}$ el número total de términos que componen los textos alternativos. Una vez más, ésta función está normalizada, de modo que:

$\sum_{1...k} f_{textalt}(t_i, d_j) = 1$, donde k es el número de términos diferentes presentes en el conjunto de términos presentes en los textos alternativos.

2.2.2 Establecimiento de los coeficientes de la combinación de criterios

En este epígrafe se determinan los coeficientes para la combinación lineal de los criterios presentados en la sección anterior. Los valores de los coeficientes fijan la importancia que se le quiere dar a cada criterio.

Para la estimación de los valores de los coeficientes $C_{frec}, C_{encab}, C_{enf}, C_{textalt}$ se consideró, que no debían ser valores específicos de una determinada colección, por lo que los valores que se establecieran en este punto no deberían de modificarse en posteriores experimentos desarrollados con otras colecciones.

El conjunto de valores para los coeficientes se estimó mediante el uso del *método estadístico frecuencial*.

Primeramente se realizó un muestreo sistemático con un plazo de 7 días, con el fin de crear una colección de 200 páginas web relacionadas con la temática "Lenguajes de Programación". Para la selección de las páginas se emplearon términos de consulta muy diversos en distintos motores. En ninguno de los casos se descargaba un número mayor de 10 documentos por servidor accedido, tratando de representar lo más posible la naturaleza heterogénea de la web.

Luego del muestreo sistemático, se aplicó el criterio de algunos especialistas en informática, concretamente especialistas en lenguajes de programación, para la selección de los 10 términos más relevantes por páginas. Quedando finalmente el conjunto de términos relevantes de la temática.

A continuación a dichos términos se le aplicaron las funciones de captura de cada criterio (visto en la sección anterior):

$$f_{frec}(t_i, d_j), f_{encab}(t_i, d_j), f_{enf}(t_i, d_j), f_{textalt}(t_i, d_j)$$

A partir de las medias calculadas para cada una de las funciones de captura se arrojaron los siguientes resultados (Figura 4):

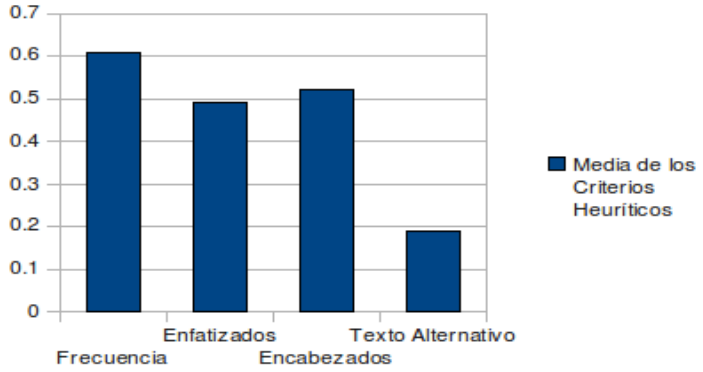

Figura 4: Media de los criterios heurísticos aplicados a 2000 términos relevantes

Restringiendo a que: $C_{frec} + C_{encab} + C_{enf} + C_{textalt} = 1$

Se pasó a fijar el valor de los pesos:

Partiendo de que, $0.61 + 0.52 + 0.49 + 0.19 = 100\%$ entonces se tiene que: 1.61=100%

Posteriormente se determinó que parte ese 100% representa cada media de frecuencia, quedando como sigue:

$0.61 = 33.7\%$, $0.52 = 28.7\%$, $0.49 = 27.0\%$, $0.19 = 10.40\%$

Por lo que, $0.337 + 0.287 + 0.27 + 0.104 = 1$

De donde se obtienen los coeficientes que cumplen con la restricción.

Quedando, $C_{frec} = 0.337$, $C_{encab} = 0.287$, $C_{enf} = 0.27$ y $C_{textalt} = 0.104$

Se define la función como sigue:

$$F(t_i, d_j) = 0.337\ f_{frec}(t_i, d_j) + 0.287\ f_{encab}(t_i, d_j) + 0.27\ f_{enfa}(t_i, d_j) + 0.104\ f_{textalt}(t_i, d_j)$$

2.3 Propuesta del algoritmo "HTMLExtractor"

Como ya se vio, el proceso de la Generación Automática de Extractos se puede dividir en tres fases: fase de ***análisis***, fase de ***transformación*** y fase de ***síntesis,*** dispuestas en ese mismo orden. HTMLExtractor seguirá este mismo esquema y cada una de estas fases se detalla en las secciones que continúan adaptadas a la propuesta.

2.3.1 Fase de Análisis

Durante la fase de análisis del proceso de Generación Automática de Extractos se construye una representación interna del documento fuente, o se puede ver esta fase como la implementación de una función que asocia a cada documento fuente una estructura de datos que lo representa.

Esta función, por su complejidad, es especificada por la composición de varias funciones que se encargarán, entre otras operaciones, de: extraer del código HTML toda la información referente a los criterios heurísticos definido, y a esta acción se le nombrará "obtención de información", identificar el idioma del documento, descomponer el texto en entidades sintácticas elementales tales como palabras, símbolos de puntuación, etc., además de identificar otros elementos del texto más complejos dependiendo del nivel de análisis lingüístico que se emplee y verificar la consistencia de estos elementos, por último identificación de oraciones.

2.3.2 Obtención de Información del Documento HTML

Durante la captura de información, se analiza el documento HTML para recoger información de carácter tipográfico presente en la página. En concreto, información relacionada con los criterios heurísticos propuestos.

Primeramente se recomienda obtener el objeto DOM del documento, luego, a partir esta representación arbólica resultará mucho más sencillo el análisis del HTML.

Para el criterio relacionado con los términos **enfatizados**, se obtienen todos los *nodos textos* del árbol que tengan como ancestro cualquiera de las etiquetas relacionadas con el criterio de enfatizado (“b”, ”u”, ”em”, ”i”, ”big”, ”cite”, ”strong”).

Luego, todos los términos diferentes de estos nodos textos formarán parte del conjunto de términos enfatizados.

Para el caso del conjunto de términos de **encabezados**, se tomarán todos los que aparecen en los *nodos textos* que tengan como ancestro algún nodo con la etiqueta: “h1”, “h2”,” h3”,” h4”,” h5”,” h6”.

Para el criterio relacionado con los términos **textos alternativos**, se buscan en el árbol primero los nodos con etiquetas “img”, a partir de éste, se verifica si tiene el atributo “alt” y se toma su valor. Posteriormente, todos estos textos formarán el conjunto de términos de *textos alternativos.*

Ya con toda esta información recopilada, se puede pasar a trabajar con el texto plano de la página o sea el texto sin etiquetas HTML. Para ello se puede usar diferentes técnica de extracción del texto, siempre y cuando no se tomen (o se tomen lo menos posibles) elementos que aparecen en la página pero que no se relacionan con el contenido de la misma, como le caso del footer[28], propagandas, etc.

Para estos casos se usaron diferentes heurísticas que permiten podar las ramas del DOM, pero sólo se presentarán algunas de las usadas. Usualmente la propaganda se encuentra sobre etiquetas *iframe* que obtienen su contenido desde URLs las cuales son posibles identificar mediante el uso de expresiones regulares y comparación con listas que están disponibles sobre estas URLs. Los footer se encuentra usualmente contenidos sobre nodos que en su atributo class o id

[28] Pie de la página web.

contienen esta palabra, siendo posible su poda sin mayores complicaciones. También es conveniente eliminar del DOM, todas las ramas que tengan como raíz etiquetas de encabezados, de esta forma se evita que en el resumen de la página aparezcan los títulos, los cuales resultarían elementos inadecuados.

2.3.3 Identificación de Idioma.

Identificar el idioma en el que está escrito el documento que se quiere resumir es considerada una tarea de suma importancia, pues a pesar que el algoritmo propuesto es independiente del idioma en el que fue escrito el documento, sí es necesario conocerlo de antemano para poder realizar una representación adecuada del mismo.

La identificación automatizada del idioma en un texto puede describirse como: proceso de asignación de un documento a la categoría (idioma) con la cual tiene un mayor número de características en común. La asignación o no, del documento a la categoría final, puede estar regida por valores empíricos establecidos con anterioridad.

Varios de los sistemas comerciales para la GAR encontrados en la literatura, realizan el proceso de identificación del idioma, pero los métodos empleados para lograr este fin, han sido descartados pues no se tiene ninguna referencia sobre su código. Por este motivo surge la necesidad de proponer un método para esta necesaria tarea, el cual contribuirá en el proceso de Generación Automática de Resúmenes.

Algoritmo para la identificación del idioma basado en "StopWord"

Se denominan "stopwords" a palabras sin contenido semántico, es decir, que desde un punto de vista no-lingüístico contienen poca información. Los términos de la lista de "stopwords" están carentes de todo significado a la hora de recuperar información, como, por ejemplo, el artículo "la" no posee ninguna funcionalidad en la recuperación de documentos, ya que en todos los documentos este término

aparecerá de forma casi segura y no resalta nada el contenido del documento almacenado (basado en la Ley de Zipf (Zipf, G. 1949)). Esta característica las hace ser descartadas en tareas del Procesamiento del Lenguaje Natural como la recuperación de información, la categorización automatizada de texto, generación automática de resúmenes entre otras. Sin embargo pueden ser útiles para la identificación de idiomas, pues son comúnmente muy usadas dentro de un texto. En el conjunto de "stopwords" para un determinado idioma se pueden encontrar a sus artículos, preposiciones, conjunciones, entre otras.

Para cada idioma se han confeccionado listas de "stopwords" las que son muy usadas en la solución de problemas computacionales relacionados con la minería de texto.

El algoritmo propuesto consiste en determinar con que frecuencia aparecen las "stopwords" de los diferentes idiomas en el documento, clasificándose en el idioma que más "stopwords" tuvo presentes en el mismo.

Ahora con más detalle:

Cada idioma (idiomas que se tengan sus listas de "stopwords") le da una votación al documento de la forma que sigue:

$$V_{idioma_{jd}} = \sum Stopw_{kjd} / CantStopw_{d}$$

Siendo $V_{idioma_{jd}}$ el voto que emite el idioma j al documento d, $Stopw_{kjd}$ las veces que aparece la "stopword" k perteneciente al idioma j en el documento d y $CantStopw_{d}$ es la cantidad de "stopwords" presentes en el documento.

Luego que cada idioma haya emitido su voto, se aplica la siguiente regla para definir el idioma al que pertenece el documento.

1. Se selecciona el idioma que mayor voto emitió, luego se concluye que el documento pertenece a dicho idioma si y sólo si:
    - La votación de dicho idioma supera un determinado umbral.

- La diferencia entre la votación de dicho idioma con el resto de las votaciones de los demás idiomas también supera un determinado umbral.

Los valores de los umbrales se determinaron a partir del siguiente experimento.

Con el empleo de las listas de "stopwords" utilizadas en el CLEF [29] se determinó que voto emitían los idiomas (español, inglés e italiano) a una serie de documentos de los cuales ya se conocía su idioma con antelación.

Se analizaron los votos emitidos por el idioma correspondiente a cada uno de los documentos, notándose, que para todos los casos, el idioma correcto emitía un voto mayor de 0.8, de esta forma quedó establecido para nuestro algoritmo este umbral. También se pudo apreciar que para el caso de los idiomas español e italiano, los votos emitidos para algunos documentos, superaban en ambos casos dicho umbral, esto se debió al gran número de "stopwords" coincidentes en ambas listas. Quedando establecida una segunda restricción, la cual consistió en que además de una votación superar el umbral establecido, no se puede definir dicho idioma como el correcto para el documento, mientras no se compruebe que la diferencia entre el voto de este idioma con el resto de las votaciones también superaran un segundo umbral, el cual a partir de estas pruebas quedó establecido con un valor de 0.15.

A partir de numerosas pruebas realizadas con el método propuesto se puedo estimar, que el mismo presenta una efectividad superior al 0.87, lo cual es considerado como un valor elevado.

2.3.4 Análisis léxico, lematización e identificación de oraciones.

Se propone el empleo del etiquetador léxico-morfológico TreeTagger, desarrollado en el Instituto de Lingüística Computacional de la Universidad de Stuttgart. Su

[29] http://clef-campaign.org/

selección se debió además de los magníficos resultados que se obtienen, por el gran número de idiomas con el que trabaja, entre los que se encuentran: Alemán, Inglés, Francés, Italiano, Español, Griego, Francés Antiguo, Neerlandés, Ruso, Búlgaro, Greco y Chino.

La función fundamental del TreeTagger es desambiguar las categorías gramatical de las palabras, para lo que se basa en Modelos Ocultos de Markov[30] y en árboles de decisión[31].

EL TreeTagger recibe como entrada el documento en texto plano, y como la salida en el fichero resultante por cada término incluye: la palabra analizada, seguido por la etiqueta de acuerdo a su categoría gramatical y por último la raíz o lema de la misma.

Por ejemplo, para esta entrada: “The TreeTagger is easy to use”. La salida sería como se muestra en la figura 5.

Word	Category	Lemma
The	DT	the
TreeTagger	NP	TreeTagger
Is	VBZ	be
easy	JJ	easy
To	TO	to
use	VB	use
.	SENT	.

Figura 5: Ejemplo de salida del Treetagger.

A partir de la salida del TreeTagger se pudo separar el documento en oraciones, o sea todos los términos comenzando por el primero hasta aquella entidad sintáctica que tuviese la etiqueta que indica fin de oración, corresponden a la primera oración. Los próximos términos hasta la siguiente etiqueta de fin, corresponden a

[30] Es un modelo estadístico en el que se asume que el sistema a modelar es un proceso en la cual la probabilidad de que ocurra un evento depende del evento inmediato anterior.

[31] Es un modelo de predicción utilizado en el ámbito de la Inteligencia Artificial

la segunda oración y así sucesivamente, hasta encontrar a todas las oraciones de documento.

2.3.5 Identificación de estructuras multipalabras (nombres propios).

Una técnica muy utilizada es la identificación de estructuras multipalabras, como son, las frases sustantivas, nombres propios de personas, lugares, organizaciones, etc.

Se propone este método para identificar los nombres propios que aparecen en el documento, ya que los nombres propios, por su propia naturaleza no forman parte de los diccionarios o repertorios léxicos. Para su identificación se tuvo en cuenta la siguiente proposición como heurística:

"Toda secuencia de sustantivos identificada por el TreeTagger constituye un nombre propio".

Luego bastaría con encontrar estas secuencias de sustantivos y representarlas como un solo término del documento.

2.3.6 Eliminación de "StopWords"

Cada lema o raíz es comprobado previamente, verificándose su presencia en la lista de "stopwords" del idioma al que pertenece el documento, luego es descartado el lema de la representación final del documento si se encuentra en dicha lista.

Con la eliminación de las palabras vacías se logra una reducción del documento entre un 30 y un 50% (Luhn H., 1958).

2.3.7 Representación del documento

Una de las técnicas más empleadas, para lograr tal representación, ha sido el Modelo de Espacio Vectorial (visto en Capítulo 1). Este modelo algebraico permite

la representación abstracta de documentos, escritos en lenguaje natural, mediante el uso de vectores de términos en un espacio T-dimensional, donde T es la cantidad de rasgos representativos de cada objeto. Cada rasgo debe tener asociado una relevancia o peso, que expresa su importancia en el objeto y se calcula por medio de una Función de Ponderación.

Se representó el documento como un vector de *n* componentes donde *n* es el número de raíces diferentes presentes en el mismo, o sea cada raíz corresponde a una componente diferente del vector, almacenándose por cada una, la cantidad de términos presentes en el documento que son variaciones morfológicas de ella.

Luego de construido el vector del documento (*vector patrón*), se le asocia un peso a cada componente con el empleo de la función de relevancia propuesta. O sea, se determina para cada raíz:

- frecuencia de aparición en el documento.
- frecuencia de aparición en el conjunto de términos que componen los encabezados del documento.
- frecuencia de aparición en el conjunto de términos enfatizados en el documento.
- frecuencia de aparición en el conjunto de términos que componen los textos alternativos de las imágenes.

Luego se combina toda esta información como se vio en el epígrafe anterior.

Finalmente a cada oración del documento se le asocia un vector, el cual será el mismo que representa al documento (vector patrón), con la variación que en las componentes donde el lema asociado no aparece en la oración se le asigna un valor de relevancia igual a 0.

2.4 Fase de Transformación

La fase de transformación es la encargada de, partiendo de la representación interna del documento fuente realizada en la fase de análisis, aplicar los métodos

de puntuación, selección y clasificación de las sentencias como pertenecientes al extracto o no.

Se asumió la siguiente heurística para asignarle una puntuación a las sentencias:

¨La importancia de una oración estará determinada por la importancia de los términos que la componen¨.

De donde se asume que la puntuación de las sentencias es igual a la sumatoria de la relevancia los términos que la componen. Definiéndose como:

$$W_k = \sum_{i=1}^{l} r_i$$

Donde, W_k es el valor de importancia de la sentencia k, *l* la cantidad de términos de la sentencia k y r_i es el valor de relevancia del término i en el contenido del documento.

Cuando todas las sentencias del documento tengan una puntuación asignada, se pasa a la selección de aquellas que van a formar parte del extracto. Para esto se proponen los siguientes métodos:

Método umbral por puntuación: Este método consiste en seleccionar aquellas sentencias que superan un umbral de puntuación predeterminado. Este umbral se determinó de forma empírica.

Método umbral para condensado: Como se vio en el capítulo anterior la razón de compresión del extracto es un elemento muy importante a tener en cuenta a la hora de generar el extracto del documento, este método consiste en ir seleccionando las sentencias más puntuadas mientras que la razón de compresión del extracto sea menor que un determinado umbral.

Este umbral se fijó a partir de estudios realizados donde aseveran que el tamaño del extracto ideal es aquel representa el 25% del texto que le da origen (Esaú-Villatoro, Tello, 2007) y (Lin, C. y Hovy, E. ,2003), por lo que el umbral de condensado seleccionado fue de 0.25

2.5 Fase de Síntesis

En esta fase se procede, a construir en lenguaje natural el resumen obtenido partiendo de los resultados de la aplicación de los métodos de selección analizados anteriormente.

Los elementos seleccionados en este momento, se encuentran representados en una forma poco entendible para los humanos.

Durante esta fase se convierte el extracto, de su representación interna a una representación en lenguaje natural. O sea, se muestran las oraciones que formaron parte del extracto tal y como aparecen en texto original. Además deben disponerse en el mismo orden en que aparecen originalmente.

Al finalizar la fase de síntesis ya se pude contar con un extracto completamente comprensible por los humanos.

Para contribuir con la mejorar la coherencia de los extractos generados, durante esta fase pudiesen aplicar algún método de revisión (Capítulo 1).

2.6 Conclusiones

En este capítulo se presentó un método efectivo para construir resúmenes a partir de un documento HTML. Es un algoritmo NO supervisado, incluye nuevas métricas en el proceso de ponderación de frases, toda la información necesaria para determinar la relevancia de los términos será extraída del propio código HTML.

A diferencia de otros métodos para la Generación Automática de Extractos encontrados en la literatura, se realizó una asignación semántica a determinados etiquetas del vocabulario HTML, se define una función de relevancia fundamentada en una combinación lineal de criterios heurísticos, logrando extrapolar información relativa a la intención del autor en el momento de la creación de la página.

La novedad del método presentado para la construcción de resúmenes es, precisamente, la explotación de la información de marcado de los documentos HTML que, combinado con diferentes técnicas y heurísticas, produce buenos resúmenes como será mostrado en el capítulo 3. Este método es robusto, independiente del dominio y puede ser aplicado a documentos escritos en varios idiomas.

Capítulo 3. Evaluación

Aunque la mayoría de los trabajos de generación automática de resúmenes tienen una componente teórica importante, pues suelen establecer hipótesis o proponer técnicas, son evaluados de una manera formal y rigurosa.

En este capítulo se exponen los resultados obtenidos en la evaluación. Para ello primeramente se implementaron dos variantes del algoritmo propuesto (HTMLextractorV1 y HTMLextractorV2), en HTMLextractorV1 se utilizó durante la Fase de Transformación el método **umbral por puntuación** y HTMLextractorV2 el método **umbral para condensado**.

Para tener un mayor grado de conocimiento sobre la calidad de los resúmenes generados por HTMLExtractor, se incluye en las comparativas los resultados obtenidos por tres sistemas generadores de extractos: *Copernic Summarizer*[32]*,Pertinence Summarizer*[33] *y Swesum*[34] plenamente admitidos y utilizados como referencia para la evaluación por la comunidad internacional.

3.1 Descripción de los Generadores de Extractos usados para la evaluación.

Copernic Summarizer, es una herramienta multilingüe comercial de generación de resúmenes a partir de textos o páginas web con el objetivo de disminuir el tiempo de acceso del usuario a la información importante. Obtiene los conceptos y frases clave a partir de una razón de compresión dado. Se integra fácilmente en procesadores de texto, navegadores y clientes de correo. Los algoritmos y técnicas usadas no son públicos, sólo se revela que usa “sofisticados” algoritmos estadísticos y lingüísticos, eliminando automáticamente contenido y texto irrelevante.

[32] http://www.copernic.com/en/products/summarizer/
[33] http://www.pertinence.net/index_en.html
[34] http://swesum.nada.kth.se/index-eng-adv.html

Pertinence Summarizer, es una herramienta comercial de generación de resúmenes que se basa en técnicas extractoras, mediante el procesamiento de la relevancia (pertinencia la denominan ellos) de cada sentencia, tomando en cuenta posibles palabras clave, diccionarios de términos y marcadores lingüísticos genéricos. Es multilingüe y es usada la versión online para la evaluación.

Swesum, Es un generador de resúmenes multilingüe. Utiliza múltiples aspectos para valorar las sentencias, como su posición o valor numérico en un esquema, de modo que las sentencias iniciales tienen un peso adicional, así como las numeradas. Para la evaluación se ha usado la versión online, con las opciones por defecto.

Estos productos cuentan actualmente con prestigio y popularidad en el mercado mundial. Los detalles concretos de los algoritmos que usan no son públicos, pero por su fácil acceso son muy usados por la comunidad de investigadores en tareas de evaluación.

3.2 Descripción del Corpus para la Evaluación

El corpus utilizado para la evaluación de todos los algoritmos fue la colección "HTMLcollection" la cual fue creada por el grupo investigativo "Inteligencia Artificial Aplicada" de la Universidad de las Ciencias Informáticas en el año 2008.

HTMLcollention está formada por 100 documentos HTML escritos en diferentes idiomas y sobre diferentes dominios temáticos. Además se encuentran etiquetados de acuerdo a la cantidad de información de marcado que contienen (nivel alto de marcado, nivel medio de marcado, y nivel bajo de marcado).

Para cada uno de los documentos del corpus, existe un resumen (resumen ideal), elaborado por expertos en las materia que aborda dicho documento, además todos los resúmenes elaborados representan el 25% del documento original.

3.3 Evaluación de la Calidad de los Extractos

Como se vio en el epígrafe “Evaluación de la calidad de los resúmenes" del Capítulo 1, los métodos de evaluación de resúmenes construidos de manera automática se dividen en métodos intrínsecos y extrínsecos. En este capítulo se realiza la evaluación de resúmenes generados usando ambos métodos.

3.3.1 Evaluación Intrínseca de los Extractos

Los métodos Intrínsecos evalúan a los resúmenes como entes individuales, generalmente comparándolos con un resumen de referencia. Todos los resúmenes generados por los sistemas comerciales y HTMLExtractorV2 fueron evaluados contra los resúmenes de referencias, para la realización de dicha evaluación, se decidió utilizar **ROUGE** como medida de evaluación automática. **ROUGE**, desarrollado por el *Informatio Science Institute en la University of Southern California,* es una herramienta automática que compara un resumen generado por un sistema automatizado con uno o más resúmenes de referencias, los llamados ‘modelos'. **ROUGE** ha sido usado desde 2004 y hasta la fecha, en las Conferencias DUC como herramienta de evaluación en las competiciones y es un estándar asumido por la comunidad internacional, razón por la que fue usada para la evaluación.

En (Lin, 2003) se ha concluido que ciertas métricas de ROUGE se correlacionan mejor con los criterios humanos que otras dependiendo en la tarea que se esté evaluando, en el caso de la generación automática de resúmenes de un sólo documento, Lin encontró que las puntuaciones de ROUGE-N (N = 1. . . 4), ROUGE-L y ROUGE-W podrían considerarse las mejores, siendo desde entonces las más usadas en este campo.

ROUGE-N calcula el promedio de emparejamiento de n-gramas entre el conjunto de resúmenes a evaluar y el conjunto de resúmenes de referencia pertenecientes al corpus. Estas métricas brindan el valor de cobertura (recall) de los extractos generados.

ROUGE-L calcula la subcadena más larga que tienen en común ambos conjuntos. **ROUGE-W** es muy similar a medida L, con la salvedad de que memoriza los tamaños de los emparejamientos consecutivos, para quedarse con el mayor de ellos.

Así, para todas las medidas de **ROUGE**, mientras más alto es su valor, mejor es el rendimiento del algoritmo, ya que altos valores en las medidas indican mayor superposición entre el conjunto de resúmenes generados automáticamente y sus resúmenes ideales.

Para la evaluación del algoritmo propuesto cada herramienta generó el resumen a 75 documentos presentes en el corpus, fijando para todos, como razón de compresión el 25% del tamaño del documento original.

ROUGE es una herramienta que permite evaluaciones parametrizadas en función de ciertos valores, que orientan la tarea al tipo especial de documento original.

Se emplearon dos evaluaciones, una con los parámetros que se han usado DUC y otra con los valores por defecto de ROUGE:

Evaluación con los parámetros del DUC

Para esta tarea, el método de evaluación usado es ROUGE-N(N=1 y N=2), el puntaje que se obtiene es una medida que indica, qué tanto de los n-gramas utilizados por los expertos para la creación de su propio resumen, se obtienen con el algoritmo propuesto.

En esta configuración se utilizan corte de palabras a su raíz (stemming) y manteniendo “stopwords” (listas de palabras a ignorar). La versión usada para la evaluación fue ROUGE-1.5.5[35]

La prueba estadística aplicada a los resultados de las métricas fue el Análisis de Varianza (ANOVA) unifactorial. Para discriminar las medias se aplicó la dócima de

[35] La versión utilizada en este trabajo,ROUGE-1.5.5,se encuentra disponible en: http://haydn.isi.edu/ROUGE

comparaciones múltiples de Diferencia Mínima Significativa de Fisher. Se utilizó el paquete estadístico (STATGRAPHICS Centurion, 2006).

Se evalúa el algoritmo HTMLExtractorV2 con el objetivo determinar el comportamiento de HTMLExtractorV2 en documentos con diferentes niveles de marcado, para ello, se seleccionaron 25 documentos por cada nivel.

Tabla 1: Resultados de HTMLExtractorV2 con las métricas ROUGE-1, ROUGE-2.

NIVEL DE MARCADO	MÉTRICAS	
	ROUGE 1	ROUGE 2
Bajo	0,397c	0,377b
Medio	0,419b	0,375b
Alto	*0,453a*	*0,442a*

Nota: *Se asume que valores con letras iguales en columnas no difieren estadísticamente según dócima de Diferencia Mínima Significativa de Fisher*

Después de analizar los resultados arrojados (Tabla 1), se puede apreciar que HTMLExtractorV2 obtuvo los mejores resultados (muy alentadores) en documentos con un nivel alto de marcado, además se pudo notar que entre una métrica y otra, no hubo diferencias; es decir que según Dócima de Diferencia Mínima Significativa de Fisher no hay diferencia significativa. Para documentos con nivel de marcado medio los resultados fueron un poco menores que para los documentos con nivel alto, existiendo diferencia significativa en la métrica ROUGE-1, pero sus resultados fueron estables para las dos métricas. Para documentos con nivel bajo de marcado HTML los resultados fueron más desfavorables con diferencias significativas con el resto de los documentos.

De lo que se puede concluir que HTMLExtractor, es un algoritmo en que la calidad de los resúmenes estará muy ligada a la cantidad de información de carácter tipográfico presente en el documento, o sea, entre más información el autor de la página transmita a través del lenguaje HTML, mayor nivel de informatividad tendrán los resúmenes generados.

Los resultados de las evaluaciones de los extractos generados por Copernic Summarizer , Pertinence Summarizer , Swesum y HTMLExtarctorV2(Tabla 2).

Tabla 2: Resultados de las métricas ROUGE-1 y ROUGE-2

ALGORITMOS	MÉTRICAS	
	ROUGE 1	ROUGE 2
Copernic	0,422a	0,422a
Pertinence	0,360b	0,328c
Swesum	0,350b	0,337c
HTMLExtractor	0,423a	0,398b

Nota: *Se asume que valores con letras iguales en columnas no difieren estadísticamente según dócima de Diferencia Mínima Significativa de Fisher*

Se puede observar que para la métrica ROUGE-1 los mejores resultados fueron arrojados por Copernic y HTMLExtractorV2 sin diferencias estadísticas significativas entre sus resultados. Para la métrica ROUGE-2 hay cierta ventaja por parte de los resultados de Copernic con respecto al resto de los algoritmos. No obstante, a partir del valor de la evaluación de ROUGE-1 a HTMLExtractorV2 se pudo inferir que el contenido informativo presente en los extractos generados se puede considerar elevado.

Los resultados de Pertinence Summarizer y Swesum resultaron discretos con respecto al resto de los algoritmos.

Evaluación con los parámetros ROUGE por defecto.

Se decidió realizar una evaluación que recoja y presente las métricas de ROUGE-L y ROUGE-W. Estas métricas pueden indicar en que medida se están utilizando las palabras correctas en el orden correcto.

Tabla 3: Resultados de las métricas ROUGE-L y ROUGE-W

ALGORITMOS	MÉTRICAS	
	ROUGE L	ROUGE W
Copernic	0,428a	0,423a
Pertinence	0,327d	0,318c
Swesum	0,337c	0,302d
HTMLExtractor	0,396b	0,355b

Nota: *Se asume que valores con letras iguales en columnas no difieren estadísticamente según dócima de Diferencia Mínima Significativa de Fisher*

Las mejores puntuaciones fueron arrojadas por *Copernic Summarizer (Tabla 3)*, seguidas por los resultados de HTMLExtractor-V2 el cual se comportó de forma estable con las diferentes métricas. Este último con diferencias significativas con respecto a Swesum y a Pertinence.

Los resultados alcanzados por HTMLExtractor-V2, aunque no fueron los mejores, parecen indicar que el camino tomado y la estrategia de solución del problema son adecuados.

Ahora, si se analizan los resultados de Copernic Summarizer, en las diferentes métricas se puede notar que a medida que los N-gramas aumentan de tamaño, los resultados mejoraron considerablemente, aunque no se conoce la causa de dicho comportamiento, la lógica indica que los valores obtenidos por ROUGE-L y ROUGE-W deben ser de menor puntaje que los obtenido por ROUGE-1 y ROUGE-2, según (Esaú-Villatoro, T., 2007) un humano siempre hará uso de sus conocimientos del "mundo" para la creación de sus resúmenes, por lo que existirá en el resumes un número considerado de palabras que no aparecen en el documento original.

3.3.2 Evaluación Extrínseca de los Extractos.

Para evaluar la eficiencia y desempeño de los resúmenes se utilizó el método extrínseco *Lectura de comprensión.*

Para la ejecución de este experimento se tomaron extractos generados por *Copernic Summarizer* (obtuvo los mejores resultados con ROUGE-L y ROUGE-W) y extractos generados por HTMLExtractor-V1. Estos extractos fueron provenientes de documentos con un nivel medio de marcado, escritos en el idioma español y sobre la temática "Programación".

Descripción del experimento y resultados

Primeramente se necesitó la intervención de tres personas, de ellas dos fueron las encuestadas y una tercera sirvió de moderador. Para la selección de estas persona se tuvo en cuenta que tuviesen un nivel bajo de conocimientos de programación.

Se repartió a cada uno de los encuestados 30 de los resúmenes generados por los métodos a evaluar. Al encuestado número 1 se les asignaron los resúmenes generados por *Copernic* y al encuestado número 2 los resúmenes generados por HTMLextractorV1.

Luego, el moderador a partir de los documentos originales les realizó preguntas relacionadas con dichos documentos (el número de preguntas variaba en dependencia del tamaño del documento) marcando para cada documento el número de respuestas correcta e incorrectas de cada encuestado.

Los resultados de este experimento (Figura 6) se puede apreciar en primer lugar, que como promedio el número de respuesta incorrectas supera las respuestas correcta para ambos algoritmos, por lo que se puede deducir que gran parte de la información importante de los documentos no fue representada en los extractos, influyendo de forma negativa en la eficiencia y desempeño de los mismos.

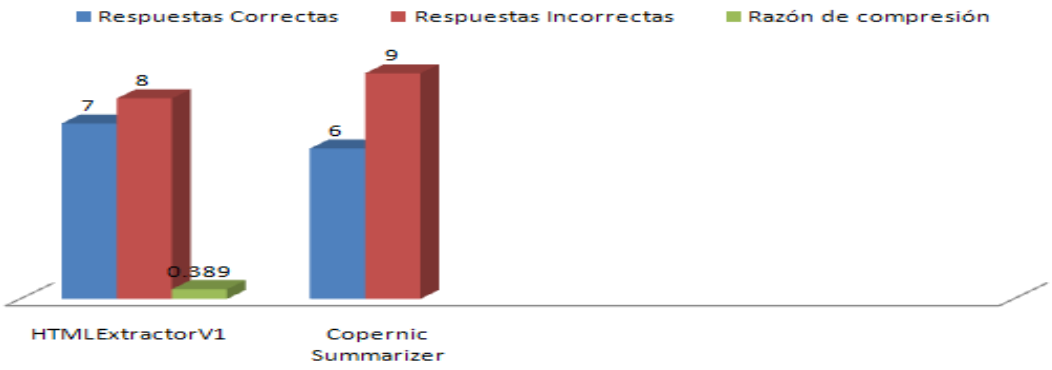

Figura 6: Resultados de HTMLExtractorV1 y Copernic Summarizer usando el método de Lectura de comprensión

HTMLExtractorV1 superó en una respuesta correcta a Copernic Summarizer que aunque la diferencia no fuese significativa se podía pensar que existiese cierta ventaja a favor de HTMLExtractorV1, dado esto se pasó a analizar la razón de compresión de los resúmenes generados por HTMLExtractorV1, para no fundir expectativas erronas ante este resultado, pues podía ser que para estos casos los resúmenes generados hubiesen sido más extensos y resultó ser cierta nuestra sospecha, pues la razón de compresión en esta variante del algoritmo en la mayoría de los resúmenes generados superaba la razón de compresión del 0.25, resultando un elemento negativo.

Este resultado sugirió que el valor umbral seleccionado para la clasificación de las sentencias debería ser modificado.

3.4 Conclusiones

En el capítulo que concluye se evaluó la calidad del algoritmo propuesto tanto de forma extrínseca como intrínseca, para ello, se generaron extractos por el algoritmo HTMLExtractor (algoritmo propuesto) y se compararon con los extractos generados por las herramientas comerciales *Copernic Summarizer, Pertinence Summarizer y Swesum.*

Las evaluaciones realizadas indican que la estrategia de resolución del problema es adecuada. Pero se ha de ser muy prudente en la evaluación de los resultados obtenidos debido a los discretos resultados obtenidos por generadores de resúmenes prestigiosos, como lo son: *Pertinence Summarizer y* Swesum.

Conclusiones Generales

De los resultados obtenidos en esta tesis se arriba a las siguientes conclusiones:

- Del estudio realizado sobre los algoritmos de Generación Automática de Resúmenes, resultó que ninguna de sus aproximaciones emplean la información de marcado presentes en las páginas web.
- Se definió una función de carácter local para determinar la relevancia de los términos en el contenido de un documento HTML, basado en una combinación lineal de criterios heurísticos.
- Se desarrolló un método para la identificación del idioma de los documentos, que puede emplearse en tareas donde se necesite de esta información.
- Se desarrolló un algoritmo para la generación automática de extractos de un documento HTML que aprovecha la información de marcado presente en la página web que podrá ser implementado por cualquier sistema que requiera de esta información.
- Se compararon los extractos generados por el algoritmo desarrollado con los generados por sistemas comerciales de gran popularidad, obteniéndose resultados significativamente superiores en la métrica ROUGE-1 y para el resto de las métricas sólo fue superado por el sistema Copernic Summarizer.

Recomendaciones

- Analizar la posibilidad de incorporar otros criterios heurísticos aplicados al proceso de lectura de páginas web que contribuyan a determinar la relevancia de un término en el contenido de una página.
- Considerar los enfatizados que se logran mediante el uso de CSS.
- Aplicar métodos de resolución de anáforas para mejorar la calidad de los extractos generados.
- Profundizar en el análisis de selección de los umbrales usados.
- Valorar el uso de Bases de Datos, como la WordNet, para reducir las ambigüedades en el sentido de las palabras.
- Incorporar el algoritmo desarrollado a proyectos dedicados al análisis de los contenidos de Internet que se ejecutan en la UCI.

Referencias Bibliográficas

1. **Amigó, E. (2006)**. Síntesis de información: desarrollo y evaluación de un modelo interactivo. Madrid: Universidad Nacional de Educación a Distancia. http://www.kriptia.com/MATEMATICAS/CIENCIA_DE_LOS_ORDENADORES/1#117897

2. **Amigó, E. et al (2005)**. QARLA: A framework for the evaluation of text summarization systems. En Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics. Michigan. http://nlp.uned.es/pergamus/pubs/articuloACL2005V5.pdf

3. **Barzilay, R. y Elhadad, M. (1999)**. Using Lexical Chains for Text Summarization. En Advances in Automatic Text Summarization, MIT Press. Cambridge, ISBN: 0-262-13359-8 p: 111–121. http://scholar.google.com.cu/scholar?hl=es&q=Text+Summarization+Using+Lexical+Chains&btnG=Buscar&lr=&as_ylo=&as_vis=0

4. **Berners-Lee, T. et al (1992)**. World-Wide Web: The Information Universe. Electronic Networking: Research, Applications and Policy. 2(1). http://www.w3.org/History/1992/ENRAP/Article_9202.ps

5. **Brandow, R. et al (1995)**. Automatic Condensation of electronic publications by sentence selection. Information Processing and Management. *31(5):675-685*

6. **Brunn, M. et al (2001)**. Text Summarization Using Lexical Chains. Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.15.9903&rep=rep1&type=pdf

7. **Cerezo, A. (1994)**. Texto, contexto y situación.Guía para el desarrollo de las competencias textuales y discursivas. Barcelona: Octaedro.

8. **Doran, W. et al (2004).** Assessing the impact of lexical chain scoring methods and sentence extraction schemes on summarization. Proceedings

of the 5th International conference on Intelligent Text Processing and Computational Linguistics CICLing-2004.
http://www.cicling.org/2004/Abstracts/29450622.pdf

9. **Gallo, D. (2006)**. Blindlight. Tesis(Doctoral) p:500

10. **Edmundson, H. (1969).** New methods in automatic extracting. Journal of the Association for Computing Machinery, 16(2):264:285.
http://eprints.kfupm.edu.sa/53107/1/53107.pdf

11. **Endres-Niggemeyer, B. (1998).** Summarizing Information. Journal of Information Processing & Management. 36(2):336-338.

12. **Erkan, G. y Radev, D. (2004a).** LexPageRank: Prestige in multi- document text summarization. Proceedings of EMNLP.
http://www.aclweb.org/anthology/W/W04/W04-3247.pdf

13. **Erkan, G. y Radev, D. (2004b).** "The University of Michigan at DUC 2004". DUC. http://www-nlpir.nist.gov/projects/duc/pubs/2004papers/umich.erkan.ps

14. **Esaú-Villatoro, Tello. (2007).** Tesis de Maestría: Generación Automática de Resúmenes de Múltiples Documento. Tesis (Maestría). España: Instituto Nacional de Astrofísica.
http://ccc.inaoep.mx/~villasen/tesis/TesisMaestria-EsauVillatoro.pdf

15. **Fresno, V. (2006).** Representación Autocontenida de Documentos HTML: una propuesta basada en Combinaciones Heurísticas de Criterios. Tesis (Doctoral). Móstoles, España: Universidad Rey Juan Carlos, Escuela Superior de Ciencias Experimentales y Tecnología, Departamento de Ingeniería Telemática y Tecnología Electrónica. p: 271.
http://www.escet.urjc.es/~vfresno/phd_sp.html

16. **Fuentes, M. et al (2003)**. "Mixed approach to headline extraction for DUC 2003". Proceedings of DUC 2003.
http://www-lpir.nist.gov/projects/duc/pubs/2003papers/ugirona.pdf

17. **Fukumoto, F. et al (1997).** An Automatic Extraction o f Key Paragraphs Based on Context Dependency. Applied Natural Language Conferences.

p:291 - 298
http://www.aclweb.org/anthology/A/A97/A97-1043.pdf

18. **Fukumoto, F. et al (2000)**. Extracting Key Paragraphs Based on Topic and Event Detection - Towards Multi-Document Summarization. Proceedings of the Workshop on Automatic Summarization at the 6th Applied Natural Language Processing Conference and the 1st Conference of the North American Chapter of the Association for Computational Linguistics. http://www.aclweb.org/anthology/W/W00/W00-0404.pdf

19. **Piat, G. et al (1997).** How to appreciate the quality of automatic text summarization. In Proceedings of the ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization. p: 25–30. http://www.lalic.paris4.sorbonne.fr/articles/1997-1998/Minel/Madrid.pdf

20. **Hall, M. (2008).** Introduction to the Document Object Model URL http://www.brainjar.com/dhtml/intro/

21. **Hardy, H. et al (2001).** Cross-Document Summarization by Concept Classification. Document Understanding Conference, a SIGIR'01 workshop. https://portal.acm.org/poplogin.cfm?dl=GUIDE&coll=GUIDE&comp_id=564399&want_href=delivery.cfm%3Fid%3D564399%26type%3Dpdf%26CFID%3D61926075%26CFTOKEN%3D31280282&CFID=61926075&CFTOKEN=31280282&td=1257746618226

22. **Hovy, E. (1999).** Automated Text Summarization in SUMMARIST. En Advances in Automatic Text Summarization. p: 81-94. http://acl.ldc.upenn.edu/W/W97/W97-0704.pdf

23. **Hovy, E. et al (2005).** Evaluating DUC 2005 using basic elements. Proceedings of the Document Understanding Conferences (DUC). http://duc.nist.gov/pubs/2005papers/usc-isi-zhou2.pdf

24. **Jing, H. y McKeown, K. (2000).** Cut and Paste-Based Text Summarization. Proceedings of the 6th Applied Natural Language Processing Conference and the 1st Meeting of the North American. p: 178-185. http://www.ldc.upenn.edu/acl/A/A00/A00-2024.pdf

25. **Spyropoulos y Karkaletsis (2005).** Information Extraction and Summarization from Medical Documents. Artificial Intelligence in Medicine 33(2):107-198

26. **Kupiec, J. et al (1995).** A Trainable Document Summarizer. En Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. p: 68-73. http://www.icst.pku.edu.cn/course/TextMining/0708Spring/%E5%8F%82%E8%80%83%E6%96%87%E7%8C%AE/0901%20A%20Trainable%20Document%20Summarizer.pdf

27. **Drummey, K. et al (2000).** A comparison of rankings produced by summarization evaluation measures. Proceedings of the Workshop on Automatic Summarization at the 6th Applied Natural Language Processing Conference and the 1st Conference of the North American Chapter of the Association for Computational Linguistics. http://www.aclweb.org/anthology/W/W00/W00-0408.pdf

28. **Lawrence, S. y Giles, C. (1999)**. Accessibility of information on the web. Nature. http://clgiles.ist.psu.edu/papers/Nature-99.pdf

29. **Lita, L. et al (2001).** Learning Within-Sentence Semantic Coherence. http://www-2.cs.cmu.edu/~llita/papers/lita.coherence-emnlp2001.pdf

30. **Lin, C. (1999).** Training a selection function for extraction. In Proceedings of the 8th Annual International ACM Conference on Information and Knowledge Management CIKM. http://research.microsoft.com/en-us/people/cyl/cikm99.pdf

31. **Lin, C. y Hovy, E. (2003)**. Automatic evaluation of summaries using n-gram co-occurrence statistics. Proceedings of HLTNAACL. http://www.isi.edu/natural-language/people/hovy/papers/03HLT-NAACL-ROUGE eval.pdf

32. **Lorch, R. (1993).** Effects of signaling topic structure on text recall. Journal of Educational Psychology.85(2):281-90.

33. **Luhn, H. (1953).** A new method of recording and searching information. American Documentation 4(1):14-16

34. **Luhn, H. (1958).** The automatic creation of literature abstracts. IBM Journal of Research and Development, 2(2):159-165.

35. **Mani, I. (2001).** Automatic Summarization. MITRE Corporation and Georgetown University. Natural language processing series, edited by Ruslan Mitkov, vol: 3. ISBN:1-58811-059-1.

36. **Mani, I. (1999a).** Improving Summaries by Revising Them. Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, p: 558-565.

37. **Mani, I. (1999b).** Advances in Automatic Text Summarization. MIT Press. Cambridge, ISBN:0-262-13359-8,

38. **Mani, I. (1998).** Machine learning of generic and user-focused summarization. In Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI'98), p:821–826.

39. **Marcu, D. (1997).** The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts. Tesis (Doctoral).University of Toronto.Canada. http://www.isi.edu/~marcu/papers/phd-thesis.ps.gz

40. **Marcu, D. (1999)**. The automatic construction of large-scale corpora for summarization research. In Proceedings of the SIGIR'99. http://www.isi.edu/~marcu/papers/sigir99.ps

41. **McKeown, K. et al (2002).** The Columbia Multi-Document Summarizer. Proceedings of the Workshop on Multi-Document Summarization Evaluation of the 2nd Document Understanding Conference at the 40th Meeting of the Association for Computational Linguistics.

42. **Miike, S. et al (1994).** A full-text retrieval system with a dynamic abstract generation function. In Proceedings of the 17th International Conference on Research and Development in Information Retrieval (SIGIR'94), p: 152–161.

43. **Miller, G. (1990)**. Wordnet: An on-line lexical database. International Journal of Lexicography, 3(4):235–312.

44. **Musciano, C. y Kennedy, B. (2000)**. HTML & XHTML: The Complete Guide.

45. **Nenkova, A. et al (2004).** Evaluating content selection in summarization. En Proceedings of the HLT-NAACL Conference. Boston. http://www.cs.columbia.edu/~ani/papers/pyramid.pdf

46. **Nielsen, J. (1997a).** How users read on the web. http://www.useit.com/papers/webwriting/writing.html

47. **Nielsen, J. (1997b).** Report from a 1994 web usability study. http://www.useit.com/papers/1994_web_usability_report.html

48. **Nomoto, T. y Matsumoto, Y. (2001).** A new approach to unsupervised text summarization. http://www.icst.pku.edu.cn/course/TextMining/0708Spring/%E5%8F%82%E8%80%83%E6%96%87%E7%8C%AE/0902%20A%20new%20approach%20to%20unsupervised%20text%20summarization.pdf

49. **O'neill, E. el al (2009).** Trends in the Evolution of the Public Web. D-Lib Magazine. 9(4) ISSN: 1082-9873 http://www.dlib.org/dlib/april03/lavoie/04lavoie.html

50. **Pierre, J. (2001).** On the automated classification Web Site. http://www.sukidog.com/jpierre/etai.pdf

51. **Radev, D. et al (2002).** Introduction to the Special Issue on Summarization Computational Linguistics. 28(4):399-408. http://www.mitpressjournals.org/doi/pdf/10.1162/089120102762671927

52. **Riboni, D. (2002).** Feature selection for web page classification. http://homes.dico.unimi.it/~riboni/eurasia02.pdf

53. **Salton, G. (1994).** Automatic Analysis, Theme Generation, and Summarization of Machine-Readable Texts. Science, 264(3):1421-1426.

54. **Salton, G. (1996).** Automatic Text Decomposition and Structuring. Information Processing & Management, 32(2):127-138.

55. **Salton, G. y McGill, M. (1987).** Introduction to Modern Information Retrieval. En McGraw-Hill Book Company.

56. **Salton, G. (1996)**. Automatic Text Decomposition Using Text Segments and Text Themes. Hypertext '96, p: 53-65.

57. **Salton, G. (1997).** Automatic Text Structuring and Summarization. Information Processing & Management, 33(2):193-207.

58. **Salton, G. et al (1994).** Automatic Text Theme Generation and the Analysis of Text Structure. Ithaca, NY: Cornell University.

59. **Sanchez, R. et al (2001).** Effects of heading s on text processing strategies. University of Kentucky. 26(3): 418-428

60. **Shannon, C. (1951).** Prediction and entropy of printed English. Bell Systems Technical Journal.
http://languagelog.ldc.upenn.edu/myl/Shannon1950.pdf

61. **Sparck-Jones, K. (1999).** Factors and directions. In Advances in Automatic Text Summarization. Automatic summarizing, p: 1–12.
http://www.ldc.upenn.edu/acl/J/J98/J98-2013.pdf

62. **Sparck-Jones, K.(1995)**.Reflections on TREC. Information Processing & Management. 31: 291-314

63. **STATGRAPHICS Centurion XV. 2006.** Stat Point, Inc.

64. **Vanderwende, L. et al (2006).** Event-centric summary generation. http://duc.nist.gov/pubs/2004papers/microsoft.banko.pdf

65. **Zipf, G. (1949).** Human behavior and the principle of least effort. Addison-Wesley Press. Cambridge, Mass.

Siglario de Término

Bin Función de ponderación binaria, (Binary)

BinIDF Función de ponderación basada en la frecuencia inversa del documento, (Binary-InverseFrequency Document)

CLEF Cross Language Evaluation Forum

DC Agrupación de documentos, (Document Clustering)

DOM Document Objet Model

GAR Generación Automática de Resúmenes

HTML Lenguaje de marcado de hipertexto, (HyperText Language Markup)

SGML Standard Generalized Markup Language

TC Clasificación automática de textos, (Text Classification)

TF Función de ponderación basada en frecuencias de aparición o bolsa de palabras, (TermFrequency)

TF-IDF Función ponderación basada en la frecuencia de un rasgo corregida con la frecuencia inversa del documento, (Text Frequency - Inverse Document Frequency)

RI Recuperación de información

URL Ubicador Uniforme de Recursos, (Uniform Resource Locator)

VSM Modelo de espacio vectorial, (Vector Space Model)

CCS Cascading Style Sheets

Glosario de Términos

Recuperación de Información*:* Localización, dentro de una colección de documentos, de un subconjunto relevante para una consulta formulada por un usuario.

Categorización de Textos: Asignación de un documento a una categoría previamente conocida.

Agrupamiento de Textos: Agrupación de documentos con características similares.

Minería de Texto: Consiste en la búsqueda a partir de técnicas de aprendizaje automático de regularidades o patrones que se encuentran dentro de un texto

Procesamiento del lenguaje Natural (PLN**):** Es una subdisciplina 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.