Una panorámica de la construcción de extractos de un texto

Henry Anaya Sánchez[1], Aurora Pons Porrata[1], y Rafael Berlanga Llavori[2]

[1] Departamento de Computación, Universidad de Oriente.

Patricio Lumumba s/n, Santiago de Cuba, Cuba.

{henry, aurora}@csd.uo.edu.cu

[2] Departamento de Lenguajes y Sistemas Informáticos, Universidad Jaume I.

Campus del Riu Sec, Castellón, España.

berlanga@lsi.uji.es

Title An overview of single document summarization by text extraction.

Resumen Este trabajo proporciona una panorámica de los diferentes algoritmos de construcción de extractos de un documento de texto. Se analizan tanto aproximaciones supervisadas como no supervisadas. El trabajo concluye con las principales ventajas y desventajas de estos métodos.

Abstract This paper provides an overview of different single document summarization algorithms based on text extraction techniques. Both supervised and unsupervised approaches are discussed. The paper concludes with the main advantages and disadvantages of these methods.

Palabras claves: resúmenes de textos, extractos.

Keywords: text summarization, extracts.

1. Introducción

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

¿Por qué la construcción automática de resúmenes? En los últimos años, el rápido crecimiento de la WWW y de los servicios de información electrónica ha traído consigo un vertiginoso crecimiento del volumen y la redundancia de la información de que disponen los usuarios, lo que hace difícil su asimilación. La construcción automática de resúmenes es una técnica indispensable para el tratamiento de este problema. Ejemplo de ello ha sido su uso en bibliotecas

científicas (para la elaboración de resúmenes de artículos), y en el área de la Recuperación de Información (para ayudar a presentar los resultados de una búsqueda y reducir la representación de los documentos en las colecciones).

Un resumen puede ser construido a partir de un solo documento de texto (en inglés, *single document summarization*) o de un conjunto de documentos (*multi-document summarization*). Además, un resumen se dice que es un extracto si está compuesto sólo por material copiado de la fuente de información, y si al menos una parte de él no ocurre en ella se dice que es un abstracto.

Este artículo pretende brindar una panorámica de la construcción automática de extractos de un documento de texto. La segunda sección tratará los términos y nociones básicas del área. La tercera presentará algunos algoritmos de construcción de extractos y la cuarta tocará el tema de la revisión de los extractos. Finalmente, se mostrarán las principales ventajas y limitaciones de los algoritmos de construcción de extractos existentes en la actualidad.

2. Preliminares

Un **extracto**, de manera más formal, es un resumen obtenido mediante la aplicación de operaciones de selección al contenido de una fuente. Consideraremos que una operación de selección se puede definir a través de una función cualquiera f que cumple que, si $X=e_1e_2\ldots e_n$ es un texto, donde para todo $k\in\{1, \ldots, n\}$ e_k es un elemento de X, entonces existe un texto $Y=e_{i_1}e_{i_2}\ldots e_{i_m}$ tal que $f(X)=Y$ y existen $j_1, \ldots, j_m$ tales que $1\leq j_1<\ldots<j_m\leq n$ 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*. Un **abstracto** es simplemente un resumen que no es un extracto.

Un resumen, abstracto o extracto, puede ser clasificado también según el tipo de usuario al que está destinado. En tal sentido, dos clases de resúmenes se pueden distinguir: la clase de los resúmenes genéricos y la clase de los resúmenes que pueden ser enfocados a un usuario, a un tópico o a una consulta [13]. 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 su construcción se deben tener en cuenta, además del contenido de la fuente, los intereses del usuario (expresados

generalmente a través de una consulta). Ejemplos de resúmenes enfocados a un usuario son los que muestran los buscadores de Internet para cada documento recuperado.

Otra forma de catalogar a los resúmenes es según su función, esto es, los resúmenes pueden ser **indicativos** o **informativos** [13]. Un resumen informativo cubre toda la información relevante de un texto fuente a un determinado nivel de detalle, y debe servir como sustituto del mismo. Los resúmenes indicativos son usados para condensar textos de poca estructuración y gran extensión tales como editoriales, ensayos, libros, etc., y proporcionan un indicio, también a determinado nivel de detalle, del contenido de un texto fuente para ayudar al lector a decidir si leer o no el texto completo.

Otros términos de importancia en esta área son: la **razón de compresión** de un resumen (o **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). 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.

El proceso de la construcción automática de resúmenes se puede dividir en tres fases: la **fase de análisis**, la **fase de transformación** y la **fase de síntesis** [23]. 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, por medio de operaciones de condensación (selección, agregación y generalización), la representación interna obtenida a la representación interna del resumen. Por último, la fase de síntesis transforma el resumen 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.

Los algoritmos de construcción automática de resúmenes se pueden clasificar en cuanto a los niveles de análisis lingüístico que emplean y a los tipos de elementos del texto sobre los que operan en dos grandes grupos: los **de**

estrategia poco profunda y los **de estrategia profunda** [13]. Los niveles de análisis lingüístico son (por orden de complejidad de menor a mayor): el nivel morfológico, el sintáctico, el semántico y el nivel de discurso.

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 al nivel semántico. Por su parte, los algoritmos de estrategia profunda realizan el análisis al menos al nivel semántico y los elementos del texto sobre los que operan no son menos complejos que las cláusulas. De manera general, los algoritmos de estrategia poco profunda producen extractos y son muy robustos, mientras que los de estrategia profunda generan abstractos y se aplican a fuentes de un dominio específico.

2.1. Evaluación de los resúmenes construidos de manera automática

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 [24], 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, mientras que los **métodos extrínsecos** evalúan la eficiencia y el desempeño de los resúmenes en una tarea determinada. Por lo general, los métodos intrínsecos 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 problemas de coherencia tales como referencias sin resolver y fisuras en la estructura de discurso. De aquí que un resumen se pueda evaluar según su coherencia, la cual puede medirse teniendo en cuenta la presencia de anáforas sin resolver y la falta de preservación de ambientes estructurados como listas y tablas en su texto [20]. Otra medida de coherencia [7] 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* [25] 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 definen como:

<<Fórmula 1>>

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

- **El contenido del resumen**. Un resumen puede ser evaluado comparando su contenido con el de un resumen de referencia o con el de su texto fuente [5]. 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 [22]. Si en esta evaluación está involucrado un abstracto, debe usarse algún tesauro de términos a la hora de representar los textos.

- **Los *n*-gramas del resumen**. Para evaluar la calidad de un resumen, en lugar de considerar las sentencias, se pueden tener en cuenta los *n*-gramas[1]. ROUGE-*n* [12] es una medida basada en la ocurrencia de los *n*-gramas que evalúa la relevancia de un resumen *r* a partir de un conjunto de resúmenes de referencia *C* y se define como:

<<Fórmula 2>>

donde *n-gramas*(*p*) es el conjunto de *n*-gramas del texto *p* y *cant*(*g*, *p*) denota la cantidad de veces que aparece el *n*-grama *g* en *p*.

Otras medidas de evaluación intrínsecas se pueden encontrar en [5].

Por otra parte, los métodos de evaluación extrínsecos exigen casi siempre una activa participación de personas. Uno de los más sencillos es el de *Lectura de comprensión* [13]. 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 ([13] y [24]), éste puede ser utilizado también para evaluar el contenido informativo de un resumen.

Con el objetivo de comparar la calidad de los algoritmos de construcción de resúmenes cada año se realiza una competición internacional que se conoce

[1] Un *n*-grama es una secuencia de n palabras consecutivas de un texto.

con el nombre de DUC (*Document Understanding Conference*[2]). En la realizada en el año 2004, se estableció como medida de evaluación ROUGE-*n*. 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.

3. Construcción de extractos de un documento de texto

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

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 o no al extracto. Esta clasificación se realiza teniendo en cuenta algunos rasgos de los elementos del texto, que pueden ser: lingüísticos, estadísticos, comunicativos o ser rasgos específicos del dominio del texto que se resume. Precisamente por reducirse a problemas de clasificación, los algoritmos de construcción de extractos pueden ser supervisados o no supervisados, por lo que organizaremos las aproximaciones existentes según este aspecto.

Es necesario mencionar que los algoritmos que presentaremos no han sido elegidos al azar, pues ellos constituyen la base fundamental de la mayoría de los sistemas de construcción de extractos actuales.

3.1. Algoritmos de construcción de resúmenes 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. Debido al uso de tal colección de entrenamiento, por lo general, estos algoritmos obtienen resúmenes de textos de materias específicas. Algunos de los algoritmos supervisados más representativos se presentan a continuación.

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

Algoritmo de Edmundson [6]. En este algoritmo cada sentencia del texto fuente se representa como un vector de cuatro componentes, que se corresponden con los valores de los rasgos *palabras-pistas*, *palabras-claves*, *palabras-título* y *localización* de la sentencia. A cada sentencia *s* se le asigna una puntuación definida como $W(s)=\alpha C(s)+\beta K(s)+\gamma L(s)+\delta T(s)$ donde $C(s)$, $K(s)$, $T(s)$ y $L(s)$ denotan, respectivamente, los valores de los rasgos *palabras-pistas*, *palabras-claves*, *palabras-título* y *localización* de *s*, y α, β, γ y δ son sus pesos asociados. El primer rasgo pondera las sentencias de acuerdo con la frecuencia de aparición de sus palabras en los resúmenes de una colección de entrenamiento. Los rasgos *palabras-título* y *palabras-claves* favorecen a las sentencias que presentan palabras del título y, palabras muy frecuentes en el documento, respectivamente. Al mismo tiempo, el rasgo *localización* favorece a las sentencias que se encuentran cerca del comienzo y del final del texto, pues se supone que éstas contienen información importante para los resúmenes por pertenecer a la introducción y a las conclusiones.

El algoritmo clasifica a las *m* sentencias de mayor puntuación como pertenecientes al extracto. A él se le censura el carácter lineal de la función de evaluación de las sentencias. A pesar de esto, es considerado un paradigma entre los algoritmos de construcción de extractos debido a que la inmensa mayoría incluyen versiones de los rasgos usados por Edmundson.

Algoritmo de Kupiec [10]. En este algoritmo las sentencias de un texto fuente se representan mediante un vector de valores $(v_1,\ldots,v_n)$ correspondientes a los rasgos $R_1,\ldots,R_n$. Si denotamos por $R_1(s),\ldots,R_n(s)$ a los valores de estos rasgos en la sentencia *s*, la función de evaluación que emplea el algoritmo se define como la probabilidad de que la sentencia *s*, representada por el vector $(v_1,\ldots,v_n)$, sea incluida en el extracto, esto es, $W(s)=P(s\in E|R_1(s)=v_1\ldots R_n(s)=v_n)$. Aplicando Naïve Bayes esta probabilidad puede ser calculada mediante la ecuación:

<<Fórmula 3>>

donde $P(s\in E)$ denota la probabilidad a priori de que una sentencia de un texto fuente sea incluida en su extracto, y $\forall k\in\{1,\ldots,n\}$ los valores $P(R_k(s)=v_k|s\in E)$ y $P(R_k(s)=v_k)$ son constantes que denotan, respectivamente, la probabilidad de que el rasgo R_k de una sentencia que pertenece al extracto sea igual a la

constante v_k y la probabilidad de que el rasgo R_k de una sentencia sea igual a v_k. Las tres probabilidades anteriores deben ser calculadas a partir de una colección de entrenamiento. Los rasgos usados son todos discretos y tienen en cuenta: si la longitud de la sentencia es mayor que un umbral, la localización de la sentencia dentro de uno de los 10 primeros o 10 últimos párrafos del texto, y la presencia de frases que indican resumen, de palabras temáticas y de siglas frecuentes del texto fuente en la sentencia. Al igual que el algoritmo de Edmundson, éste clasifica a las *m* sentencias de mayor puntuación como pertenecientes al extracto.

Es de resaltar que este algoritmo ha sido también un punto de partida en el diseño de otros algoritmos de construcción de extractos basados en colecciones de entrenamiento. Entre otros, podemos citar a los algoritmos descritos en [14] y [11]. En estos trabajos se consideran grupos más refinados de rasgos y la generación de extractos enfocados a un tópico. Ambos usan un árbol de decisión como clasificador.

Entre los sistemas supervisados que participaron en DUC-2004 se encuentran: *Lake System* [4] y *News Story Gisting* [26], que construyen resúmenes mediante la extracción de frases claves de la fuente considerando rasgos lingüísticos y estadísticos. Ellos obtuvieron valores promedios de 0,1854 y 0,1966, respectivamente, al ser evaluados con la medida ROUGE-1, los cuales son similares a los obtenidos por los otros sistemas.

Un esquema general de la estrategia de los algoritmos basados en colecciones de entrenamiento se puede apreciar en la Figura 1 [13].

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

Generalmente, la colección de entrenamiento con que se cuenta está compuesta por pares texto *fuente - abstracto* en lugar de pares *texto fuente -*

extracto. Esto hace necesario disponer de algoritmos que generen un extracto a partir de un abstracto. En [13] se describen dos estrategias bastante generales para la solución de este problema: la estrategia de emparejamiento combinado y la de emparejamiento individual. Estas estrategias difieren en la forma de evaluar a las sentencias del texto fuente para construir el extracto.

<<Figura 1>>

En la estrategia de emparejamiento combinado, a cada sentencia del texto fuente le es asignada una puntuación basada en su semejanza con el abstracto completo tomado como una sentencia. La función de semejanza usada en [14], basada en la medida del coseno, fue:

<<Fórmula 4>>

donde i_{s_1} y i_{s_2} son los valores respectivos de los *tf-idf* de la palabra *i* en las sentencias s_1 y s_2, N_1 es el número de palabras en común de s_1 y s_2, y N_2 es el número de total de palabras de s_1 y s_2. Luego, las sentencias se ordenan de mayor a menor de acuerdo con su puntuación y se seleccionan para formar parte del extracto las primeras que constituyan no más del 100*c*% del total, donde *c* es la razón de compresión deseada del extracto.

Los algoritmos de emparejamiento combinado pueden ser usados también para obtener un resumen de un texto fuente enfocado a una consulta considerando a la misma en lugar del abstracto.

En la estrategia de emparejamiento individual, a cada sentencia del texto fuente le es asignada como puntuación el máximo valor obtenido de la comparación de ésta con cada una de las sentencias del abstracto. Luego se procede como en la estrategia de emparejamiento combinado para seleccionar las sentencias del extracto.

Es preciso señalar que existen otras estrategias, como el algoritmo ávido de Marcu [17], que no son ni de emparejamiento combinado ni individual. En ésta se genera un extracto partiendo de un texto igual al texto fuente, al que se le van eliminando iterativamente las sentencias que menos se parecen al abstracto, mientras no decrezca la semejanza entre él y el abstracto.

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

3.2. Algoritmos no supervisados

Los algoritmos no supervisados siguen básicamente dos esquemas. El primero consiste en ponderar cada elemento del texto individualmente, considerando propiedades intrínsecas de éstos en el texto (estadísticas y lingüísticas), para luego clasificar los elementos de mayor peso en elementos del extracto de manera similar al algoritmo de Edmundson. Un algoritmo que proporciona un marco bastante general para los algoritmos no supervisados del primer esquema se puede encontrar en [8]. 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.

En esta subsección trataremos los algoritmos que siguen el segundo esquema pues son de estrategia profunda; y además, como se indicó, los algoritmos no supervisados que siguen el primer esquema son muy similares al algoritmo de Edmundson (su diferencia radica en la no utilización de una colección de entrenamiento). Los algoritmos del segundo esquema, se basan en dos características del texto: la cohesión y la coherencia, y se dividen por ellas.

La cohesión de un texto está dada por ciertas relaciones semánticas que se establecen entre sus palabras y frases, y que establecen cuán estrechamente conectado está el texto. Estas relaciones son de dos clases: gramaticales y léxicas. Entre las de la primera clase están las relaciones anafóricas y las de elipsis, mientras que las relaciones de sinonimia, hiperonimia y repetición, entre otras, se encuentran en la segunda. La idea es que la cohesión de un texto sea usada para obtener los conceptos o los tópicos prominentes de un texto para luego generar a partir de éstos el extracto.

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

Algoritmo de Barzilay basado en la cohesión léxica [2]. Se ha comprobado de manera experimental que los rasgos que tienen en cuenta la frecuencia de ocurrencia de las palabras en un texto tienden a ser de los menos importantes

a la hora se seleccionar los elementos del extracto. Para solucionar este problema proponen un algoritmo que usa las cadenas léxicas de un texto[3].

Este algoritmo para cada cadena léxica fuerte del texto fuente, selecciona la primera sentencia del texto que contiene un miembro representativo de la cadena y la clasifica como perteneciente al extracto. Para el cálculo de las cadenas léxicas, se tienen en cuenta las relaciones léxicas existentes entre las palabras tomando como base la ontología WordNet [19]. La fortaleza de una cadena léxica toma en consideración su longitud y homogeneidad. Un miembro de una cadena es representativo si su frecuencia de aparición en la cadena supera la frecuencia promedio.

Algoritmo de Nomoto y Matsumoto [21]. Este algoritmo define a un extracto como un conjunto de sentencias extraídas de un texto fuente que cubren su esencia; y basa la construcción de tal conjunto en dos importantes propiedades de un texto: la diversidad y la redundancia de los conceptos que en él ocurren. El algoritmo particiona el conjunto de las sentencias del texto fuente usando un algoritmo de agrupamiento de manera tal que cada subconjunto esté compuesto por sentencias que representan un tópico del texto. Luego define el extracto como el conjunto formado por la sentencia más importante de cada subconjunto. La elección de tales sentencias se realiza teniendo en cuenta la frecuencia de sus términos en el documento.

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

Existen muchos algoritmos y sistemas de construcción de extractos basados en la cohesión de un texto. Ejemplos de ellos, que participaron en DUC-2004, son *ERSS System* [3] y *K.U. Leuben Summarization System* [1]. Éstos alcanzaron

[3] Una cadena léxica es una secuencia de palabras próximas en un texto que cubren o caracterizan un tópico de éste.

en promedio las puntuaciones de 0.2 y 0.16675 usando la medida ROUGE-1, respectivamente. Hay que señalar que algunos algoritmos supervisados e incluso algoritmos no supervisados del primer esquema especificado, emplean también rasgos de cohesión en sus clasificadores.

Por otra parte, los algoritmos basados en la coherencia de un texto son extremadamente escasos en la literatura, debido a que la obtención de la estructura de discurso de un texto no es una tarea sencilla y continúa aún siendo un desafío, y a que ellos no garantizan la obtención de extractos coherentes. Estos algoritmos construyen un árbol etiquetado que representa la estructura de discurso del texto a resumir. Sus hojas son elementos del texto y sus nodos interiores representan la relación existente entre sus nodos hijos y tienen asociados los elementos del texto que constituyen el núcleo de esa relación. Luego, construyen un orden parcial entre los elementos del texto teniendo en cuenta la profundidad del nodo del que constituyen su núcleo. Por último, se clasifican los primeros *m* elementos del orden como pertenecientes al extracto. Algoritmos de este tipo se pueden encontrar en [18] y [16].

4. Revisión

Una vez creado un extracto le pueden ser aplicados métodos de revisión, como parte de la fase de síntesis, para mejorar su coherencia y contenido informativo. Existen dos métodos de revisión fundamentales, ellos son: el ajuste de la coherencia y la revisión completa de los extractos [13].

En el primer método se aplican estrategias sencillas para reconocer y, excluir o reparar ambientes estructurados del texto, tales como tablas y listas, que se presentan dañados en el extracto. También se eliminan sentencias que comiencen con anáforas, o se resuelven éstas últimas mediante la inclusión de sentencias que ocurren en su contexto. Incluso, algunas fisuras en la estructura de discurso se enmiendan mediante la eliminación de conjunciones y partículas adverbiales en las sentencias.

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. Para más detalles de este tipo de revisión se puede consultar [15].

Un aspecto muy importante a la hora de aplicar estos métodos de revisión es que se debe tener conocimiento de la estructura sintáctica e incluso de la estructura de discurso de los extractos.

5. Conclusiones

La mayoría de los algoritmos de construcción de extractos de textos que existen en la actualidad son de estrategia poco profunda. Al tener esta característica, requieren de poco conocimiento lingüístico y por tanto son adaptables a cualquier lenguaje, además de ser de fácil instrumentación. No obstante, se ha desarrollado también un buen número de algoritmos que emplean la cohesión del texto como característica inherente al nivel de discurso.

Entre las limitaciones que se les pueden señalar a los algoritmos que construyen extractos están: la falta de coherencia en los textos generados (aún cuando en la fase de síntesis se pueden aplicar métodos de revisión, éstos llegan muy tarde), y la no consideración de la razón de compresión del resumen como parte del proceso de clasificación de los elementos de un texto.

Además, estos algoritmos tienden a generar resúmenes indicativos, y aquellos que tratan de generar extractos informativos obtienen materiales que o son redundantes en informaciones específicas o no cubren toda la información relevante del texto. Muestra de ello lo constituyen los bajos valores que obtienen al ser evaluados con respecto a los resúmenes de referencia.

Es por ello que en la actualidad se continúa trabajando en el diseño de algoritmos de extracción que superen las limitaciones antes mencionadas. En particular, dos líneas de investigación que han surgido recientemente son la generación de titulares (*headline generation*), y la tarea *Novelty* [9]. A grandes rasgos la generación de titulares puede ser vista como una especialización de la tarea de construcción de resúmenes encaminada a la creación de sumarios coherentes con elevada razón de compresión. Por su parte, la tarea de Novelty consiste en, dados un tópico y un flujo ordenado de documentos relevantes al mismo, obtener las sentencias de cada documento que son relevantes al tópico y que aportan información novedosa sobre él.

Referencias

1. Angheluta, R.; Mitra, R.; Jing, X. and Moens, M.-F.: K.U. Leuven Summarization System at DUC-2004. In *Document Understanding Workshop*, 2004.

2. Barzilay, R. and Elhadad, M.: Using Lexical Chains for Text Summarization. In *Advances in Automatic Text Summarization*, pp. 111–121, MIT Press, 1999.

3. Bergler, S.; Witte, R.; Li, Z.; Khalife, M.; Chen, Y.; Doandes, M. and Andreevskaia, A.: Multi-ERSS and ERSS 2004. In *Document Understanding Workshop*, 2004.

4. D'Avanzo, E.; Magnini, B. and Vallin, A.: Keyphrase extraction for summarization purposes: The Lake System at DUC-2004. In *Document Understanding Workshop*, 2004.

5. Donaway, R. L.; Drummey, K.W. and Mather, L. A.: A comparison of rankings produced by summarization evaluation measures. In *Proceedings of the Workshop on Automatic Summarization at the 6th Applied Natural Language Processing Conference*, pp. 69–78, 2000.

6. Edmundson, H.P.: New methods in automatic extracting. *Journal of the Association for Computing Machinery*, 16:264–285, 1969.

7. Eneva E.; Hoberman, R and Lita, L.: Learning Within-Sentence Semantic Coherence. In *2001 Empirical Methods in Natural Language Processing (EMNLP 2001)*, 2001.

8. Goldstein, J.; Kantrowitz, M.; Mittal, V. and Carbonell, J.: Summarizing text documents: sentence selection and evaluation metrics. In *Proceedings of SIGIR'99*, pp. 121–128, 1999.

9. Harman, D.: Overview of the TREC 2002 novelty track. In *Proceedings of TREC 2002 (Notebook)*, pp. 17–28, 2002.

10. Kupiec, J.; Pedersen, J. and Chen, F.: A trainable document summarizer. In *Proceedings of SIGIR'95*, pages 68–73, 1995.

11. Lin, C.-Y.: Training a selection function for extraction. In *Proceedings of the 8th Annual International ACM Conference on Information and Knowledge Management CIKM*, 1999.

12. Lin, C.-Y. and Hovy, E.: Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of HLTNAACL*, 2003.

13. Mani, I.: *Automatic Summarization*. John Benjamis Publishing Co., 2001.

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

15. Mani, I. and Maybury, M.T.: *Advances in Automatic Summarization*. MIT Press, 1999.

16. Marcu, D.: *The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts*. PhD thesis, University of Toronto, 1997.

17. Marcu, D.: The automatic construction of large-scale corpora for summarization research. In *Proceedings of the SIGIR'99*, pp. 137–144, 1999.

18. Miike, S.; Itoh, E.; Ono, K. and Sumita, K.: 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)*, pages 152–161, 1994.

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

20. Minel, J.L.; Nugier, S. and Piat, G.: How to appreciate the quality of automatic text summarization. In *Proceedings of the ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization*, pages 25–30, 1997.

21. Nomoto, T. and Matsumoto, Y.: A new approach to unsupervised text summarization. In *Proceedings of SIGIR*, pages 26–34, 2001.

22. Salton, G. and McGill, M. J.: *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company, 1983.

23. Sparck-Jones, K.: Automatic summarizing: Factors and directions. In *Advances in Automatic Text Summarization*, pages 1–12. MIT Press, 1999.

24. Sparck-Jones, K. and Galliers, J. R.: Evaluating natural language processing systems: An analysis and review. *Lecture Notes in Artificial Intelligence*, 1083, 1995.

25. van Rijsbergen, C.J.: *Information Retrieval*. 2nd Edition, Butterworths, London, 1979.

26. Doran, W.; Stokes, N.; Newman, E.; Dunnion, J.; Carthy, J. and Toolan, F.: News Story Gisting at University College Dublin. In *Document Understanding Workshop*, 2004.

<<Fórmula 1>>:

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

<<Fórmula 2>>:

$$\frac{\sum_{t \in C} \sum_{g \in n-gramas(t)} mín\{cant(g,t), cant(g,r)\}}{\sum_{t \in C} \sum_{g \in n-gramas(t)} cant(g,t)}$$

<<Fórmula 3>>

$$P(s \in E \mid R_1(s) = v_1 \ldots R_n(s) = v_n) = P(s \in E) \frac{\prod_{i=1}^{n} P(R_i(s) = v_i \mid s \in E)}{\prod_{i=1}^{n} P(R_i(s) = v_i)}$$

<<Figura 1>>

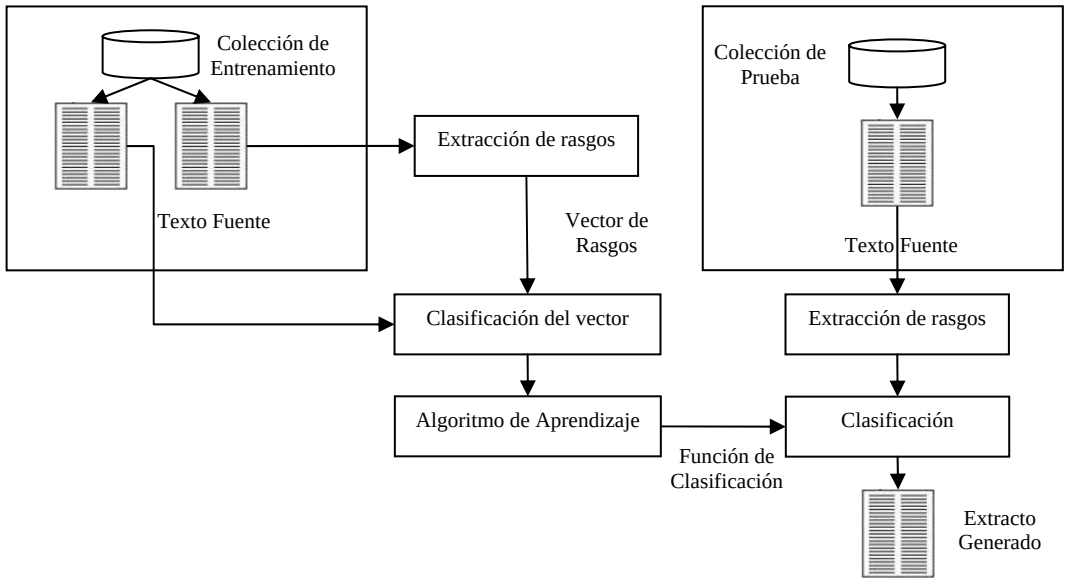

Figura 1. Esquema general de los algoritmos basados en una colección de entrenamiento.

<<Fórmula 4>>

$$N_1 + \frac{\sum_{i=1}^{N_2} i_{s_1}\ i_{s_2}}{\sqrt{\sum_{i=1}^{N_2} i_{s_1}{}^2 \sum_{i=1}^{N_2} i_{s_2}{}^2}}$$